



HAL
open science

Approches exactes et heuristiques pour l'exploration d'architectures logicielles et matérielles

L Lemarchand

► **To cite this version:**

L Lemarchand. Approches exactes et heuristiques pour l'exploration d'architectures logicielles et matérielles : Applications dans les domaines du Temps Réel, du Cloud et du Routage. Informatique [cs]. Université de Bretagne Occidentale, Brest, FRA., 2023. tel-04323458

HAL Id: tel-04323458

<https://hal.univ-brest.fr/tel-04323458>

Submitted on 5 Dec 2023

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.

HABILITATION À DIRIGER DES RECHERCHES

L'UNIVERSITÉ DE BRETAGNE OCCIDENTALE

ÉCOLE DOCTORALE N° 601
*Mathématiques et Sciences et Technologies
de l'Information et de la Communication*
Spécialité : *Informatique*

Par

Laurent LEMARCHAND

**Approches exactes et heuristiques pour l'exploration
d'architectures logicielles et matérielles**

Applications dans les domaines du Temps Réel, du Cloud et du Routage

HDR présentée et soutenue à Brest, le 27 novembre 2023

Unité de recherche : Lab-STICC, UMR 6285

Rapporteurs avant soutenance :

Olivier BEAUMONT Chercheur, INRIA Bordeaux
Clarisse DHAENENS Professeur, Université de Lille
Xavier GANDIBLEUX Professeur, Université de Nantes

Composition du Jury :

Président :	Frank SINGHOFF	Professeur, Université de Bretagne Occidentale
Rapporteurs :	Olivier BEAUMONT	Chercheur, INRIA Bordeaux
	Clarisse DHAENENS	Professeur, Université de Lille
	Xavier GANDIBLEUX	Professeur, Université de Nantes
Examineurs :	Jean-Philippe BABAU	Professeur, Université de Bretagne Occidentale
	Patrick MEYER	Professeur, IMT Atlantique

REMERCIEMENTS

Je remercie vivement mes trois rapporteurs, Olivier Beaumont, Clarisse Dhaenens et Xavier Gandibleux pour le temps qu'ils ont consacré à mes travaux et leurs retours sur ce manuscrit, ainsi que Patrick Meyer pour sa participation au jury de cette HDR.

Je ne remercie pas Frank Singhoff et Jean-Philippe Babau pour leur participation au jury de HDR uniquement, mais je les associe à Jalil Boukhobza dans mes remerciements pour leur soutien sans faille toutes ces années et nos collaborations en recherche sur leurs thématiques respectives.

Je remercie également chaleureusement les doctorants et tous mes autres collègues du Lab-STICC et du département informatique avec qui je travaille au quotidien en enseignement, en recherche et au niveau collectif, dans une ambiance très professionnelle mais aussi très agréable et conviviale.

Merci à tous !

TABLE DES MATIÈRES

Parcours de recherche	1
Introduction	3
Optimisation combinatoire	4
Contributions	5
Plan du document	8
1 Optimisation multi-objectif	9
1.1 Définitions et métriques de comparaison	9
1.1.1 Définitions	9
1.1.2 Métriques de comparaison	11
1.2 Méthodes de résolution	12
1.2.1 Méthodes mono-objectifs à résultat unique	15
1.2.2 Approches mono-objectifs à résultats multiples	15
1.2.3 Approches évolutionnaires multi-objectifs	17
1.3 Conclusion	20
2 Exploration d’architectures pour les systèmes embarqués temps réel	21
2.1 Problématique	21
2.2 Exploration architecturale analytique utilisant une méthode MILP en 2 phases	24
2.3 Exploration architecturale par simulation avec MOEA	26
2.4 Exploration d’architectures temps réel hiérarchiques sécurisées par MOEA	30
2.5 Conclusion	34
3 Systèmes de stockage	37
3.1 Introduction	37
3.2 Réserve de bande passante dans un réseau clos	37
3.2.1 Problématique	37
3.2.2 Modélisation et résolution	39
3.2.3 Résultats	41
3.3 Politique de stockage de mémoire Flash avec MOEA	43
3.3.1 Problématique	43
3.3.2 Résultats	44

3.4	Placement d'objets dans le Fog avec MILP et approche <i>diviser pour régner</i>	46
3.4.1	Problématique	46
3.4.2	Résultats	47
3.5	Stockage dans le Cloud	49
3.5.1	Stockage hybride	49
3.5.2	Placement d'objets dans des Clouds fédérés avec MILP et objectifs multiples	50
3.6	Conclusion	55
4	Réseaux de transport, routage et routage embarqué	57
4.1	Introduction	57
4.2	Réseaux de transport	58
4.2.1	Réticulum endoplasmique avec MILP et Branch & Cut	58
4.2.2	Transport multi-modal avec MILP et méthode MOO ϵ -constraints	59
4.3	Routage de voiliers avec MOEA	60
4.4	Routage de drones avec MILP et Branch & Cut	63
4.5	Conclusion	66
5	Bilan et perspectives de recherche	69
5.1	Bilan des actions de recherche	69
5.2	Projets en cours	71
5.2.1	MEDORAA : Décision embarquée exploitant un ensemble de Pareto	71
5.2.2	IA-AUV : TSPP pour la définition de missions d'un ROV	72
5.2.3	DISPEED : cybersécurité embarquée pour meute de drones	73
5.2.4	RESSACH : réseau de surveillance acoustique	74
5.3	Projet à court terme : OMER, routage de voiliers	74
5.4	A moyen terme	77
5.4.1	Résolution exacte pour la définition de missions de drone marin	77
5.4.2	Méthodes par substitution de modèle pour l'exploration d'architectures temps réel	78
5.4.3	Répartition d'équipes sportives en poules de championnat	79
5.5	A plus long terme	79
	Conclusion	81
	Bibliographie	83
	Publications	83
	Références bibliographiques	89

TABLE DES FIGURES

1	Projets de 2011 à 2022, par thème, avec les collègues et doctorants impliqués (ceux marqués de * sont en poste à l'étranger ; ceux marqués d'un + sont hors Lab-STICC ; les autres sont dans SHAKER), ainsi que les publications communes et les outils RO utilisés.	6
1.1	Relation de dominance et ensemble/front de Pareto	10
1.2	Comparaison de 2 fronts bi-objectifs avec 2 métriques (a) HV et (b) IGD	12
1.3	Une approche ϵ -constraint pour le calcul d'un ensemble de solutions par découpage de l'espace objectif (problème bi-objectif)	16
1.4	Principe de l'algorithme PAES	18
1.5	Classement par rang de dominance effectué par NSGA II (d'après [31]) pour 2 objectifs à minimiser	19
1.6	Déroulement de l'algorithme NSGA II ([31, 92])	19
2.1	Evolution du nombre de lignes de code embarquées à l'horizon 2030 dans les avions civils et militaires [26]	21
2.2	Les grandes étapes de la construction d'une architecture temps réel	22
2.3	Déploiement d'un ensemble de fonctions représenté par un flot de données sur une plate-forme d'exécution pour obtenir un modèle de déploiement	24
2.4	Temps d'exécution suivant le nombre de fonctions pour un déploiement sur 2 processeurs et 2 bus. Comparaison entre une approche MILP 2 phases avec un algorithme génétique et une approche MILP standard. O_l est la valeur de la fonction objectif [C06]	25
2.5	Méthode de résolution en 2 phases pour le déploiement	26
2.6	Approche MOEA/PAES pour l'exploration d'architecture par simulation [22]	27
2.7	Front exact de 7 solutions obtenu par énumération pour un cas de test à 11 fonctions [22]	28
2.8	(a) Qualité comparée et (b) accélération pour différentes stratégie de parallélisation de PAES pour un cas de test [22]	29
2.9	Corrélation positive, négative ou insignifiante entre préemptions (f_1), laxité (f_4) et temps de blocage (f_6) pour un jeu de test de 100 cas [22] avec un seuil de confiance de 0.05 [146]	30

2.10	Encodage d'une solution pour le partitionnement de tâches et la sécurisation des communication [7]	32
2.11	Compromis échéances ratées - communications non sécurisées pour les 3 stratégies de mutation [W07], en fonction du taux d'occupation du processeur par les tâches	33
2.12	Compromis trouvés pour un DSE sur 17x3 tâches, 2 partitions et 57 cœurs maximum [W07]	34
3.1	Le profil (débits dans le temps) d'un film, avec (a) les politiques de réservation constante maximale et moyenne et (b) une politique de réservation variable constituée de 4 configurations. Son coût associé en bande passante perdue correspond à la surface entre le profil et l'enveloppe. Une contrainte de durée minimale d'une configuration est illustré par la flèche verte.	38
3.2	2 enveloppes de réservation possibles et leur équivalent en terme de chemin dans le graphe. La première ($P = 3$) correspond à une période de reconfiguration minimale de 3, la seconde à une reconfiguration régulière toutes les 3 unités de temps. Chaque enveloppe correspond à un chemin dans le graphe de modélisation du problème [R04]	40
3.3	Coût en termes de bufférisation et de délai suivant la bande passante réservée pour (a) une réservation de taille constante (débit constant du Hierarchical Token Bucket (HTB)) et (b) une politique optimale avec une contrainte sur le nombre de réservations possibles $M = 149$ [R04]	41
3.4	Front exact et front obtenu après 500 itérations de PAES pour une trace synthétique [R02]	44
3.5	Approche par partitionnement pour l'affectation d'objets dans le Fog[C15]	46
3.6	Accélération des temps de résolution (à gauche) et perte d'optimalité (à droite) pour des approches par partitionnement du placement de données dans le Fog. Solution de référence exacte calculée par résolution IP en 283 secondes [C15] . .	48
3.7	Matheuristique pour le placement d'objets dans un Cloud fédéré [31]. A partir du modèle (a), un ensemble de MILP sont résolus en variant les poids (b) pour fournir les solutions supportées de départ pour l'exécution de la partie MOEA (<i>injection</i>). Un opérateur de <i>réparation</i> est inclus dans le cycle de l'algorithme évolutionnaire.	52

3.8	Comparaison des résultats (hypervolume, HV) et des temps de calcul (à droite) pour plusieurs approches de résolution. Résultats cumulés pour un ensemble de cas de test [C21]. I et R font référence au fait d’injecter (I) les solutions calculées par la méthode exacte et d’inclure l’opérateur de réparation (R). 2 MOEA sont comparés : NSGA II et PSO (une version de l’algorithme <i>Particle Swarm Optimization</i>). L’hypervolume de référence est celui correspondant à un ensemble de 10 solutions supportées calculées par résolution exacte MILP (trait pointillé bleu à gauche, courbe bleue à droite). A droite, les chiffres sur les courbes permettent de comparer, pour un nombre donné de solutions calculées avec CPLEX, les temps d’exécution et la qualité du front obtenus en injectant ces mêmes solutions dans la matheuristique NSGAI-IR	53
4.1	Un réseau vivant de transport de nutriments : le réticulum endoplasmique (à gauche) et des simulations de sa construction (2 images de droite)	59
4.2	Routage par isochrone, principe et résultat [C17]	61
4.3	(a) Opérateurs de mutation pour le routage de voiliers de course et (b) tirage de bord pour routage face au vent [C17]	62
4.4	Un circuit de routage d’un drone de supervision dans le port de Brest et les points couverts par son lidar à partir d’une position et d’une portée donnée (en médaillon) [W08]	63
4.5	(a) Un cas de test avec sa grille de sommets et des obstacles statiques [R10] et ses caractéristiques, (b) et (b’) valeur de fonction objectif <i>énergie</i> associée aux solutions avec taux de <i>couverture</i> $\geq 95\%$ en fonction du temps de calcul pour les 4 algorithmes PAES, PAES+, NSGA et NSGA+	65
5.1	Principe de l’approche MEDORAA	71

TABLE DES FIGURES

LISTE DES TABLEAUX

1	Quelques aspects positifs et négatifs des grandes familles de méthodes d'optimisation combinatoire	7
3.1	Coûts et valeur pic finaux pour différences approches concernant la diffusion simultanée de 4 flux multimédia, avec un décalage possible de $\Delta = 5$ [R04]	42

PARCOURS DE RECHERCHE

Titulaire d'un doctorat de l'université de Rennes 1 en 1999, ma thèse s'est déroulée dans l'équipe Architectures et Systèmes du Laboratoire d'Informatique des Systèmes Complexes (LISyC, EA2215) de l'université de Bretagne Occidentale, sous la direction conjointe de R. Euler (pour les aspects liés à l'optimisation combinatoire) et B. Pottier (pour le domaine applicatif de la synthèse de circuits FPGA). Ces deux aspects sont liés au parallélisme et à son utilisation dans le cadre de la résolution de problèmes de grande taille. J'ai ensuite été recruté en 1999 comme maître de conférences à l'UBO, intégré au LISyC. Durant la période 2000-2002, j'ai continué à travailler sur les perspectives avancées en doctorat concernant la synthèse de circuits [C01, RN01, RT01].

A partir de 2003, j'ai changé de domaine, en m'intéressant au test de cartes hybrides numériques/analogiques, en collaboration avec V.A. Nicolas, qui encadrait une thèse CIFRE sur le sujet avec la société ISIS-MP de Brest. Le doctorant (à mi temps) était Bertrand Gilles, PAST à l'UBO et ingénieur qualité chez Thalès. Il soutint sa thèse en 2009 [W03, W02, W01, W04]. Durant cette période, je me consacrais principalement aux tâches pédagogiques et administratives. Je fus notamment, à partir de 2010, chargé de mission pour l'UBO.

Lors de l'intégration des membres de l'EA 3883 au Lab-STICC (UMR CNRS 6285) en 2012, j'en suis devenu membre associé, puis rapidement membre permanent en 2013. J'ai saisi l'occasion de l'intégration au Lab-STICC pour m'impliquer en recherche sur 2 thématiques : (i) la modélisation sous forme de Programmes Linéaires en Nombres Mixtes (MILP¹ en anglais) pour des problématiques de placement et d'ordonnancement de systèmes dans le cadre de la thèse de Asma Mehiaoui [C04, C06] et (ii) des travaux que j'ai mené en propre avec Jean-Philippe Babau (Lab-STICC UBO), dans le prolongement de la thèse de Maxime Louvel, sur la diffusion avec garantie de qualité de service de vidéos dans des environnements réseau contrôlés. Pour répondre à une problématique soulevée par la thèse de Maxime, j'ai proposé une approche et développé un algorithme original inspiré de la théorie des graphes, algorithme décliné suivant les besoins avec plusieurs versions de calcul de plus courts chemins [C05, R04, C09]. En parallèle, à partir de 2011, j'ai commencé à m'intéresser à la thématique de l'ordonnancement temps réel, en rejoignant un projet Egide SAPIENT qu'animait Frank Singhoff (Lab-STICC UBO) avec des collègues de l'université de Lisbonne, Portugal [CN01]. Après les thématiques MILP et Graphes, je m'intéressais à nouveau aux métaheuristiques, et notamment aux algorithmes génétiques, sur

1. Mixed Integer Linear Programming

lesquelles j'avais travaillé dès 1992 [R01]. En discutant avec mes collègues pour les approches RO que je leur proposais, je me suis rendu compte du caractère souvent multi-objectif de leur problématique. Je décidais donc d'approfondir le sujet, et pris contact avec une équipe du CEMPS à Exeter (GB) qui utilisait des approches métaheuristiques pour de l'optimisation multi-objectif (MOEA). Suite à ces contacts, j'ai monté un dossier de financement local (IBNM et DEI UBO) en 2014 pour passer 8 mois comme chercheur invité à Exeter. Mon but était de monter en compétence sur le MOEA et de faire du routage de voiliers de course en optimisant le parcours en fonction de la météo. Ces travaux, en liaison avec Jean-Philippe Babau et Goulven Guillou de l'UBO, n'aboutirent pas immédiatement (ils ne furent finalisés que en 2019 [C17]) mais j'acquis une précieuse expérience auprès de R. Everson et J. Fieldsend sur la thématique MOEA. Ce séjour fut fructueux car il me permit par ailleurs, fort de mon expérience en modélisation MILP acquise durant la thèse d'Asma, d'entamer un travail de recherche avec une membre du CEMPS d'Exeter, C. Lin, sur la thématique de la simulation de processus biologiques. Ces travaux furent ensuite poursuivis après mon retour et donnèrent lieu à publication en conférences [C08] et en revue [R06].

Mes 3 thématiques de recherche principales, Graphes, MILP et MOEA, toutes liées à l'optimisation m'ont ensuite permis de développer des collaborations sur des projets et des encadrements de thèse à partir de 2014, en enrichissant et parfois réorientant les travaux avec mon savoir-faire. Avec Frank Singhoff, nous avons redéfini un sujet de thèse pour une doctorante tunisienne, Rahma Bouaziz [22], puis en 2018 obtenu un financement pour Ill-Ham Atchadam [7]. Les 2 thèses ont pour domaine l'ordonnancement de processus temps réel critiques. J'ai aussi apporté mon expertise lors de collaborations avec Jalil Boukhobza sur des thématiques applicatives stockage (thèse de Djillali Bouthekef [23]), Fog/Cloud (thèse de Islam Naas [114]), Cloud (thèse de Amina Chikhaoui [31]), mémoire non volatiles [R02]. Enfin, la thématique routage de bateau, exploitée en propre avec Jean-Philippe Babau et Goulven Guillou [C17], m'a donné la légitimité pour superviser la thèse de Hand Ouelmokhtar [116] en collaboration avec l'UBS à Lorient et l'UMBB à Bourmedès, Algérie, sur du routage de drones, là encore en multi-objectif.

INTRODUCTION

Dans de nombreuses situations, le temps joue un rôle crucial dans un problème à résoudre. La notion de temps s'entend ici à plusieurs niveaux : (i) Comme un critère de choix, (ii) au niveau du temps nécessaire ou disponible pour effectuer les calculs liés à la résolution du problème et (iii) en termes de ressources humaines pour la modélisation du problème à résoudre et de la définition d'une méthode de résolution.

Pour le premier aspect, le temps est souvent un facteur de choix dans la manière de réaliser une tâche ou de mettre en œuvre une stratégie : pour gagner une course, pour rejoindre un centre de secours, pour respecter les contraintes temporelles d'un système embarqué temps réel, pour optimiser une latence pour de la lecture/écriture en mémoire, dans le Fog, l'IoT ou le Cloud, pour fournir une qualité de service pour du streaming, pour respecter un délai de traitement pour de la détection d'intrusion, etc. Cependant, le plus souvent, ce temps n'est pas le seul facteur à considérer en termes de contraintes ou de critère d'optimisation mesuré par une fonction objectif. En particulier dans les systèmes embarqués, la limitation des ressources matérielles est également souvent à prendre en compte, avec éventuellement leur coût. Les ressources concernées sont par exemple de la mémoire, de la bande passante ou encore des ressources de calcul dédiées (cores, FPGA, GPU). Divers aspects comme la sécurité, la sûreté, l'usure, le risque, les pénalités de retard, peuvent également s'ajouter. Tout cela fait que les problèmes sont souvent par nature multi-critères, avec plusieurs objectifs et contraintes.

A un second niveau, le temps ou l'effort de calcul est un facteur essentiel pour la résolution des problèmes d'optimisation. Ceux-ci sont souvent complexes, de grande taille par rapport aux capacités de traitement. Des résolutions exactes sont alors impossibles en un temps satisfaisant, et des méthodes de résolution approchée mais plus rapides doivent être mises en œuvre.

Enfin, après les aspects (i) qualitatif et (ii) temps de résolution, le temps consacré à l'étude et la mise en pratique d'une méthode pour le problème posé est limité par les ressources humaines disponibles : pour des problèmes spécifiques, une connaissance métier doit être acquise. Le problème d'optimisation n'est alors qu'une partie d'un problème de recherche plus vaste et les solutions proposées en terme de recherche opérationnelle (RO) doivent être relativement simples à appréhender et à mettre en œuvre du point de vue des spécialistes métier qui apportent ce problème.

Ce document présente mes travaux de recherche, fruit de plusieurs collaborations au sein de mon laboratoire, le Lab-STICC, et avec des équipes extérieures. Les projets présentés ici ne

relèvent pas directement du domaine de la recherche opérationnelle pure, mais les outils de la RO ont contribué à les résoudre. Comme nous le verrons par la suite, beaucoup font appel aux techniques sur les graphes, à la programmation linéaire et/ou aux métaheuristiques de résolution multi-objectif, dont les propriétés permettent de répondre aux problématiques évoquées plus haut. L'essentiel des travaux concerne les systèmes embarqués, à la fois comme sujet d'étude, mais aussi pour les travaux récents et les perspectives comme plate-forme d'exécution en ligne des méthodes proposées.

La problématique générale de l'optimisation combinatoire et de son exploitation qui est à la base de mon travail est tout d'abord présentée. Les projets auxquels j'ai contribué sont ensuite évoqués brièvement, avant le plan de la suite du document.

Résolution de problèmes d'optimisation combinatoire

L'optimisation combinatoire est un vaste domaine [126]. La plupart de mes travaux de recherche sont applicatifs, c'est à dire pour moi liés à la spécialisation des algorithmes ou à la combinaison de techniques pour un domaine métier précis par opposition à un travail de recherche sur les algorithmes et méthodes en eux-mêmes. Nous verrons par la suite que mes contributions ont eu lieu dans des domaines très variés des thématiques mentionnées précédemment (temps réel, stockage et routage pour les systèmes embarqués, ...). Mes contributions relèvent de la problématique décrite ci-après.

Contexte métier Pour chaque problématique, il faut appréhender le domaine métier, définir avec les partenaires quelles sont les données disponibles, ce qu'est une solution (en discrétisant un espace de recherche) et les *critères de choix* d'une ou plusieurs *bonnes* solutions. Il faut modéliser certains critères de choix sous la forme de fonctions objectifs. Il faut aussi prendre en compte le temps imparti pour la résolution du problème (calcul en ligne ou hors ligne, temps acceptable si hors-ligne), la précision désirée pour la ou les solutions (e.g. solution exacte ou approchée, borne qualitative), la plate-forme d'exécution et ses contraintes (e.g. algorithme embarqué ou non, disponibilité d'un solveur MILP, exigence d'intégration à une plate-forme logicielle comme Cheddar [132]).

Solutions Un problème d'optimisation combinatoire associe à un ensemble fini de solutions réalisables une valeur de fonction objectif ou un vecteur objectif dans un contexte d'optimisation multi-objectif (MOO, *Multiple Objective Optimisation*). La manière dont est représentée une solution définit un espace de recherche, contenant l'ensemble des solutions réalisables. Une difficulté est de définir un espace de recherche aussi resserré que possible autour de l'espace des solutions réalisables, pour ne pas effectuer d'efforts de calcul inutiles sur la partie non faisable

de cet espace.

Une solution peut, par exemple, être représentée par :

- des variables de décision d'un MILP [129]
- les données d'un chromosome, dans le cadre de l'utilisation d'algorithmes évolutionnaires [65] ou d'autres métaheuristiques [131]
- un chemin, un partitionnement, une sélection de sommets ou d'arcs résultant d'un calcul sur un graphe [129]

Contraintes et objectifs Suivant le paradigme utilisé, les fonctions objectifs sont plus ou moins simples à extraire. Dans le cas des approches MILP, elles se formulent par la ou les fonctions objectifs et par les contraintes linéaires. La distinction entre les deux n'est parfois pas claire et il faut choisir si tel ou tel aspect du problème est modélisé comme une contrainte ou une fonction de coût. C'est notamment le cas si on veut utiliser directement l'approche MILP (mono-objectif). Si une modélisation via les graphes est exploitée, cela se fait en corrélation avec le ou les algorithmes à utiliser sur le graphe, et donc son résultat. Enfin, pour les approches évolutionnaires, tout est possible : les contraintes peuvent être satisfaites par construction (la représentation chromosomique assure directement la validité des solutions) ou au contraire, l'espace de recherche associé comprend des portions infaisables. Dans ce cas, les (méta)-heuristiques peuvent gérer nativement l'infaisabilité, ou des stratégies de réparation chromosomique peuvent être développées.

Contributions

En tant que membre de l'équipe SHAKER du Lab-STICC, j'ai été amené à collaborer sur des domaines métier liés à l'embarqué que j'ai catégorisés en 3 grandes familles :

- l'exploration d'architectures logicielles liées aux problèmes d'ordonnancement temps réel,
- les problématiques liées au stockage des données, et notamment à leur temps d'accès,
- les problèmes de routage et de réseau.

Comme le montre la figure 1, j'ai participé à des encadrements de thèse et des projets pouvant utiliser mes compétences en matière d'optimisation combinatoire. Je promeuts et popularise auprès de mes collègues l'utilisation de méthodes et outils de Programmation linéaire, notamment pour la formalisation et la résolution exacte de problèmes d'optimisation (MILP). Un second aspect concerne l'utilisation du formalisme et des algorithmes de la théorie des graphes. Un troisième aspect se focalise sur les méthodes méta- et matheuristiques. Enfin, je mets en avant des approches par optimisation multi-objectif pour des problèmes variés.

Les paradigmes mis en œuvre sont principalement liés à la programmation mathématique et à l'utilisation d'algorithmes évolutionnaires. Les deux paradigmes peuvent se décliner soit pour

	Permanents	Doctorants	Publis communes	Période										Outils RO			
				2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	Graphes	MILP
DSE TR	JPB, ST+	Asma M.	C03 C05	Ordonancement								modèle					
	FS, BZ*	Rahma B.	R05 C10 S03					Ordonancement RT				meta x					
	FS, NT	III-Ham A.	C13 C14 W6 W7					Ordo RT et sécurité				meta x					
Stockage	JPB		R04 C08 C04	Qos et streaming								x					
	JB, SR		R02					stockage Flash				x					
	JB, PR+, KB*	Islam N.	R08 S05 C11 C12					Placement Fog				x x adhoc					
	JB, KB*	Djillali B.	R07 S04					Cloud				meta					
	JB, KB*	Amina C.	R03 S08					Fédération de clouds				x meta x					
Routage	PR*		C09 S02 W05 OA1	Localisation d'écoles								x x meta x					
	CL*, RE, DM		R06 C07					Simulation biologique				x x					
	JPB, GG		S06					Routage Voiliers				x x					
	FS, JPD+, YB*	Hand O.	R09 R10									Routage drones x x x					
	FS, MM*		CN05									ROV x x					
	CD	M2 Evan F.	W08 CN03									Medoraaal x x					

FIGURE 1 – Projets de 2011 à 2022, par thème, avec les collègues et doctorants impliqués (ceux marqués de * sont en poste à l'étranger ; ceux marqués d'un + sont hors Lab-STICC ; les autres sont dans SHAKER), ainsi que les publications communes et les outils RO utilisés.

une utilisation en optimisation mono-objectif, soit pour une utilisation en optimisation multi-objectif. Dans tous les cas, la première phase de la collaboration consiste à définir un modèle du problème à traiter, avant d'exploiter celui-ci avec des méthodes plus ou moins complexes de résolution.

Il est parfois possible de résoudre le problème à l'optimal (cf sections 3.2, 3.4, 3.5, 4.2.1, 4.2.2, 4.4). Si une formulation MILP est possible mais non solvable globalement à cause de la taille des instances à traiter ou en présence d'objectifs multiples, il faut alors exploiter astucieusement le modèle pour fournir des solutions approchées via des heuristiques (section 3.5). Enfin, une approche analytique n'est pas toujours possible si les problèmes traités ne passent pas l'échelle (en mono ou en multi-objectif) ou si une formulation mathématique n'est pas disponible (e.g. évaluation par simulation). On a alors recours à des heuristiques ou des métaheuristiques (cf sections 2.3, 2.4, 3.5, 4.2.1, 4.2.2, 4.4).

Approches pour la résolution Le tableau 1 résume quelques aspects des méthodes que j'ai utilisées couramment et mon retour d'expérience sur leur emploi, qui peuvent guider le choix d'une approche.

Les approches MILP sont applicables dans un large ensemble de cas, avec la restriction de pouvoir effectuer une formulation linéaire, même si des méthodes de linéarisation de contraintes existent. Elles permettent bien entendu théoriquement une résolution exacte mais également une résolution approchée et du calcul de bornes par relaxation. En utilisation directe avec un outil de résolution comme CPLEX [136], elles permettent d'obtenir une solution optimale sans effort de mise en œuvre particulier dans le cas mono-objectif. Cependant, leur gros désavantage est leur extensibilité, i.e. la limite de taille des problèmes qu'elles peuvent traiter (en termes de variables

	MILP	Graphes	EA
Applicabilité	++	+	+++
Résolution exacte	+++	++	-
Résolution approchée	+	+++	++
Prouvabilité	+++	+++	-
Extensibilité	-	+++	+++
Facilité de mise en œuvre	+	++	- à +++

TABLE 1 – Quelques aspects positifs et négatifs des grandes familles de méthodes d’optimisation combinatoire

de décision et de contraintes) utilement. Les temps de calcul associés aux instances intéressantes les rendent souvent inexploitable pour résoudre celles-ci, ou les ressources mémoire impliquées sont trop importantes pour ces instances. Le modèle ne peut souvent être exploité directement et des méthodes additionnelles sont requises pour la résolution (voir par exemple Sections 2.2, 3.4, 3.5, 4.2.2).

Les approches basées sur les graphes couplent une représentation sous forme de graphe avec un algorithme de résolution le plus souvent existant, et également très souvent efficace pour les problèmes de grande taille (ou alors des algorithmes de résolution approchée efficaces en temps de calcul sont substituables). La mise en œuvre est plus ou moins complexe suivant la problématique de production du graphe à partir des données d’entrée. Par contre, les algorithmes de calcul sont le plus souvent disponibles sous forme de bibliothèques. La difficulté de leur utilisation réside donc principalement dans la modélisation (voir par exemple Sections 3.2, 3.4, 4.2.1). Les approches MILP et Graphes ont le plus souvent des solutions prouvables (résolution exacte ou bornes).

Enfin, les algorithmes évolutionnaires (EA² en anglais) offrent un grand degré de liberté et permettent notamment de découpler complètement l’évaluation d’une solution du processus d’optimisation. Cela permet entre autres d’utiliser n’importe quelle méthode pour évaluer une solution (par exemple une simulation d’un système déduit à partir de la représentation chromosomique, voir Sections 2.3, 2.4, 3.3). Il existe également une souplesse importante sur les ressources et l’effort de calcul utilisés pour la résolution. L’inconvénient majeur est leur manque de prouvabilité : souvent, il n’y a pas de preuve de convergence et donc de garantie sur la qualité des résultats produits.

2. Evolutionary Algorithm

Plan du document

Le premier chapitre présente brièvement les méthodes et outils utilisés par la suite pour résoudre des problèmes multi-objectifs. Le chapitre suivant s'intéresse au temps considéré comme une contrainte, dans les systèmes temps réel embarqués, qui correspondent aux travaux de thèse de Asma [110], Rahma [22] et Ill-Ham [7] sur l'ordonnancement et l'exploration des architectures logicielles pour ces systèmes. Les problématiques de temps liées à l'usage de ressources de stockage et d'accès aux données sont évoquées ensuite, avec les travaux sur la QoS dans un réseau domestique, la définition de politique de cache pour des mémoires Flash, les travaux de Djillali [23] et Islam [114] sur le stockage dans le Fog et l'IoT, et ceux de Amina [31] pour le placement de données dans un Cloud fédéré. Les travaux sur le temps pour du routage de bateaux ou de drones avec entre autres les travaux de Hand [116], et l'utilisation en ligne des algorithmes, notamment via des plate-formes embarquées font l'objet de la dernière partie sur la présentation de mes travaux. Dans cette partie, la construction de réseaux de transport spécifiques est aussi abordée. Enfin, le chapitre 5 présente un bilan de mes contributions, mes travaux en cours et les axes de recherche vers lesquels je m'oriente.

OPTIMISATION MULTI-OBJECTIF

Nous introduisons brièvement dans ce chapitre les concepts et outils de base de l'optimisation multi-objectif. Des ouvrages de référence présentent notamment les approches évolutionnaires [41, 35] (en français, voir le chapitre dédié dans [131]). Les méthodes MOO recherchent un ensemble de *bonnes* solutions à un problème d'optimisation comportant plusieurs fonctions objectifs pour évaluer ces solutions. La problématique réside dans le fait que ces objectifs sont antagonistes. Optimiser suivant l'un d'entre eux entraîne une dégradation des autres valeurs de fonctions objectifs. La notion de dominance permet de prendre en compte cet aspect et est utilisée dans de nombreuses méthodes. La première section donne les définitions liées à la dominance de Pareto et aux ensembles de solutions associées (ensemble et front de Pareto). Elle présente aussi les outils de mesure comparative des algorithmes MOO entre eux. La section suivante présente les méthodes les plus classiques (résolution en mono-objectif, méthodes de type ϵ -contraintes, méthodes évolutionnaires).

1.1 Définitions et métriques de comparaison

1.1.1 Définitions

Un problème d'optimisation multi-objectif peut se formuler ainsi :

$$\begin{cases} \text{Minimize } F(X) = (f_1(X), f_2(X), \dots, f_k(X)), \text{ avec } X = (x_1, x_2, \dots, x_n) \text{ et } k \geq 2 \\ g_j(X) \geq 0, \text{ avec } j \in \{1, \dots, m\} \end{cases} \quad (1.1)$$

$X = (x_1, x_2, \dots, x_n)$ est le vecteur de décision, représentant une solution faisable dans l'espace de recherche/décision \mathcal{D} , défini par l'ensemble des contraintes $g_j(X)$. $F(X)$ est la fonction à optimiser, avec k composantes, à valeurs dans l'espace objectif \mathcal{O} . Sans perte de généralité, et sauf mention contraire, il s'agit toujours d'une minimisation dans ce document.

Les méthodes de sélection de solutions se basent pour la plupart sur la notion de dominance de Pareto [35]

Définition 1. (*dominance de Pareto*) Une solution X_1 domine une solution X_2 (i.e. $X_1 \preceq X_2$), ssi $\forall i \in \{1, \dots, k\}, f_i(X_1) \leq f_i(X_2)$, et $\exists j \in \{1, \dots, k\}$ t.q. $f_j(X_1) < f_j(X_2)$.

On en déduit la propriété de non-dominance

Définition 2. (Solution non dominée) Une solution $X_1 \in \mathcal{D}$ est non dominée ssi $\nexists X_2 \in \mathcal{D}$ t.q. $X_2 \preceq X_1$

Les solutions dominées ne sont pas intéressantes du point de vue de l'optimisation, car il en existe d'autres meilleures sur tous les objectifs d'optimisation et strictement meilleures sur au moins l'un d'entre eux. Les solutions non dominées sont celles qui représentent les bons compromis recherchés car elles optimisent au moins l'une des fonctions objectifs en minimisant le coût pour les autres fonctions.

Définition 3. (ensemble et front de Pareto) L'ensemble des solutions non dominées de \mathcal{D} est l'ensemble de Pareto, noté \mathcal{PS}^* et son image dans l'espace objectif \mathcal{O} est le front de Pareto \mathcal{PF}^* .

Définition 4. (point de Pareto) Un élément p du front de Pareto est un point de Pareto, $p \in \mathcal{PF}^*$.

Notons que dans la suite de ce document, on utilise parfois le vocable ensemble (resp. front) de Pareto pour désigner un ensemble de solutions non dominantes entre elles (resp. leur image dans \mathcal{O}), et de ensemble/front de Pareto optimal pour désigner $\mathcal{PS}^*/\mathcal{PF}^*$. De même, même si plusieurs solutions de \mathcal{D} ont le même point de Pareto, nous considérons comme complet (i.e. optimal) un ensemble de Pareto comportant au moins une de ces solutions (ensemble minimum [35]).

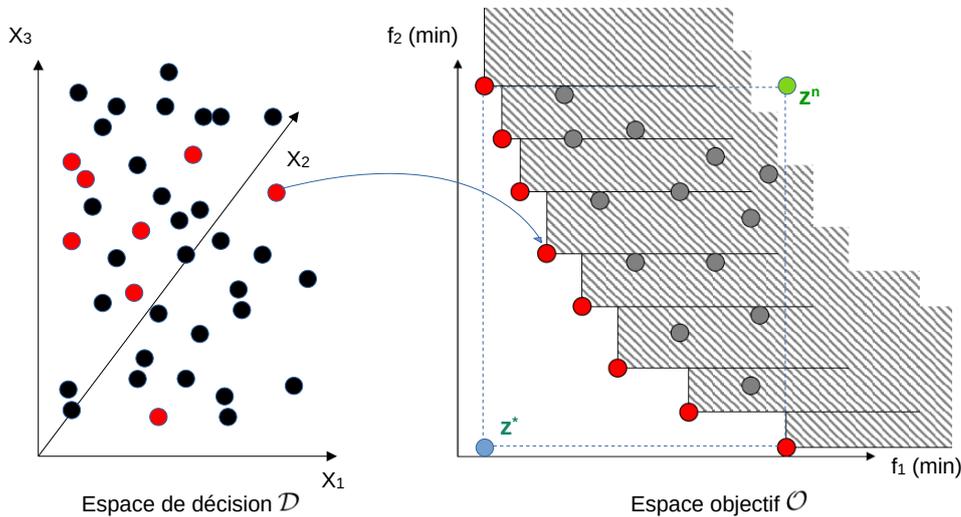


FIGURE 1.1 – Relation de dominance et ensemble/front de Pareto

Sur la figure 1.1, pour un espace de décision à $n = 3$ dimensions et un problème bi-objectif ($k = 2$), les points rouges représentent, dans l'espace décisionnel (à gauche) et l'espace objectif

(à droite) l'ensemble \mathcal{PS}^* et le front de Pareto \mathcal{PF}^* , avec pour ce dernier des points dominés en gris. Deux points de référence, le point idéal z^* et le point nadir z^n sont également représentés. Le premier correspond à l'ensemble des valeurs optimales ($\forall i \in 1, \dots, k, z_i^n = \min_{X \in \mathcal{D}} f_i(X)$), le second correspond aux pires valeurs de fonctions objectifs ($\forall i \in 1, \dots, k, z_i^n = \max_{X \in \mathcal{D}} f_i(X)$). Le point théorique z^n est dominé par l'ensemble des points de \mathcal{O} .

Une des méthodes de calcul les plus simples pour trouver un point du front de Pareto consiste à agréger les fonctions objectifs en une fonction unique $f_0(X) = \sum_{i=0}^n w_i \cdot f_i(X)$, en attribuant un poids spécifique w_i à chaque fonction. La solution optimale $X_0^* \in \mathcal{D}$ de la minimisation de $f_0(X)$ est une solution non dominée *supportée*.

Définition 5. (*solution supportée*) Une solution $X_0^* \in \mathcal{D}$ est supportée ssi $\exists W = \{w_1, w_2, \dots, w_k\}$ t.q. $\forall X \in \mathcal{D}, \sum_{i=0}^k w_i \cdot f_i(X_0^*) \leq \sum_{i=0}^k w_i \cdot f_i(X)$

L'ensemble des points de \mathcal{PF}^* correspond à des solutions (non dominées) supportées si le front de Pareto est convexe comme c'est le cas sur la figure 1.1, mais certaines d'entre elles ne le sont pas si le front n'est pas convexe [35]. Le front exact n'est alors pas calculable directement par une méthode par agrégation faisant varier les poids w_i .

1.1.2 Métriques de comparaison

Des méthodes exactes calculant \mathcal{PS}^* et des heuristiques fournissant des ensembles et fronts approchés cohabitent. De nombreuses métriques existent [35, 8] pour mesurer la qualité des ensembles de Pareto fournis par ces algorithmes et pour les comparer. Les aspects classiquement capturés par ces métriques sont la convergence (proximité des points avec le front de Pareto exact), la diversité (les points s'étalent-ils sur l'ensemble des valeurs possibles pour les fonctions objectifs) et la taille de ces fronts en nombre d'éléments.

Certaines métriques ne nécessitent pas la connaissance de \mathcal{PF}^* et d'autres le requièrent. Nous en avons utilisé une de chaque type dans les travaux décrits dans ce document.

La première métrique est l'hypervolume (aussi nommée S-métrique), proposé par [155]. Elle ne nécessite pas la connaissance du front de Pareto, mais implique la définition d'un point de référence z_{ref} (par exemple le point nadir si il est connu) dominé par l'ensemble des points de l'ensemble \mathcal{P} . Si on veut comparer qualitativement 2 ensembles \mathcal{P}_1 et \mathcal{P}_2 , alors z_{ref} doit être dominé par l'ensemble de tous les points. L'hypervolume correspond au volume compris entre le point z_{ref} et les points de \mathcal{P} . Mathématiquement, pour un problème à k objectifs, $HV(\mathcal{P}, z_{ref}) = \lambda_k \cup_{z \in \mathcal{P}} ([z; z_{ref}])$, avec λ_k la k -mesure dimensionnelle.

La seconde métrique est l'IGD (Inverted Generational Distance, [35]). Elle mesure la distance d_i de chaque point de \mathcal{PF}^* au point de l'ensemble évalué \mathcal{P} le plus proche, et moyenne les valeurs

obtenues.

$$IGD(\mathcal{P}, \mathcal{PF}^*) = \frac{(\sum_{i=1}^{|\mathcal{PF}^*|} d_i^p)^{1/p}}{|\mathcal{PF}^*|} \quad (1.2)$$

L'IGD pénalise les ensembles éloignés du front exact et/ou ne couvrant pas suffisamment ce front.

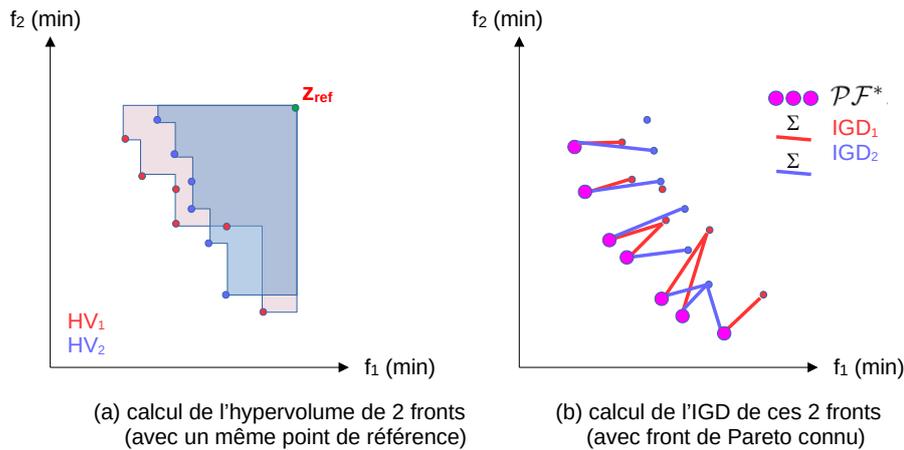


FIGURE 1.2 – Comparaison de 2 fronts bi-objectifs avec 2 métriques (a) HV et (b) IGD

La figure 1.2 illustre les 2 métriques. Notez que les calculs de métriques sont le plus souvent associés à une normalisation des valeurs de fonction objectif, pour ne pas biaiser les comparaisons.

1.2 Méthodes de résolution

La résolution de problèmes multi-objectifs peut se faire suivant plusieurs critères :

- rechercher une seule solution (*Single Objective Optimization*, SOO) ou un ensemble de solutions intéressantes (MOO). Dans le premier cas, on peut soit calculer une seule solution, en mono-objectif, soit utiliser un outil de choix parmi un ensemble de solutions déterminées par l'algorithme. Dans le premier cas, le décideur peut ainsi indiquer ses préférences avant le calcul en biaisant la recherche mono-objectif vers une des fonctions objectifs. Dans le second cas, il peut influencer le choix a posteriori, lors du processus de décision parmi l'ensemble des solutions produites lors du processus d'optimisation multi-objectif. Notez que des algorithmes d'aide à la décision comme Topsis [77] ou Rmethod [123] peuvent aussi être utilisés, en restant neutres par rapport aux objectifs.
- utiliser ou non le concept de dominance pour la détermination des solutions intéressantes.

La plupart des algorithmes le font, mais pas tous (voir par exemple VEGA [37]). Dans le premier cas, il est alors possible de rechercher un front de Pareto exact ou un front approché.

Les grands domaines de la Recherche Opérationnelle sont associés à des travaux sur l'optimisation multi-objectif.

- En théorie des graphes : pour les problèmes de cheminement dans un graphe, des méthodes spécifiques ont pu être développées par extension des algorithmes utilisés dans le cas SOO. L'algorithme de Martins [108] utilise la dominance pour calculer le front de Pareto en se basant sur les algorithmes à label comme Dijkstra. Un plus court chemin multi-objectif correspond au fait d'attribuer plusieurs labels à chaque sommet, la fonction objective associée à un chemin correspondant à une somme sur une catégorie de labels. Des variantes existent pour calculer à la fois un chemin de longueur minimale sur une série de pondérations et un chemin de capacité maximale pour une autre série de labels [60]. Des plus courts chemins avec pondérations variables au cours du temps peuvent aussi être considérés [142][30].
- En Programmation Linéaire Mixte (MILP) : de manière générale, des méthodes d'énumération implicite simple, ou par séparation et évaluation sont également adaptables, notamment dans le cadre de la programmation linéaire multi-objectif, avec des techniques disponibles parfois réduites au cas bi-objectif [80]. Des outils comme vOptsolver implantent plusieurs méthodes [62] (parfois dédiées à un problème particulier) et quelques implantations d'algorithmes spécifiques sont disponibles comme AIRA [117].
- Avec des métaheuristiques MOEA : des approches métaheuristiques ont été développées notamment pour résoudre des instances de problèmes de grande taille, comme dans le cas SOO. Des algorithmes correspondant aux différentes familles de métaheuristiques (algorithmes évolutionnaires, de coopération d'essaims, par analogie physique, etc) existent. Des ouvrages comme [35] en détaillent les particularités. Des libraires comme MOEA Framework [59], jMetal [46], PISA [21], ParadisEO-MOEO [98] ou pyMOO [20] en implémentent les plus courants et permettent de tester différents MOEA.

Les alternatives sont nombreuses pour le choix d'un algorithme multi-objectif pour une application donnée. Il n'y a pas de solution universelle au problème du choix de l'algorithme (*no free lunch theorem* de Wolpert [147]). L'optimisation multi-objectif se heurte aux mêmes problématiques que dans le cas mono-objectif, en cherchant à traiter des problèmes de grande taille pour lesquels il n'existe pas d'approche efficace connue. Il faut de plus garder en tête que si c'est un ensemble de solutions qui sont recherchées (ensemble de Pareto ou approximation), le nombre d'évaluations de solutions effectuées est démultiplié par rapport à une version mono-objectif du même problème (voir la section ci-dessous sur les approches mono-objectifs à résultats multiples). De plus, par nature, plusieurs fonctions objectives sont à évaluer pour une même solution. Les

approches coûteuses en terme d'évaluation des solutions (par ex. avec une formulation MILP complexe ou des simulations, cf par ex. Section 3.3) devront être plus tôt remplacées par des approches approximatives moins coûteuses lorsque la taille des instances traitées augmente (par exemple avec une approche MOEA).

Les approches MOEA sont largement répandues dans des domaines variés, comme le montre d'ailleurs les applications pour lesquelles nous présentons nos travaux dans ce document. Elles requièrent cependant des points d'attention :

- Comme nous l'avons déjà mentionné, nombre d'entre elles ont été développées. Comme mentionné dans [45], il n'y a pas de résultats théoriques sur la convergence exploitables en pratique pour guider le choix d'une méthode.
- le nombre d'objectifs couramment utilisé est de 2 ou 3, mais des approches spécifiques existent pour l'optimisation *many-objectives*. Par exemple, NSGA III présenté ci-après ambitionne de traiter jusqu'à 15 objectifs. Des méthodes de réduction du nombre d'objectifs sont aussi explorées [103], avec une problématique similaire à celle de sélection des caractéristiques (*features*) en apprentissage.
- Comme dans le cas mono-objectif, les algorithmes évolutionnaires nécessitent un encodage soigneusement choisi des solutions pour réduire l'espace de recherche au maximum en le restreignant dans l'idéal à l'espace réalisable des solutions. Par exemple, dans la section 2.3, un encodage non redondant des solutions est proposé. Dans la section 4.4, la taille de la représentation est automatiquement ajustée à l'instance traitée.
- En liaison avec le point précédent, les fonctions de voisinage doivent permettre d'explorer efficacement l'espace de recherche, en restant si possible dans l'espace réalisable. Par exemple, dans la section 2.4, la fonction de voisinage est modifiée en cours d'exécution, avec un espace de recherche agrandi. Dans la section 3.5, un opérateur standard de croisement est utilisé et une fonction de *réparation* permet de rendre réalisables les solutions éventuellement infaisables résultant du croisement.
- Pour nous, la métaheuristique MOEA choisie n'est pas si importante, l'encodage et les opérateurs de voisinage sont prépondérants. Une métaheuristique peut être substituée à une autre. La validation d'une approche est souvent réalisée par des tests comparatifs (voir par exemple la section 2.4 pour des tests paramétriques ou une comparaison PAES *vs* NSGA II dans la section 4.4). L'importance des opérateurs est par exemple soulignée dans [45]. Dans le panorama historique [134], les auteurs soulignent aussi l'importance des progrès à réaliser en matière théorique sur le domaine des métaheuristicues par rapport au fait d'inventer de nouvelles métaheuristicues basées sur des processus variés.

D'autres aspects, comme la robustesse (persistance de la validité des solutions lors d'une variation dans l'environnement ou elle est appliquée), le budget (effort de calcul) ou l'hybridation des méthodes sont également des aspects à considérer, comme c'est le cas dans certaines des

applications présentées dans ce document.

Nous présentons dans cette section les approches les plus utilisées par la suite. Nous détaillons tout d'abord les approches SOO pour la résolution puis les approches MOO basées sur la dominance de Pareto.

1.2.1 Méthodes mono-objectifs à résultat unique

L'approche la plus simple est la méthode par pondération, qui agrège les valeurs des fonctions objectifs en une fonction pondérée unique.

$$f(x) = \sum_{k=1}^n \alpha_k \cdot f_k(x) \quad n \geq 2, \forall 1 \leq k \leq n, \alpha \in [0..1]$$

La pondération peut être équilibrée ou biaisée en fonction des choix du décideur. Il est à noter qu'une normalisation des $f_k()$ est souvent mise en œuvre conjointement, pour éviter un biais non maîtrisé.

Une variante de ce type d'approche est le *Goal Programming*. La qualité d'une solution est évaluée par rapport à des objectifs de référence (valeurs cible T_k) pour chacune des fonctions objectifs.

$$f(x) = \sum_{k=1}^n |T_k - f_k(x)| \quad n \geq 2, \forall 1 \leq k \leq n$$

Là aussi une pondération et une normalisation des termes est possible pour biaiser les résultats vers un des objectifs.

Un autre type d'approche, les méthodes lexicographiques, ordonnent les objectifs par ordre de priorité : l'optimisation porte tout d'abord sur la fonction objectif la plus prioritaire, puis l'optimisation se fait en fonction du deuxième objectif, en intégrant sous forme de contrainte les résultats obtenus lors de la première itération. Le processus s'achève lorsque une solution unique concernant la fonction objectif courante est obtenue, ou que la dernière des fonctions objectifs a été optimisée.

1.2.2 Approches mono-objectifs à résultats multiples

Il est possible d'englober les méthodes précédentes dans un processus de recherche qui permet de définir un ensemble de solutions intéressantes, pas forcément Pareto optimales.

La méthode de la somme pondérée peut être appliquée en faisant varier les poids utilisés : la méthode peut alors permettre de détecter toutes les solutions supportées d'un ensemble de Pareto (voir déf. 5). La première phase de la méthode 2-phases de Ulungu et Teghem [139] permet de trouver ainsi toutes les solutions supportées en procédant par dichotomie sur l'espace

objectif. Une seconde façon de procéder est de rechercher une solution optimisant un objectif, puis d'ajouter une contrainte excluant cette solution sans en exclure d'autres, et d'itérer le processus. On peut aussi découper l'espace objectif en sous-espaces, en bornant par exemple toutes les fonctions objectifs sauf une par l'ajout de contraintes. On recherche itérativement pour chacun des sous-espaces ainsi créé un optimal sur la dernière fonction objectif. Ces méthodes de type ϵ -constraint sont notamment utilisables avec des approches par programmation linéaire [28], où l'ajout de contraintes d'intervalles se fait aisément pour des valeurs de fonctions objectifs entières. La figure 1.3 illustre le processus dans le cas de 2 objectifs. Les extrema sont d'abord calculés (étapes (a), (b)). L'espace est ensuite découpé suivant une valeur Δ choisie (étape (b)). Puis pour chaque portion $[\epsilon_\alpha \dots \epsilon_{\alpha+1}]$ de l'espace, un point de Pareto est ensuite calculé (étape (c)). A chaque fois, une des coordonnées du point est calculée en optimisant une des fonctions objectifs (voir étape (a), point $\textcircled{1}$). La seconde coordonnée du point est ensuite déterminée en optimisant la seconde fonction objectif, avec une contrainte ajoutée pour que la valeur de la première coordonnée reste optimale (voir étape (a), point $\textcircled{2}$).

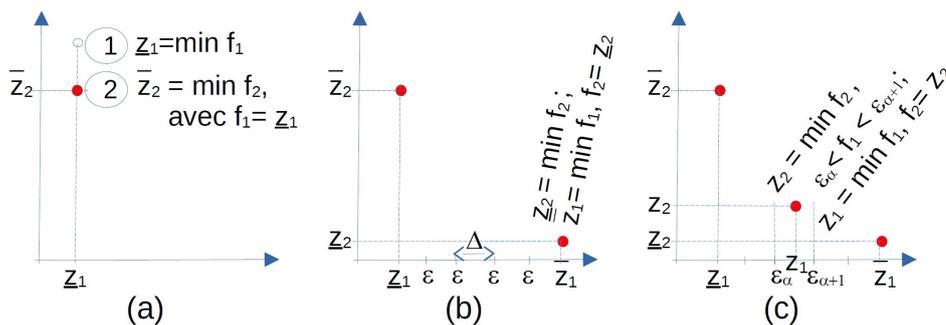


FIGURE 1.3 – Une approche ϵ -constraint pour le calcul d'un ensemble de solutions par découpage de l'espace objectif (problème bi-objectif)

Un des premiers algorithmes évolutionnaires prenant en compte la dimension multi-objectif est VEGA (Vector Evaluated Genetic Algorithm, [37]). C'est un algorithme génétique classique [65], mais lors de la sélection pour la génération suivante, des sous populations distinctes sont générées. Chaque sous-partie des individus de la population est sélectionnée en utilisant pour leur évaluation une fonction objectif différente. La population est ensuite utilisée globalement pour l'application des opérateurs génétiques. Notez que contrairement aux métaheuristiques présentées par la suite, la dominance de Pareto n'est pas utilisée ici. L'inconvénient de la méthode réside dans le fait que des solutions présentant des valeurs intéressantes sur l'ensemble des fonctions objectifs risquent de ne pas émerger car elles sont surclassées dans chaque sous-population par des éléments très performants uniquement sur la fonction associée à cette population.

1.2.3 Approches évolutionnaires multi-objectifs

Potentiellement toutes les classes de métaheuristiques possèdent une version multi-objectif, e.g. MOPSO [113, 91] pour les essaims de particules, MOSA [137] pour le recuit simulé, etc. Les algorithmes récents peuvent inclure directement leur version multi-objectif (e.g. optimisation basée sur le comportement d'une meute et de son leader, le Path Finder Algorithm [150]). Les algorithmes MOEA (Multi-Objective Evolution-based Algorithm) se basent principalement sur les algorithmes génétiques et leurs mécanismes d'exploration, en tenant compte dans la gestion des solutions des relations de dominance existant entre ces solutions. En plus de viser à fournir des solutions proches du front de Pareto, c'est à dire de rechercher la convergence vers l'optimalité pour les fonctions objectifs, elles doivent aussi inclure un mécanisme de diversification pour garantir la bonne répartition de ces solutions le long du front. Les algorithmes les plus connus développés dans les années 2000, sont, entre autres, SPEA2 [154], IBEA [153], SMS-EMOA [48], ou encore MOEA/D [152]. Nous présentons brièvement deux d'entre eux, Pareto Archived Evolution Strategy [87] et NSGA II [42] que nous avons utilisés. Le premier manipule une seule solution et stocke dans une archive externe des solutions non dominées entre elles. La diversité est assurée en répartissant les solutions dans une grille dont le maillage est calculé dynamiquement. Le second manipule une population et trie les solutions suivant leur degré de dominance. Les derniers individus sélectionnés le sont sur la densité de solutions qui leur sont similaires en termes de valeurs de fonction objectifs (i.e. par analogie, dans la même niche écologique).

Pareto Archived Evolution Strategy

PAES a été proposé en 2000 par J. Knowles et D. Corne. Il s'agit d'un algorithme MOEA dont la population est réduite à un individu. Il stocke dans une archive de taille bornée les solutions non dominées entre elles qu'il veut préserver. Il itère sur sa solution courante en lui appliquant des opérations de type mutation/exploration locale (algorithme 1 + 1). Ce schéma présente l'avantage d'économiser à chaque génération le nombre d'évaluations faites avant les opérations de sélection, ce qui peut s'avérer intéressant quand ces évaluations sont coûteuses. Comme le montre la figure 1.4, l'archive est mise à jour à chaque étape : une nouvelle solution est choisie en fonction de la dominance. Pour les solutions non dominées entre elles, l'isolement de ces solutions dans l'espace objectif sert à les départager.

L'isolement de chaque solution est calculé par le nombre de solutions situés dans la même zone de pavage de l'espace objectif. Le pavage lui-même est calculé dynamiquement, en divisant récursivement (suivant une profondeur fixée) l'espace pour chaque dimension sur l'intervalle des valeurs présentes dans l'archive pour cette dimension, et en affectant à chaque fois la solution du bon côté en fonction de la valeur de fonction objectif considérée. Associé à un encodage binaire, cela permet de recalculer les pavés et leur contenu de manière efficace. Cela permet de s'affranchir du choix d'une taille de niche pour le calcul de l'encombrement/isolement de chaque

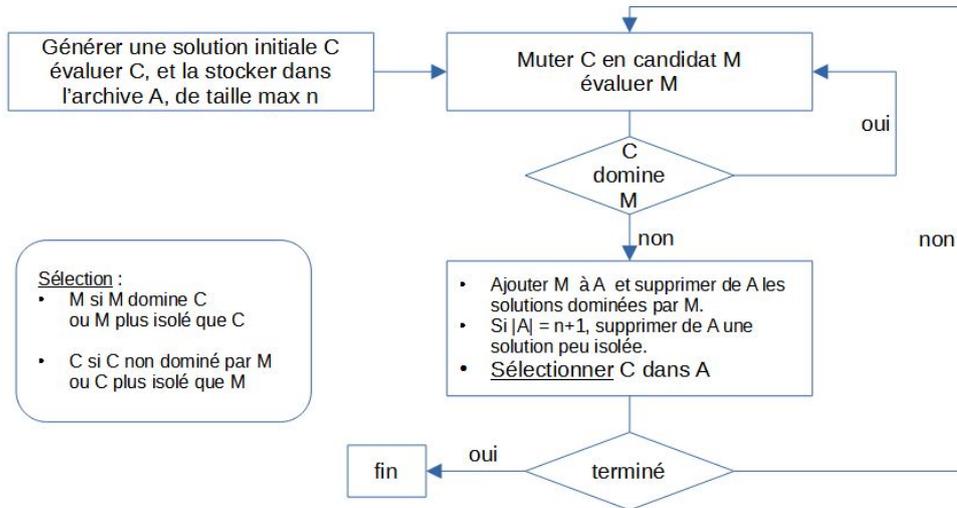


FIGURE 1.4 – Principe de l’algorithme PAES

solution.

Avec une archive de taille n et g itérations, la complexité de l’algorithme est de $\mathcal{O}(gn)$.

NSGA II

NSGA II (proposé par K. Deb en 2002) est un algorithme évolutionnaire élitiste, basé sur la dominance de Pareto pour prioriser la sélection lors d’un passage d’une génération à l’autre de l’algorithme. L’algorithme reprend l’idée de classement proposée par D. E. Goldberg [65] : les individus sont classés par rang de dominance pour constituer une série de fronts F_1, F_2, \dots, F_l . F_1 contient les solutions non dominées à l’intérieur de la population, F_2 regroupe celles qui deviennent si les éléments de F_1 ne sont pas considérés, etc. La figure 1.5 illustre ce classement.

Le fonctionnement de l’algorithme est décrit sur la figure 1.6 : à chaque itération, après génération d’une descendance (qui augmente la taille de la population), celle-ci est classée par rang. Les premiers rangs sont sélectionnés pour la génération suivante, à concurrence de la taille de la population initiale. Pour le premier rang qui ne peut être complètement inclus dans la génération suivante, la sous partie des individus de la population sélectionnés l’est sur deux critères : (i) les individus correspondant à un extremum sur une des fonctions objectifs sont sélectionnés (élitisme) et (ii) les individus les plus isolés dans l’espace objectif sont également prioritaires. L’isolement est mesuré par la distance d’encombrement via une somme sur les objectifs des distances aux deux individus encadrant chaque solution pour l’objectif. Les distances sont normalisées pour éviter tout biais.

NSGA II est un des algorithmes les plus populaires (voir par exemple [144] pour un survey sur son utilisation), avec une complexité (hors opérateurs génétiques) en $\mathcal{O}(gm^2)$, pour g itérations

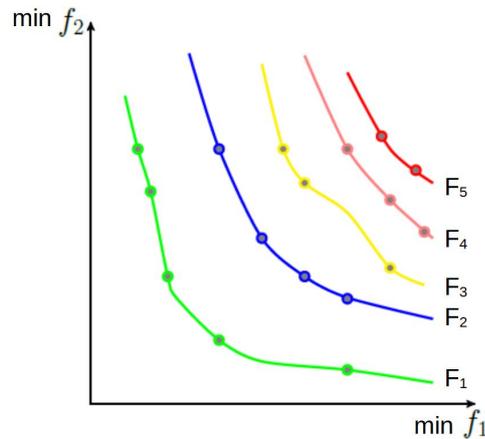


FIGURE 1.5 – Classement par rang de dominance effectué par NSGA II (d’après [31]) pour 2 objectifs à minimiser

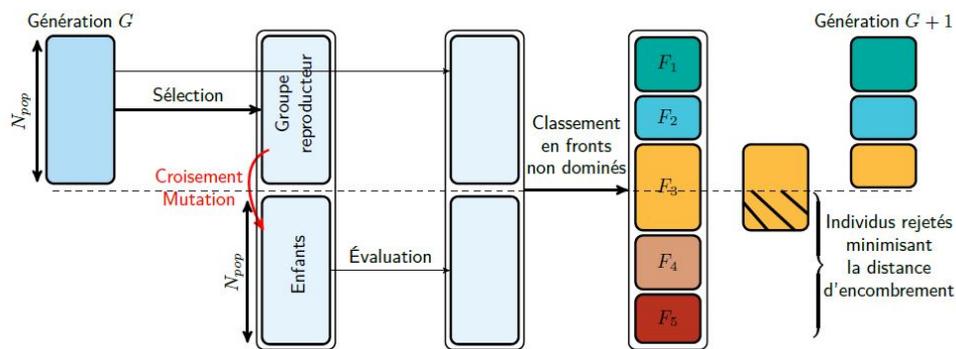


FIGURE 1.6 – Déroulement de l’algorithme NSGA II ([31, 92])

et une population de taille m . Une version NSGA III [40] a été proposée par K. Deb et H. Jain en 2014 pour traiter la problématique *many-objectives*, pour le cas où le nombre de fonctions objectifs dépasse 3 (les auteurs envisagent jusqu’à 15 objectifs). Plus le nombre de fonctions objectifs augmente, plus la proportion de solutions non dominées est importante. En plus des problèmes de représentation graphique d’un espace à plusieurs dimensions, cela accroît les temps de calcul pour les mesures de diversité des solutions et pour les métriques de performance comme l’hypervolume. De plus, les opérateurs d’exploration peuvent s’avérer inefficaces pour converger vers le front de Pareto. Pour préserver la diversité et maximiser l’efficacité du processus de recherche, les auteurs définissent des points de référence uniformément répartis sur l’espace objectif et favorisent la recherche autour de ces points. Cela permet de préserver la diversité des solutions et d’améliorer l’efficacité des recombinaisons entre solutions sélectionnées pour leur proximité avec le même point de référence.

1.3 Conclusion

Nous avons présenté dans ce chapitre les principes et méthodes de base de l'optimisation multi-objectif, en nous focalisant sur les notions et approches que nous avons utilisé par la suite. Les méthodes sont très nombreuses et variées, mais principalement basées sur la recherche d'un ensemble et d'un front de Pareto. Elles adaptent le plus souvent les méthodes du domaine mono-objectif, notamment pour la recherche d'un ensemble de solutions supportées. Les problématiques de modélisation sont donc semblables, même si le résultat produit est plus variable : on peut s'intéresser à produire une solution unique, compromis satisfaisant, ou souhaiter générer un front complet. La décision sur la solution choisie se fait alors de manière externe. Cela est intéressant lorsque certains aspects du choix ne sont pas pris en compte dans le modèle. Par exemple, pour le choix d'un compromis *temps d'arrivée/risques pris* lors d'une compétition de voile, le choix peut être de minimiser le risque si vous êtes en tête de course ou au contraire de prendre une route plus rapide mais plus risquée si vous êtes en seconde position, afin d'espérer gagner l'épreuve.

Comme nous l'avons évoqué dans l'introduction, les méthodes MOO permettent, en plus d'un objectif de temps (de parcours, de réponse, etc), de considérer simultanément d'autres fonctions objectifs. Les méthodes disponibles permettent de la résolution exacte (au sens du calcul d'un ensemble ou d'un front de Pareto) ou la production d'ensembles approchés, en ajustant l'effort de calcul. A ce propos, même si les outils et approches liés au parallélisme n'ont pas été abordés ici, nous avons tout de même développé une méthode maître-esclave spécifique (paradigme maître-esclave, parallélisme au niveau des itérations, avec des évaluations asynchrones, d'après la classification proposée dans le survey récent [51]). Cette méthode a été utilisée en la couplant avec PAES dans les travaux présentés Sections 2.3 et 4.2.2.

EXPLORATION D'ARCHITECTURES POUR LES SYSTÈMES EMBARQUÉS TEMPS RÉEL

Parmi les systèmes pour lesquels le temps joue un rôle crucial, les systèmes temps réel sont ceux pour lesquels les processus de calcul doivent fournir leurs résultats dans des délais impartis. Ces contraintes sont imposées par l'environnement et leur violation peut potentiellement entraîner des conséquences parfois catastrophiques, comme par exemple pour les systèmes de contrôle-commande avioniques.

Une première section est consacrée à une présentation succincte du domaine de l'exploration d'architectures temps réel embarquée (DSE pour *Design Space Exploration*). Les 3 parties suivantes présentent nos travaux d'exploration d'architectures logicielles temps réel, avec des contraintes et objectifs en termes de caractéristiques liées au temps, mais aussi pour des aspects non fonctionnels comme la sécurité.

2.1 Problématique

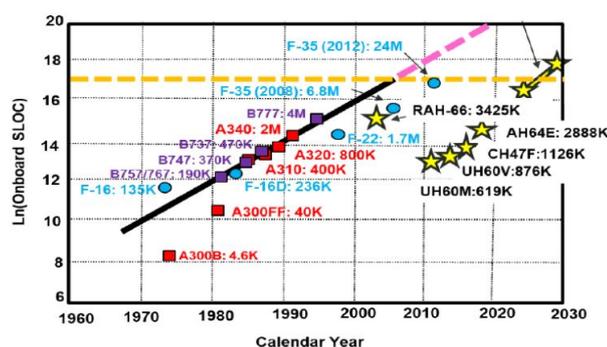


FIGURE 2.1 – Evolution du nombre de lignes de code embarquées à l'horizon 2030 dans les avions civils et militaires [26]

Comme le montre la fig. 2.1, les systèmes embarqués temps réel intègrent de plus en plus de

fonctionnalités, correspondant à plusieurs millions de lignes de code dans les avions actuels. Ces systèmes sont soumis à de nombreuses exigences qui doivent être prises en compte durant leur conception. De nombreuses alternatives sont possibles pour définir une architecture logicielle implantant ces calculs embarqués.

La conception d'une architecture logicielle prend en compte à la fois les composants logiciels à concevoir, mais aussi la plate-forme d'exécution matérielle (composants de calcul, bus, mémoire, cache, etc). Les grandes étapes de la construction d'une architecture sont résumées dans la figure 2.2.

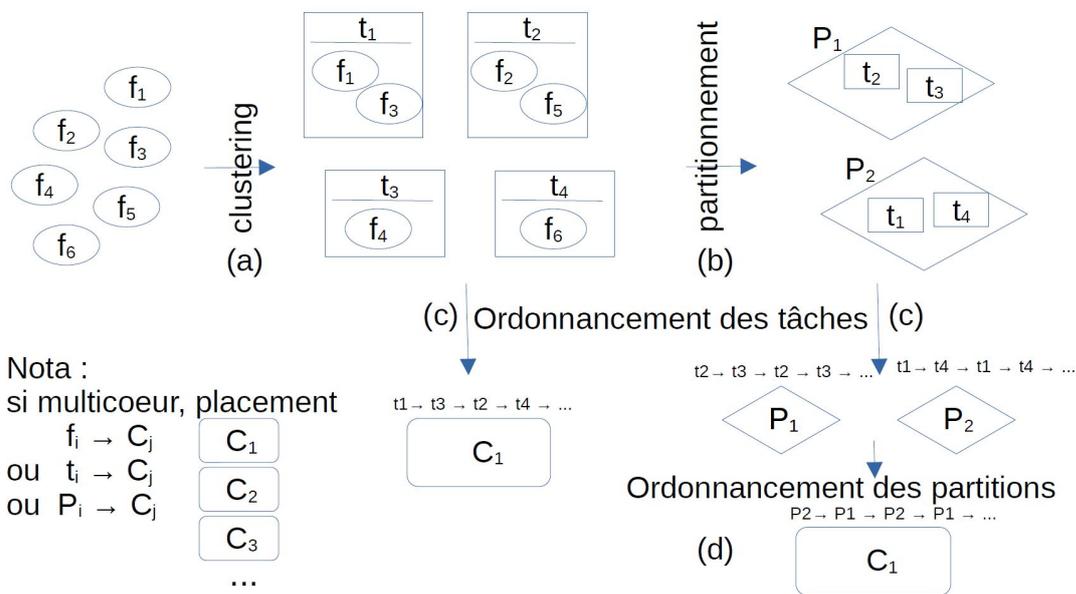


FIGURE 2.2 – Les grandes étapes de la construction d'une architecture temps réel

Les fonctionnalités se répartissent en fonctions, regroupées dans des tâches (Fig. 2.2(a), clustering), elles-mêmes éventuellement réparties dans des partitions (b). Ces tâches et partitions s'exécutent soit périodiquement, soit sur évènement, soit de manière sporadique. Dans le cas périodique, un ordonnancement (c) statique ou dynamique est possible pour exécuter les tâches construites sur une ressource de calcul donnée ou dans une partition. Si plusieurs partitions existent, l'ordonnancement (c) est réalisé partition par partition et un ordonnancement statique des partitions est également produit (d). Si l'ordonnancement des tâches est statique, leur séquence d'exécution est prédéfini. Dans le cas d'un ordonnanceur dynamique, celui-ci met en œuvre une politique d'ordonnancement qui décide de la prochaine tâche à exécuter parmi celles qui sont prêtes. Le choix se fait en fonction de la priorité associée à la tâche. Cette priorité peut être définie statiquement (ex : priorité forte pour les tâches ayant une fréquence d'exécution élevée avec l'algorithme Rate Monotic, RM) ou dynamiquement (ex : tâche dont l'échéance est

la plus courte avec Early Deadline First, EDF) [101]. De plus quand une nouvelle tâche éligible se présente, l'algorithme peut préempter ou non la ressource de calcul. Le statut des tâches (éligible à l'exécution ou non) dépend de leur périodicité d'exécution et du fait qu'elles peuvent être bloquées en attente d'une ressource ou d'un évènement (dépendances entre tâches), ou encore suspendues si la préemption par une tâche plus prioritaire est autorisée. Enfin, même avec un ordonnancement préemptif, certaines tâches peuvent être considérées comme critiques, et ne doivent pas rater leur échéance, alors que d'autres avec des contraintes d'échéance plus souples, sont autorisées à le faire.

Pour calculer un ordonnancement, des méthodes analytiques et par simulation ont été développées. Une fois un ordonnancement construit, les méthodes d'analyse d'ordonnancement permettent de vérifier que les propriétés temps réel attendues sont respectées.

Les méthodes analytiques peuvent permettre de vérifier des conditions nécessaires et/ou suffisantes pour la faisabilité d'un ordonnancement (par ex : règle de Liu et Leyland sur le taux d'occupation d'un processeur [101]). En posant des restrictions (par ex : priorités fixes), il est possible d'aller plus loin, jusqu'à calculer complètement le déploiement d'un ensemble de fonctions regroupées en tâches sur un ensemble de ressources, avec affectation de priorités aux tâches. Les approches par simulation définissent elles une période de simulation. Si cette période dépasse l'hyperpériode du système au delà de laquelle le comportement du système se répète, l'ordonnancabilité du système est assurée. Les tests de faisabilité permettent de vérifier si il est possible de construire un ordonnancement des tâches sur un système sous certaines conditions. Au delà de ces vérifications, un outil d'exploration d'architecture doit permettre la construction efficace d'un jeu de tâches. Il nécessite aussi de les déployer sur des ressources, et de les ordonnancer sans qu'aucune d'entre elles ne rate son échéance si cela n'est pas admis. La qualité de l'architecture produite se mesure sur différents aspects, comme le nombre de préemptions, les temps de blocage liés aux ressources partagées, la laxité des tâches, les temps de réponse, etc.

Les premiers travaux auxquels nous avons participé, en collaboration avec le CEA List, concernaient le déploiement de fonctions sur des ressources avec un ordonnanceur à priorité fixe et le calcul d'une solution unique. C'est l'objet de la section 2.2. Cette approche peut être utilisée pour calculer un compromis (temps de réponse / mémoire) à l'aide d'une fonction d'agrégation, et explorer l'espace de solutions à partir d'une solution racine variable. Cependant, elle est intrinsèquement mono-objectif, et n'a pas été prévue pour calculer un ensemble de compromis soumis ensuite à un décideur.

Nous avons ensuite travaillé sur des approches nativement multi-objectifs pour produire un tel ensemble de choix pour l'exploration d'architectures temps réel embarquées, en utilisant la notion de solution non dominée et de front de Pareto. Nos travaux ont porté sur deux problématiques : (i) le DSE pour le regroupement de fonctions en tâches sur un système monoprocesseur,

et le DSE pour les systèmes *Time and Space Partitioning* (TSP) avec prise en compte d'exigences non fonctionnelles au niveau de la sécurité et de la sûreté du système. Ces deux contributions sont décrites dans les sections 2.3 et 2.4.

2.2 Exploration architecturale analytique utilisant une méthode MILP en 2 phases

Dans cette première partie, nous décrivons brièvement les travaux de la thèse de Asma Mehaoui, encadrée par J.-P. Babau, en collaboration avec le CEA List. Mon apport sur ce travail concerne la mise en place de méthodes de modélisation MILP et l'utilisation de CPLEX pour la réalisation d'une stratégie de résolution MILP, qui avait été envisagée avant mon implication dans le projet.

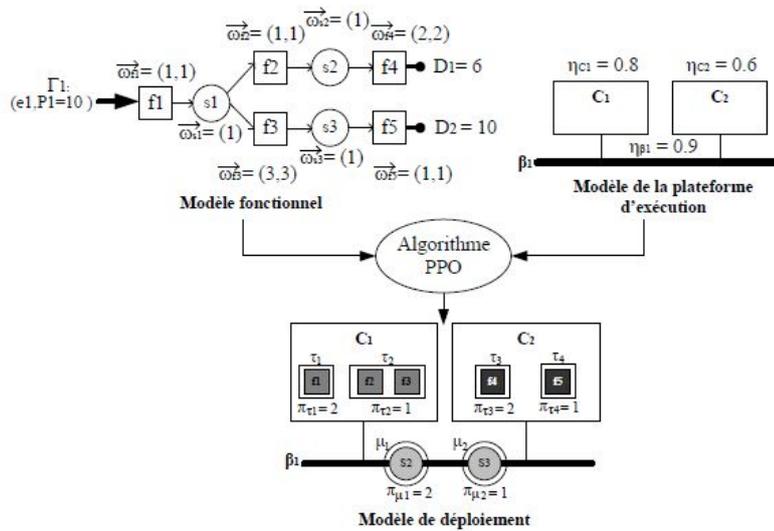


FIGURE 2.3 – Déploiement d'un ensemble de fonctions représenté par un flot de données sur une plate-forme d'exécution pour obtenir un modèle de déploiement

Le problème consiste à déterminer un modèle de déploiement d'un ensemble de fonctions sur une plate-forme d'exécution (voir la figure 2.3 tirée de [110]), en minimisant les temps de réponse, la laxité et/ou l'empreinte mémoire. La laxité correspond à la marge minimale entre la date d'achèvement d'une tâche et son échéance. L'empreinte mémoire, elle, correspond à la place occupée par les tâches en mémoire.

Si 2 objectifs sont considérés, la fonction objectif est une somme pondérée des deux valeurs. Ce déploiement est effectué en 3 étapes, à savoir :

- (i) le placement des fonctions sur les ressources de calcul (situation $f_i \rightarrow C_j$ dans la figure

- 2.2 et celui des communications entre fonctions (dénommées *signaux* par la suite) sur les ressources de communication (bus) d'une plate-forme d'exécution,
- (ii) le partitionnement de ces fonctions et signaux, respectivement en tâches et messages, et
 - (iii) l'ordonnancement des tâches et messages (c'est-à-dire l'affectation de priorités à ces tâches et messages) en tenant compte des dépendances entre fonctions.

Contributions Le problème de déploiement a été modélisé sous la forme d'un MILP. Ce modèle prend en compte les communications entre fonctions, et le pire temps d'exécution associé, ce qui constitue une première pour ces travaux, d'après [4] qui classe 188 papiers sur le problème de déploiement. Les principales difficultés portent sur la modélisation en elle-même, très complexe, avec la définition à expliciter dans un programme linéaire d'ordre partiel de priorité entre tâches. Des techniques de linéarisation des contraintes de placement ont également été introduites. Cette modélisation est une contribution importante du travail. Pour la résolution, une approche en 2 phases permet d'étendre la taille des instances solvables en exploitant le modèle MILP. De plus, la résolution exacte du problème MILP a permis de valider partiellement une approche évolutionnaire applicable à des instances de taille industrielle en comparant les résultats des deux méthodes sur des instances de petite taille.

La taille du problème MILP associé augmente rapidement (par ex : pour 16 fonctions à déployer, 15 signaux, 10 processeurs et un bus, 24000 variables et 91000 contraintes), et les temps d'exécution s'envolent, comme le montre la figure 2.4. Au delà de 60 fonctions des problèmes de gestion mémoire apparaissent à l'exécution du solveur CPLEX.

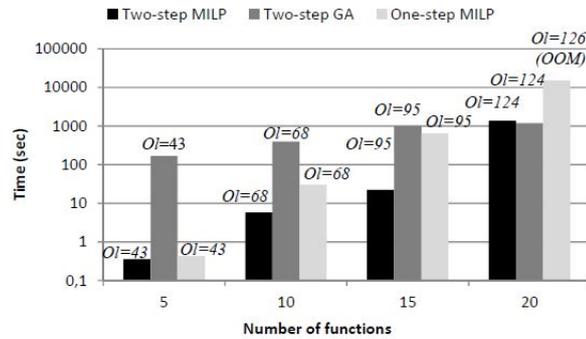


FIGURE 2.4 – Temps d'exécution suivant le nombre de fonctions pour un déploiement sur 2 processeurs et 2 bus. Comparaison entre une approche MILP 2 phases avec un algorithme génétique et une approche MILP standard. *Ol* est la valeur de la fonction objectif [C06]

Pour permettre la résolution d'instance de taille significative, les trois étapes (i-ii-iii) précédemment mentionnées peuvent être réorganisées en 2 phases : (1) le placement des tâches (avec un clustering des fonctions en tâches et un ordre de priorité prédéfini sur les tâches) suivi de (2) le clustering des fonctions affectées à un même noeud en tâches et calcul des priorités associées.

Les nouvelles tâches ainsi constituées peuvent être utilisées pour reboucler sur la phase (1). On itère ainsi jusqu’à convergence du résultat ou dépassement d’un budget de temps de résolution. Ce processus est illustré dans la figure 2.5.

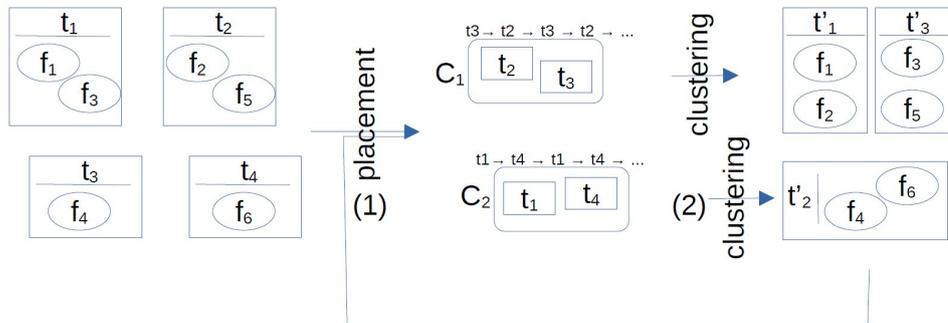


FIGURE 2.5 – Méthode de résolution en 2 phases pour le déploiement

Ces travaux ont été présentés dans [C04, C06].

2.3 Exploration architecturale par simulation avec MOEA

Les travaux précédents étaient pilotés du point de vue métier par le CEA List. Les systèmes temps réel embarqués et l’ordonnancement font aussi l’objet de travaux de recherche au sein de l’équipe MOCS du Lab-STICC à Brest. J’ai participé aux travaux autour de l’outil de simulation d’ordonnancement Cheddar [132] initiés par F. Singhoff. Cet outil permet de calculer les caractéristiques d’un ordonnancement suivant différentes politiques, en spécifiant l’architecture logicielle et matérielle simulée à l’aide du langage AADL¹ [52].

Dans le cadre d’une collaboration avec le laboratoire RedCAD de Sfax, Tunisie, j’ai encadré avec F. Singhoff les travaux de thèse de Rahma Bouaziz à partir de 2014. Nous avons orienté ses travaux vers l’exploration architecturale basée sur l’optimisation multi-objectif, avec comme support l’outil Cheddar.

L’exploration architecturale des systèmes temps réel à partir d’une description fonctionnelle nécessite comme nous l’avons vu dans la section 2.1 de regrouper des fonctions en tâches. Au delà de la latence, d’autres métriques comme la laxité, le nombre de préemptions, les temps de blocage sur les ressources partagées, etc, peuvent être considérées comme fonctions objectifs, et calculées par simulation avec un outil comme Cheddar [132]. Le fait d’utiliser un simulateur, comparativement à des approches analytiques, permet d’étendre l’applicabilité des outils à un spectre plus large de systèmes.

Les outils permettent aussi de réaliser des tests de faisabilité (l’ordonnançabilité) des solutions. Une des difficultés, comme pour dans le cas des mémoires flash (voir section 3.3) réside

1. Architecture Analysis & Design Language

dans le coût d'évaluation des solutions par simulation.

Contributions Le caractère multi-objectif de l'optimisation nous a mené à l'utilisation de la métaheuristique MOEA PAES (voir section 1.2.3, [87]), avec une spécialisation dédiée de l'encodage des solutions et des opérateurs de recherche et une version parallèle maître-esclave. Cette approche nativement multi-objectif contrastait avec les approches existantes pour le placement, qui géraient les objectifs multiples via des sommes pondérées (Section précédente 2.2, mais aussi [128] et [109]). Une seconde contribution était la mise à la disposition du logiciel, avec une architecture ouverte permettant la réutilisabilité de l'approche pour d'autres métriques. Enfin, une étude de la corrélation des objectifs optimisés a été menée pour confirmer ou infirmer les intuitions des concepteurs lors de l'exploration architecturale.

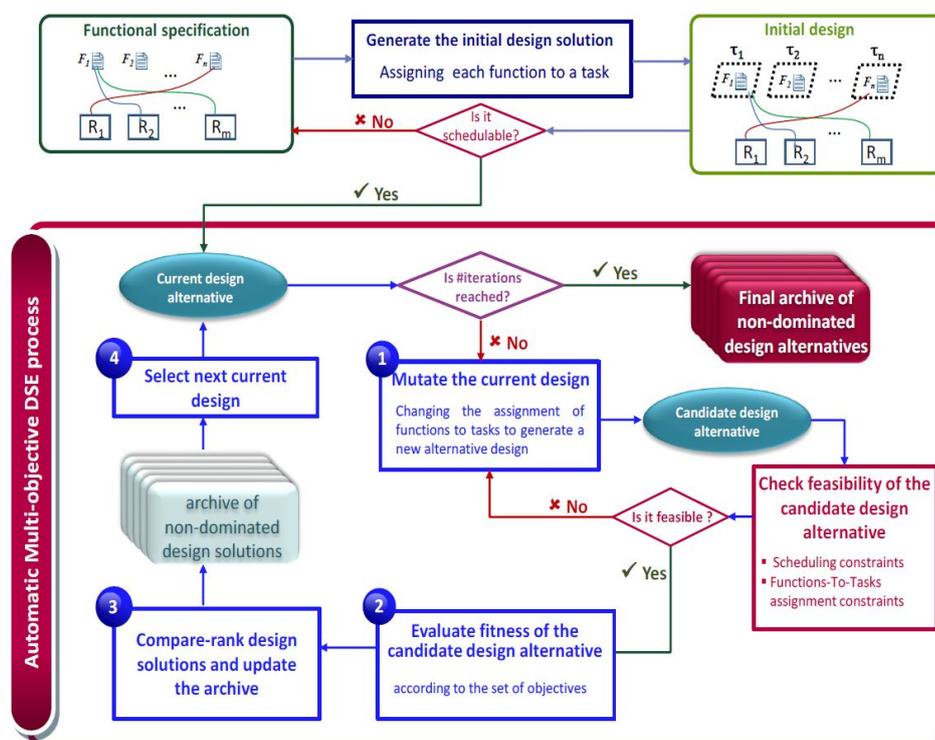


FIGURE 2.6 – Approche MOEA/PAES pour l'exploration d'architecture par simulation [22]

La figure 2.6 montre le cycle d'exploration pour l'affectation d'un ensemble de fonctions à des tâches, qui peuvent être ensuite ordonnancées sur une plate-forme mono ou multiprocesseurs (voir aussi le schéma général de la figure 2.2). Le cycle incorpore des tests de faisabilité sur la solution courante et un ordonnancement dont les caractéristiques sont évaluées par simulation. Les facteurs pris en compte sont la laxité (marge sur le respect de l'échéance d'une tâche), le

nombre de préemptions et le temps de blocage lié aux ressources partagées. Le travail a été effectué sur des cas de test du domaine de l’avionique avec des systèmes respectant le standard Ravenscar [27] (taches périodiques asynchrones, éventuellement dépendantes sur des ressources partagées, avec un ordonnancement préemptif à priorités fixes).

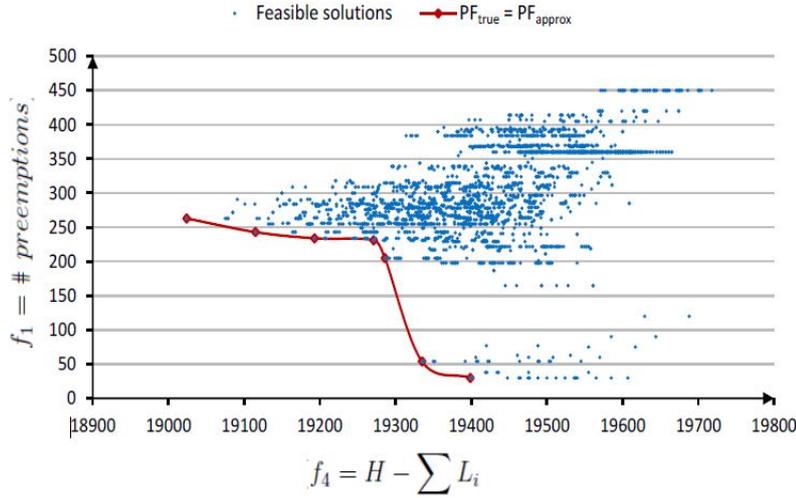


FIGURE 2.7 – Front exact de 7 solutions obtenu par énumération pour un cas de test à 11 fonctions [22]

L’approche multi-objectif a nécessité de mettre au point un opérateur de mutation spécifique s’appliquant sur un chromosome (une solution) représentant une affectation des fonctions aux taches. La représentation est simple (du type $Chrom[i] = j$ si la fonction f_i est calculée par la tache T_j), mais la représentation a été rendue non redondante [22]. L’opérateur de mutation tient compte de la sémantique en restreignant les regroupements aux taches harmoniques entre elles. Il inclut aussi des tests de faisabilité du design résultant.

L’espace de recherche est important (nombre de Bell [127]) et les évaluations sont coûteuses. Par exemple, pour un cas de test de 11 fonctions, un algorithme par énumération implicite nécessite 7h25 pour produire le front de Pareto exact présenté dans la figure 2.7 (cas de minimisation du nombre de préemptions (f_1) et de la laxité (f_4) du design [C11]). 2530 alternatives passaient les tests de faisabilité parmi les 678,570 solutions théoriques (nombre de Bell du nombre possible de clusters). 7 de ces solutions faisables constituaient l’ensemble de Pareto. Ce faible nombre de solutions non dominées peut s’expliquer par les problèmes de faisabilité des solutions.

Une parallélisation asynchrone maître-esclave de l’algorithme a été proposée. Le maître exécute l’algorithme PAES lui-même et gère l’archive. Les esclaves sont chargés de la mutation et de l’évaluation des solutions. Il n’y a pas dans cette version de solution courante, mais des solutions reçues par le maître, de façon asynchrone, après mutation et évaluation par les esclaves.

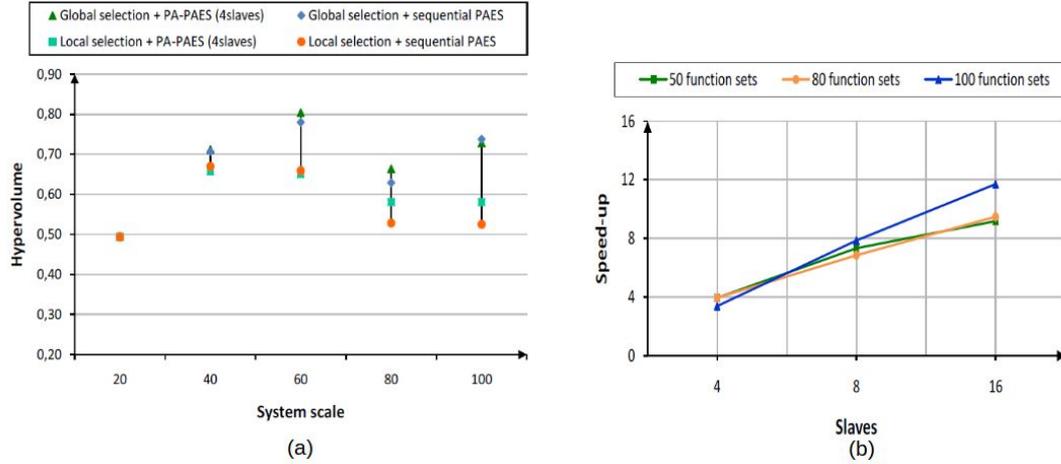


FIGURE 2.8 – (a) Qualité comparée et (b) accélération pour différentes stratégie de parallélisation de PAES pour un cas de test [22]

Ceux-ci reçoivent une nouvelle solution à traiter sélectionnée dans l’archive suivant un critère aléatoirement choisi : (i) solution aléatoire, favorisant l’exploration de l’espace de recherche (ii) une solution dans les 10% meilleures selon une des fonctions objectifs, pour l’exploitation élitiste (iii) une solution dans une zone peu représentée de l’espace objectif, pour la diversité du front produit. Une recherche locale (mutations multiples) a aussi été mise en œuvre. Les différentes stratégies parallèles permettent d’améliorer en terme d’hypervolume les résultats d’un algorithme PAES séquentiel (voir figure 2.8 (a)). Les accélérations obtenues sont présentées sur la figure 2.8(b).

Une étude de la corrélation entre les différents objectifs considérés a aussi été menée durant la thèse de R. Bouaziz. Même si l’expérience des concepteurs leur permet d’anticiper la contradiction entre certains objectifs, ce travail a permis de corroborer et quantifier ces intuitions pour un jeu de test considérant le nombre de préemptions, la laxité et les temps de blocage des tâches liés aux ressources partagées (voir fig. 2.9).

La forte corrélation négative entre nombre de préemptions et laxité était attendue, celle entre laxité et temps de blocage moins prévisible. Ce travail a, entre autres, montré l’intérêt de l’automatisation du processus de DSE considérant des objectifs multiples pour présenter un ensemble de solutions au concepteur.

Ces travaux ont été publiés dans [C11, C12, R05]. Le choix d’une approche MOEA pour l’exploration d’architectures a été motivé par le fait que ces métaheuristiques produisent un ensemble de solutions. Cet ensemble de compromis est requis par les concepteurs. Le choix de PAES a été à la fois guidé par le fait que seul un opérateur de mutation devait être défini mais aussi par la simplicité de mise en œuvre de l’algorithme. En effet celui-ci devait être réimplanté dans le logiciel Cheddar, sans nécessiter un interfaçage vers un outil extérieur, car une version

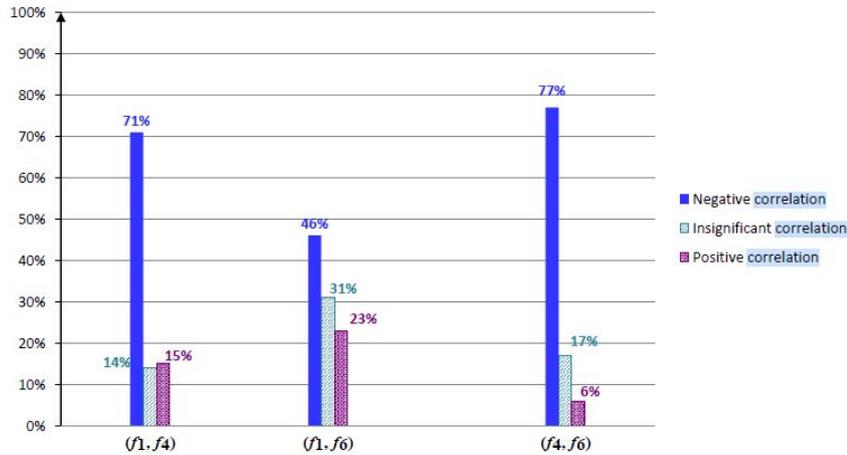


FIGURE 2.9 – Corrélation positive, négative ou insignifiante entre préemptions (f_1), laxité (f_4) et temps de blocage (f_6) pour un jeu de test de 100 cas [22] avec un seuil de confiance de 0.05 [146]

de Cheddar est commercialisée. Pour ces mêmes raisons, les composants disponibles développés pour le calcul d’un front de Pareto par énumération et l’exploration avec la métaheuristique PAES ont été réemployés dans les travaux présentés dans la section suivante.

2.4 Exploration d’architectures temps réel hiérarchiques sécurisées par MOEA

Dans le prolongement du travail de Rahma Bouaziz, la thèse de Ill-Ham Atchadam s’intéresse elle aux systèmes temps réel hiérarchiques, notamment de type TSP (*Time and Space Partitionning*, [3]). Le travail précédent consistait à regrouper des fonctions en tâches et à les ordonnancer sur une plate-forme monocore. Dans les travaux de cette section, ce sont les tâches qui sont regroupées en partitions, isolées en termes de mémoire (space) et de temps (time), et ordonnancées sur un ou plusieurs processeurs. L’approche TSP vise à faire coopérer des logiciels de provenances diverses et de complexité croissante sur une plate-forme commune. Le but est d’économiser de l’espace et de l’énergie sur les systèmes embarqués avioniques.

Le processus d’ordonnancement est celui présenté dans la figure 2.2 : (i) les tâches de chaque partition sont ordonnancées en interne comme dans le cas d’un système non hiérarchique (étape (c) de Fig. 2.2). Les tâches sont exécutées suivant l’ordonnancement prévu quand la partition est activée. (ii) Les partitions elles-mêmes sont ordonnancées statiquement, avec un temps d’exécution d’une durée prédéfinie (étape (d) Fig. 2.2). Elles s’exécutent cycliquement sur les ressources de calcul, en répartissant éventuellement les tâches qu’elles exécutent sur plusieurs processeurs.

Le partitionnement induit 2 types de communications entre les tâches. (i) Les communications sont internes à la partition si les deux tâches communicantes sont situées dans la même partition ; elles sont inter-partitions dans le cas contraire. Les mécanismes mis en œuvre dans les deux cas sont différents, induisant notamment un surcoût en terme de latence dans le second cas [7]. Une partie de la sécurisation du système repose sur l'étanchéité entre les partitions, mais cela n'est pas suffisant pour assurer la sécurité des communications entre les tâches. Celles-ci peuvent être espionnées et/ou altérées en interne à la partition, ou lors des communications inter-partitions. Pour pallier aux problèmes de confidentialité et d'intégrité, il est possible de crypter ces communications. Cependant cela entraîne un surcoût de calcul, soit dans des fonctions internes aux tâches, en augmentant le temps d'exécution de celles-ci, soit en ajoutant des tâches spécifiques de cryptage et de décryptage, qui doivent elles-mêmes être ordonnancées.

Pour limiter l'impact de la sécurisation, on peut choisir de ne protéger que certaines de ces communications, qualifiées alors de critiques. Pour les autres communications, la protection est optionnelle. Les premières induisent des tests de faisabilité sur le système sécurisé, les secondes peuvent correspondre à une métrique de sécurisation (combien parmi les communications dont la sécurité est optionnelle sont sécurisées dans une solution donnée?).

La définition des communications critiques peut se faire en utilisant des règles de bonne pratique comme Bell-Lapadula (BLP) [13] pour la confidentialité ou Biba [19] pour l'intégrité. Par exemple, le principe de BLP est *no read up, no write down* : à chaque tâche est associée un degré de confidentialité (défini par le concepteur), et une tâche ne peut communiquer sans sécurisation avec des tâches d'un degré inférieur.

Contributions Les travaux ont porté sur l'exploration d'architectures TSP sécurisées, en considérant à la fois la qualité de l'ordonnancement (nombre d'échéances ratées des tâches non critiques, i.e. autorisées à rater leur échéance) et les métriques de sécurité (nombre de violations des règles de sécurisation pour les communications non critiques). Une première contribution a donc été la définition de métriques de sécurité liées à un modèle de sécurisation. L'impact en terme d'ajout de composants logiciels au système a aussi été pris en compte. Une approche MOEA permettant l'exploration architecturale des compromis entre ordonnancabilité et sécurité des systèmes TSP a été proposée et étendue aux plate-formes multi-cœurs. Des travaux existants ciblent l'optimisation de la sécurité [94] [148] tandis que d'autres se concentrent sur l'optimisation de l'ordonnancement [149]. Dans [133] un compromis ordonnancement/sécurité est calculé par un protocole adhoc. Enfin, comme dans la section précédente ou [74] des outils d'exploration architecturale existent mais pas pour les systèmes hiérarchiques TSP. Pour finir, l'implantation a été rendue disponible au travers de l'outil Cheddar [132].

Une approche par optimisation multi-objectif basée sur PAES a là encore été utilisée. La figure 2.10 représente l'encodage d'une solution sous la forme d'un chromosome, avec trois par-

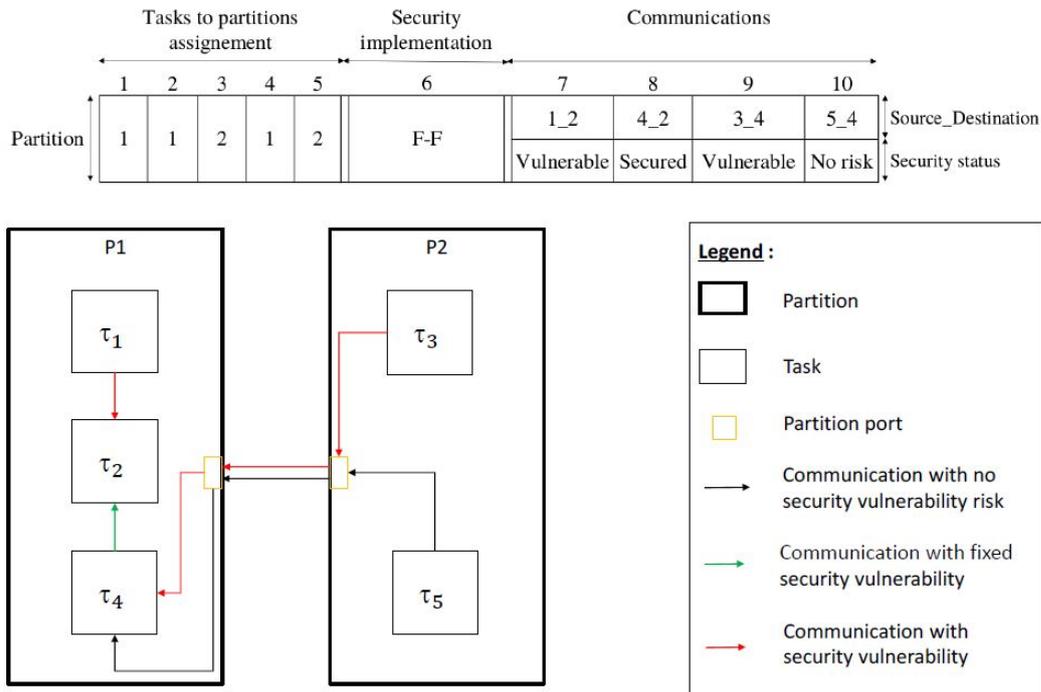


FIGURE 2.10 – Encodage d'une solution pour le partitionnement de tâches et la sécurisation des communications [7]

ties :

- l'affectation des tâches aux partitions
- le choix du mode de sécurisation des communications (7 modes possibles : avec ou sans tâche spécifique, avec une tâche unique par partition, etc)
- la sécurisation choisie pour chaque communication
- dans le cas multi-cœurs : l'affectation des tâches sur les cœurs

Comme on peut le voir, l'espace de recherche est bien plus grand que dans le cas de regroupements des fonctions en tâches (cela correspond à la première partie de l'encodage). Il a donc fallu définir une stratégie spécifique pour une recherche efficace de solutions.

L'exploration de l'espace de recherche correspond, avec PAES, à la définition d'opérateurs de mutation. Une mutation opère en fait sur les différentes portions du chromosome : changement du mode de sécurisation, changement du statut sécurisé ou non sécurisé pour chaque communication, partition choisie pour chaque tâche. Pour ce dernier aspect, plusieurs stratégies ont été proposées :

- prendre en compte les applications. Chaque tâche fait partie d'une application : par exemple un logiciel de vol est constitué d'un ensemble de tâches qui coopèrent entre elles et ont le plus souvent le même degré de confidentialité. L'algorithme de mutation des

taches peut opérer uniquement au niveau application, en conservant toutes les taches d'une application dans une même partition (algorithme *appgrain*). Cette approche réduit fortement la taille de l'espace de solutions, permettant potentiellement d'explorer celui-ci efficacement.

- ne pas prendre en compte les applications en autorisant la dispersion des taches les constituant entre partitions (algorithme *taskgrain*). L'espace de solutions est alors plus riche, mais potentiellement plus difficile à explorer correctement.
- effectuer durant le déroulement de l'algorithme global un passage de la stratégie *appgrain* à la stratégie *taskgrain*. La première phase permet d'obtenir les "bonnes solutions" dans l'espace de recherche réduit du niveau applications avant de raffiner les solutions obtenues en relâchant la contrainte de localisation des applications sur une partition unique. Cette dernière stratégie est appelée *mixgrain*

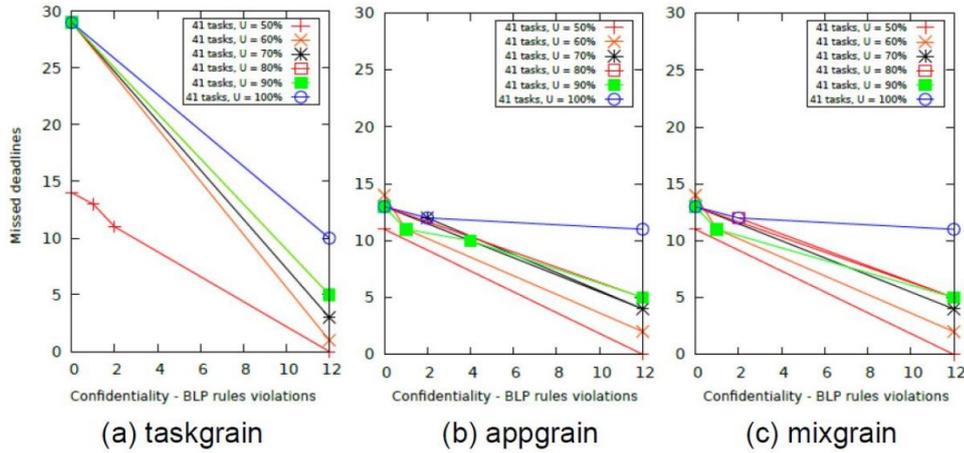


FIGURE 2.11 – Compromis échéances ratées - communications non sécurisées pour les 3 stratégies de mutation [W07], en fonction du taux d'occupation du processeur par les taches

Les résultats de l'approche mixte sont pour le cas de test présenté similaires à ceux de l'approche *taskgrain*.

Ces travaux ont été publiés dans [W06, C19, C18, W07].

Les derniers travaux que nous avons menés concernaient le déploiement multi-cœurs. Ils permettaient aussi d'aborder le problème de la sûreté des architectures, en effectuant de la redondance active de taches.

La figure 2.12 montre un ensemble de solutions obtenues en optimisant 3 métriques (échéances ratées, violations de confidentialité et nombre de cœurs) pour le DSE d'un système de 2 applications, 17 taches tripliquées, 2 partitions et 57 cœurs au maximum (1 cœur par tache), en 12 heures de calcul (la taille de l'espace de solutions est approximativement de 10^{146}).

Le système a trouvé que 11 cœurs suffisaient pour obtenir un système complètement ordon-

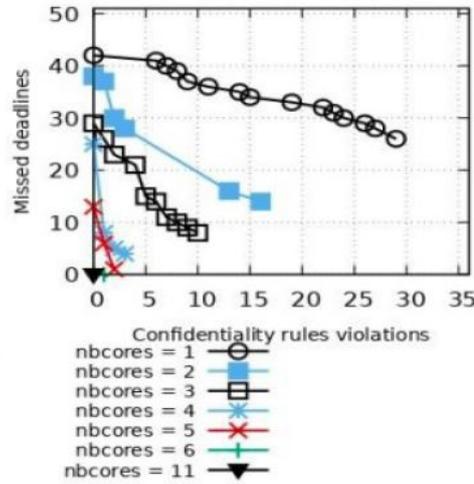


FIGURE 2.12 – Compromis trouvés pour un DSE sur 17x3 tâches, 2 partitions et 57 cœurs maximum [W07]

nancé et sécurisé. D'autres compromis ont également été définis, mais le temps de calcul reste important.

Pour pallier à ces temps de calculs importants et pouvoir éventuellement étendre le domaine architectural (par exemple en considérant des cœurs hétérogènes [12]), une exploration renforcée par une recherche locale, une parallélisation des évaluations comme dans les travaux de R. Bouaziz (cf section 2.3), un travail sur le temps consacré aux différentes phases sont des pistes à explorer.

2.5 Conclusion

Comme leur nom l'indique, le temps est une caractéristique essentielle des systèmes temps réel. Il est donc lié aux métriques à prendre en compte lors de leur optimisation. En terme de résolution, l'ordonnancement temps réel de tâches est un problème compliqué. L'espace de recherche est complexe, et trouver un ordonnancement valide est parfois ardu. Enfin, des objectifs d'optimisation comme la laxité, le nombre d'échéances ratées, les temps de blocage, le degré de sécurisation du système peuvent être pris en compte lors de l'exploration d'architectures.

Nous avons essentiellement travaillé dans le cadre de l'utilisation de méthodes MOEA, avec des évaluations d'ordonnancement par simulation. Cette approche est flexible et permet de réduire les temps de calcul, mais pose au moins deux problèmes : (i) la convergence des solutions vers un front de Pareto difficilement calculable, à part pour les instances de petite taille, par énumération ; (ii) l'évaluation des solutions par simulation peut être très coûteuse. Pour ce dernier point, le parallélisme peut aider, mais d'autres approches sont à explorer.

Ces travaux ont eu lieu dans le cadre de 3 thèses aux Lab-STICC. J'ai participé aux travaux de la première, dirigée par J.P. Babau, sans la co-encadrer. J'ai ensuite co-encadré les deux suivantes avec Frank Singhoff. Nous avons défini des modèles de l'espace de solutions dans ces deux cas. Un point remarquable de la thèse de Rahma est le travail sur la corrélation ou non des métriques utilisées qui permet de guider la conception du système. Un second point est l'utilisation d'une approche maître-esclave pour la résolution en parallèle du problème. Pour la thèse de Ill-Ham, des métriques originales pour mesurer la sécurisation de systèmes temps réel hiérarchiques ont été proposées. Une méthode de résolution en 2 phases permet de gérer la complexité de l'espace de recherche pour une exploration efficace. Pour ces deux thèses, les approches développées ont été intégrées dans le cadre du projet Cheddar au logiciel du même nom [132], distribué par la société Elidiss Technologies et intégré dans de nombreuses plateformes logicielles (produit AADL Inspector, logiciel TASTE de l'ESA, OSATE de Carnegie Mellon University, ...). La thèse de Rahma a eu lieu en collaboration avec l'université de Sfax, Tunisie. Toutes ces thèses ont été valorisées au travers de la publication avec les doctorants d'une revue, 5 conférences et 2 workshops.

SYSTÈMES DE STOCKAGE

3.1 Introduction

Dans cette partie, nous exposons les problématiques liées à l'accès et au stockage des données que nous avons rencontrées. L'accès dépend à la fois des caractéristiques du médium de communication et de celles du périphérique de stockage. Dans les deux cas, une politique d'utilisation doit être choisie. De plus, la localisation des données (production, stockage, consommation) est déterminante. La plupart du temps, une stratégie d'accès peut être optimisée pour réduire les temps de lecture/écriture des données tout en considérant d'autres facteurs.

Nous verrons dans la première section comment trouver des compromis pour la politique de gestion du réseau dans le cadre de la diffusion de films. Puis, c'est la stratégie de mise en cache des données lors de la lecture et l'écriture sur une mémoire Flash (avec des critères d'usure du support et d'encombrement des données de gestion de la mémoire). Nous nous intéresserons également à la localisation des données au plus près de leur production et de leur utilisation dans le cadre de l'internet des objets (IoT). Enfin, en s'éloignant du cadre embarqué, nous verrons comment optimiser le placement des données dans le Cloud, en profitant des mécanismes de fédération.

3.2 Réserve de bande passante dans un réseau clos

Certains problèmes se modélisent en utilisant un graphe, explicite ou sous-jacent. Dans le second cas, l'application des algorithmes issus de la théorie des graphes permet parfois de garantir l'optimalité de la solution. Ce principe a été mis en œuvre dans le domaine de la réserve de bande passante pour la diffusion de films dans un environnement réseau maîtrisé, comme par exemple le réseau interne d'un avion.

3.2.1 Problématique

Garantir la qualité de service (QoS) lors de la diffusion d'une vidéo à partir d'un serveur nécessite de réserver suffisamment de bande passante sur les équipements réseau acheminant le flux vidéo jusqu'au terminal client. Les besoins en bande passante sont connus si la vidéo ne

correspond pas à un flux en direct mais par exemple à un film. Dans ce cas, le débit nécessaire à chaque moment pour la diffusion est connu et stocké dans un *profil* du film, ou calculable avant le démarrage de la diffusion, en analysant le fichier qui stocke ce film sur le serveur. Si le débit nécessaire est constant (encodage du film en CBR, *constant bitrate*), la réservation de bande passante à effectuer est fixe et une seule configuration des équipements réseau est nécessaire. Par contre, si les besoins réseau varient car l'encodage est de type Variable Bit Rate (VBR), il faut alors être capable de faire varier le niveau de réservation des différents équipements acheminant le flux tout en respectant des contraintes opérationnelles de ces équipements embarqués. Il peut par exemple exister un délai entre une demande de réservation de bande passante et sa mise à disposition effective qui restreignent en pratique la fréquence et/ou le nombre de reconfigurations du niveau de débit possibles durant la diffusion [C09]. Il faut donc réserver suffisamment de bande passante, au plus près du besoin. Cela évite de pénaliser d'autres flux de données sur le réseau.

Approche Notre postulat est que si la bande passante réservée à tout moment de la diffusion est supérieure ou égale au besoin en débit du film à chaque instant, la QoS est garantie. Les réservations effectuées constituent alors ce que nous appelons une *enveloppe de réservation* du profil du film, i.e. des besoins en bande passante à chaque instant. La figure 3.1(b) est un exemple. Notre objectif est de trouver la meilleure enveloppe de réservation, qui garantit cette QoS maximale tout en minimisant le surcoût en bande passante réservée. Pour trouver une enveloppe optimale, notre approche s'appuie sur des algorithmes de plus court chemin. L'optimalité concerne l'utilisation des ressources réseau en tenant compte des contraintes d'implantation de ces demandes de réservation. Un serveur central de diffusion génère le flux vidéo (streaming) en effectuant des demandes d'allocation de bande passante jusqu'au destinataire (le passager dans un avion).

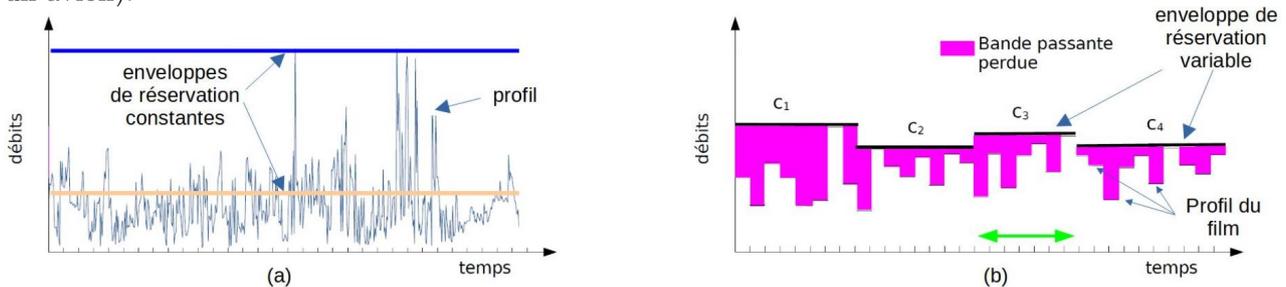


FIGURE 3.1 – Le profil (débits dans le temps) d'un film, avec (a) les politiques de réservation constante maximale et moyenne et (b) une politique de réservation variable constituée de 4 configurations. Son coût associé en bande passante perdue correspond à la surface entre le profil et l'enveloppe. Une contrainte de durée minimale d'une configuration est illustré par la flèche verte.

Comme l'illustre la fig. 3.1 (a), deux politiques triviales existent pour effectuer l'allocation. La première est basée sur une réservation constante égale à la valeur maximale de la bande

passante nécessaire [12,14,17]. Mais parce que le VBR inclut de nombreux pics de débit, significativement plus élevés que le débit moyen, cette solution conduit à une sur-réserve de la ressource. Pour éliminer toute sur-réserve il faudrait réserver exactement ce qui est requis suivant l'évolution du flux. Or il y a une limitation pour une telle solution, car la réserve ne peut pas être ajustée aussi fréquemment que les besoins en bande passante vidéo varient pendant le streaming vidéo (par exemple, pour une carte Beagleboard, une reconfiguration de la réserve nécessite 0,14 ms). Entre ces extrêmes, notre objectif est de trouver la meilleure séquence réalisable de réserves de bande passante (l'enveloppe de réserve fig 3.1 (b)) qui garantit la QoS maximale du réseau en réservant sur chaque configuration le pic de bande passante de la période associée. Une séquence est réalisable si elle respecte les contraintes techniques liées à la période minimale autorisée entre 2 (re)configurations du réseau. La qualité d'une enveloppe se mesure à la sur-réserve de bande passante qu'elle engendre (zone rose sur la fig. 3.1(b)).

Contributions La méthode de réserve de bande passante est originale, avec une modélisation à l'aide de graphes et une résolution exacte d'un problème de plus court chemin. Les politiques de réserve les plus simples (par ex. [97]) se basent sur les pics de demande de ressources pour allouer un niveau constant et donc élevé de ressources. Comme relevé dans [107], certaines approches utilisent des méthodes statistiques en analysant la structure de l'encodage VBR pour prédire les besoins en bande passante. D'autres implantent des méthodes d'adaptation en temps réel [82]. Notre approche par discrétisation en slots de temps nous permet une analyse hors ligne et un résultat théorique exact, car faisant l'hypothèse restrictive d'un réseau fermé.

3.2.2 Modélisation et résolution

Nous avons modélisé le problème à l'aide de la théorie des graphes.

Les besoins en bande passante sont modélisés par une série de valeurs $b = \{b_1, b_2, \dots, b_n\}$, b_i étant la bande requise à l'instant i ($1 \leq i \leq n$, avec n la taille de la séquence), exprimée par. ex. en kbits/s.

Soit r_i le niveau de réserve au temps i et $R = \{R^1, R^2, \dots, R^m\}$ l'enveloppe de réserve, définie comme suit. Chaque configuration R^k correspond à 2 valeurs : (i) une date de démarrage $t(R^k) = i$, et (ii) la valeur la réserve R^k elle-même, demandée au début de la période de temps i . Cette valeur est constante jusqu'à $t(R^{k+1})$. Donc, pour chaque temps i t.q $t(R^k) \leq i < t(R^{k+1})$, $r_i = R^k$. Une configuration fictive $R^{m+1} = 0$ avec $t(R^{m+1}) = n + 1$ est définie pour la fin de la diffusion.

Nous considérons que le si niveau de réserve R^k vaut au moins le débit maximal sur la période, cela permet d'obtenir la QoS maximale en minimisant les ressources de bufférisation nécessaires (Eq. (3.1)).

$$R^k \geq \max_{i \in [t(R^k)..t(R^{k+1})]} b_i \quad (3.1)$$

Le coût de l'enveloppe de réservation R est le surcoût induit par R pour le film. La meilleure enveloppe R^* est celle qui minimise le coût $C(R)$ défini par l'Eq. 3.2. Les coûts partiels sont définis par $C(R^k) = \sum_{i \in [t(R^k)..t(R^{k+1})]} (r_i - b_i)$.

$$C(R) = \sum_1^m C(R^k) = \sum_1^n (r_i - b_i) \quad (3.2)$$

Ces coûts sont illustrés sur la fig. 3.1 (b) (les configurations correspondent aux R^i)

Les contraintes techniques impliquent des critères de faisabilité, notés M et P :

- M est le nombre maximal de (re)configurations autorisées. $m \leq M$
- P est l'intervalle de temps minimal entre 2 (re)configurations des ressources réseau. $\forall k \in [1..m], t(R^{k+1}) - t(R^k) \geq P \geq 1$,

Le calcul de la séquence optimale est effectué pour une série b de valeurs (profil), et des contraintes sur le nombre maximal M de configurations et/ou la durée minimale P avant une reconfiguration. Le résultat est une enveloppe R^* d'un (sur)coût $C(R)$ minimum (Eq. 3.2).

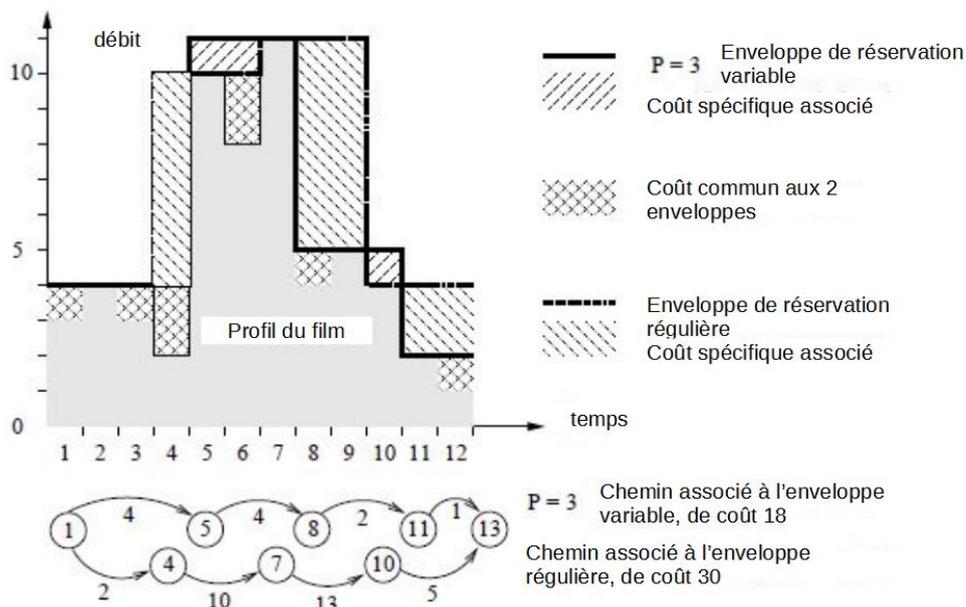


FIGURE 3.2 – 2 enveloppes de réservation possibles et leur équivalent en terme de chemin dans le graphe. La première ($P = 3$) correspond à une période de reconfiguration minimale de 3, la seconde à une reconfiguration régulière toutes les 3 unités de temps. Chaque enveloppe correspond à un chemin dans le graphe de modélisation du problème [R04]

L'ensemble des enveloppes faisables correspond aux chemins possibles dans un graphe comme l'illustre la fig. 3.2. Chaque temps i correspond à un sommet du graphe, et un arc $i \rightarrow j$ dénote une configuration débutant à i et remplacée par la suivante au temps j . L'arc est pondéré par le surcoût engendré par la réserve pour la période $[i, j]$. Tous les arcs $i \rightarrow j$ avec $j \geq i + P$ sont inclus dans le graphe (celui-ci peut être simplifié, cf [C05]).

Un plus court chemin du noeud 1 au noeud $m + 1$ correspond alors à une enveloppe optimale. En autorisant uniquement les chemins d'au plus M arcs, on respecte le nombre maximal de configurations autorisées.

L'algorithme de Bellman sur graphe acyclique orienté est utilisable [14, 36] en $\mathcal{O}(n^2)$ si seul P est considéré, et les M premières étapes de l'algorithme de Ford-Bellman doivent être exécutées dans le cas contraire, en $\mathcal{O}(M \cdot n^2)$.

3.2.3 Résultats

Nous avons déployé diverses politiques en faisant varier la valeur de M et/ou P au sein d'une architecture simulée avec l'outil NS2 [78], en ajustant les paramètres de réserve de bande passante à l'émission des données du flux vers le client (taille du *hierarchical token bucket* ou *HTB* [49]). Ce HTB permet de lisser les émissions de données à un débit qui lui est donné en paramètres, en stockant éventuellement celles-ci temporairement (dans un sceau, ou *bucket*).

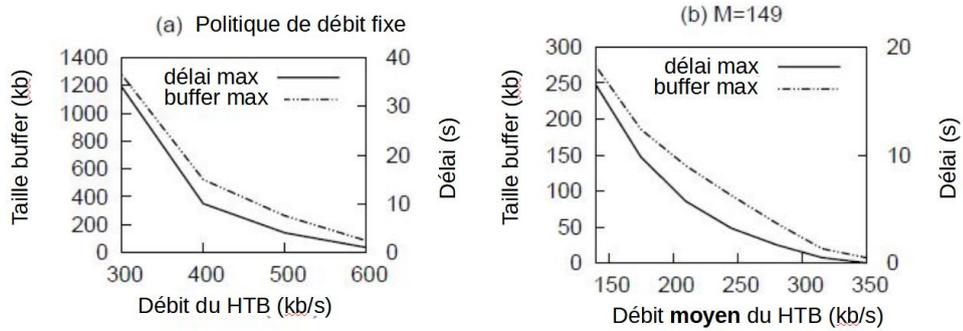


FIGURE 3.3 – Coût en termes de bufférisation et de délai suivant la bande passante réservée pour (a) une réserve de taille constante (débit constant du Hierarchical Token Bucket (HTB)) et (b) une politique optimale avec une contrainte sur le nombre de réservations possibles $M = 149$ [R04]

La figure 3.3 montre la différence de comportement (a) avec une allocation de bande passante de taille fixe (taille du débit du HTB fixe) et (b) avec une politique calculée pour $M = 149$. Suivant le débit réservé sur le lien de communication, l'écoulement des paquets nécessite plus ou moins de bufférisation, et entraîne une latence plus ou moins grande en réception. On peut voir par exemple qu'un niveau de réserve fixe de 600 kbits/s environ est nécessaire pour obtenir

un délai de transmission nul alors que la moyenne des réservations demandées est de 350 kbits/s avec la politique de réservation variable avec un nombre de configurations maximal $M = 149$.

Ces travaux se sont poursuivis en prenant en compte la gestion simultanée de plusieurs diffusions, en jouant sur la date de démarrage des films pour optimiser les pics d'utilisation de bande passante. Une technique par énumération implicite a été utilisée pour cela, en définissant un intervalle maximal de démarrage Δ , qui va correspondre pour chacune de ses valeurs possibles à une branche de l'arbre de recherche pour chaque film à visionner simultanément. *Fusionner* deux films consiste à calculer une enveloppe globale de leurs 2 profils. Deux techniques ont été proposées : la recherche peut s'appliquer (a) sur les enveloppes calculées pour chaque film, ou (b) directement sur les profils des films, ce qui démultiplie les possibilités. L'ajout peut se faire film par film (méthode heuristique) ou en considérant tous les films à visionner simultanément. La table 3.1 montre les résultats pour ces approches pour un cas de test comportant 4 films et un intervalle de démarrage de $\Delta = 5$ (secondes). Sont donnés la valeur pic requise et le surcoût en bande passante allouée par rapport à la bande passante nécessaire pour diffuser le film.

Contraintes sur enveloppe	Tech. de fusion	Exact		Heuristique	
		Pic	Surcoût	Pic	Surcoût
P5	(a)	8339	16.2%	10741	16.5%
	(b)	8010	16.1%	8141	16.8%
P10	(a)	10741	28.7%	10741	28.7%
	(b)	8010	28.6%	8141	28.9%
M1200	(a)	8077	3.6%	8094	3.5%
	(b)	8010	3.4%	8141	3.6%
M600	(a)	8077	9.5%	8293	9.6%
	(b)	8010	9.0%	8141	9.4%

TABLE 3.1 – Coûts et valeur pic finaux pour différences approches concernant la diffusion simultanée de 4 flux multimédia, avec un décalage possible de $\Delta = 5$ [R04]

Comme attendu, la fusion de profils (technique (b)) permet d'obtenir des surcoûts moindres en réservation par rapport à la fusion des enveloppes (a). La méthode heuristique (fusion 1 par 1 des profils ou enveloppes) engendre des surcoûts inférieurs à 0,5% pour nos cas de test par rapport à l'approche exacte (fusion simultanée).

Ces travaux ont été publiés dans [C09, C05, R04]. Les hypothèses sur la nature fermée du réseau et la connaissance parfaite des flux à transmettre restreignent l'applicabilité de l'approche à des cas spécifiques comme les réseaux de divertissement multimédia dans un avion. Cependant, ces hypothèses restrictives ont permis une modélisation du problème à l'aide de graphes et l'exploitation des algorithmes associés. Ils fournissent des solutions exactes de manière efficace, tant en temps d'exécution (algorithmes polynomiaux), que en ce qui concerne leur mise en œuvre.

De plus, la problématique de la réservation de ressources liées à des besoins variables dans le temps avec cette technique de calcul d’enveloppes soumises à des contraintes est générale. Si les contraintes associées à la définition des enveloppes peuvent être traduites dans le modèle basé sur les graphes, alors l’approche par calcul de plus court chemin est applicable. Dans le domaine de la diffusion vidéo, d’autres ressources (CPU, buffers) sont concernées.

3.3 Politique de stockage de mémoire Flash avec MOEA

Le mécanisme d’évaluation par simulation décrit dans la section 2.3 pour l’ordonnancement de systèmes temps réel faisait appel au simulateur Cheddar. La métaheuristique choisie, PAES, impliquait une seule évaluation par génération de l’algorithme MOEA. Pour définir les paramètres d’une politique de stockage de mémoire Flash, plusieurs métriques doivent être considérées et des simulations également chronophages sont nécessaires avec le simulateur MaCach. Une même approche MOO basée sur PAES peut donc être utilisée dans ce cas, comme présenté dans cette section.

3.3.1 Problématique

La politique de gestion de cache d’une mémoire Flash NAND (implantée dans un *Flash Translation Layer*) sert à atténuer les effets sur les performances des contraintes physiques de ce type de mémoire : (i) écriture page par page, (ii) effacement bloc par bloc, (iii) effacement avant écriture, (iv) nombre limité d’effacements (usure des blocs) [24]. Les caractéristiques à optimiser pour la politique d’adressage d’un pilote de mémoire flash sont, entre autres : la latence en lecture/écriture ; la taille de la table d’adressage (pour trouver les pages dans les blocs), l’usure du composant (nombre de cycles d’écritures) Elles peuvent être évaluées en simulant les E/S à partir de traces d’exécution d’un système de stockage pour un temps plus ou moins long.

Contributions L’outil développé, MaCach [R02] est spécifique. Le problème posé d’optimisation de sa configuration est original. Les paramètres d’entrée sont la taille de bloc, le seuil de bascule du mode d’écriture de page à bloc, etc. Dans ces travaux, l’espace de recherche est de taille réduite (3^8), défini par 8 paramètres avec 3 valeurs possibles pour chacun d’entre eux. L’évaluation d’une configuration étant très coûteuse (plusieurs minutes parfois) nous avons proposé d’utiliser la même approche MOEA (PAES, Pareto Archive Evolution Strategy [87]) que pour les travaux de la section 2.3 avec une population réduite à un individu et produisant une seule descendance par itération. Une parallélisation à gros grain de PAES avec un modèle maître esclave a également été proposée.

3.3.2 Résultats

L'approche multi-objectif a d'abord été validée en simulant pour un cas de test (une trace) de petite taille l'ensemble des configurations possibles ($3^8 = 6561$ solutions). On en a déduit le front de Pareto sur les fonctions objectifs temps de réponse et nombres d'effacement. Ce front exact a ensuite été comparé à celui obtenu par le MOEA après 500 itérations (voir fig 3.4). Comme le montre ce cas de test, PAES fournit un ensemble de solutions qui avoisine le front de Pareto, mais imparfaitement, avec 9 points optimaux découverts au bout de 500 itérations sur les 19 du front de Pareto.

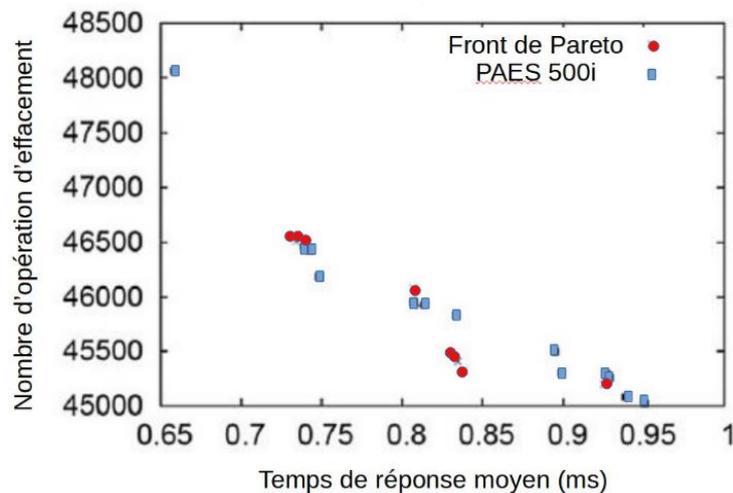


FIGURE 3.4 – Front exact et front obtenu après 500 itérations de PAES pour une trace synthétique [R02]

Le calcul d'un front sur fait sur une trace donnée. Afin de définir un ensemble de configurations intéressantes pour un ensemble de traces, la méthodologie suivante a été proposée :

- (1) calculer un front dit *de référence* pour chaque trace (avec par exemple 500 itérations du MOEA)
- (2) grâce au (1) calculer un intervalle des valeurs possibles pour les objectifs d'optimisation associés pour chaque front de référence.
- (3) resimuler pour chaque trace l'ensemble des configurations correspondant à l'ensemble des fronts de référence
- (4) Garder les solutions qui permettent de rester pour les traces dans des valeurs de fonctions objectifs acceptables. Une telle solution *couvre* la trace. L'acceptabilité est définie par un seuil de tolérance : par ex : 10% signifie que la solution doit fournir pour la trace une valeur pour chaque objectif dans les 10% des meilleures valeurs produites par l'ensemble des solutions pour cette trace

Pour un jeu de traces public [16], les résultats allaient de 63% des traces couvertes au sens

de l'étape (4) ci-dessus (par une seule solution) si le seuil d'acceptabilité était de 10% à 78% des traces couvertes (par 8 solutions) si le seuil d'acceptabilité était de 40 %.

Ces travaux ont été publiés dans [R02]. Étant donnée la complexité du système d'allocation de mémoire Flash, une approche analytique avec une modélisation MILP par exemple n'était pas envisageable ici. Le côté *boite noire* du simulateur se conjugue bien avec l'utilisation d'une métaheuristique. De plus, les partenaires métier souhaitaient expérimenter une approche multi-objectifs. L'utilisation d'une méthode par aggrégation aurait également été possible. L'approche basée sur PAES permettait de s'affranchir des problématiques de pondération/normalisation des objectifs engendrées par les approches par aggrégation.

L'utilisation d'algorithmes multi-objectifs dans ce cadre permet d'illustrer deux difficultés possibles dans l'utilisation des MOEA :

- la nécessité de réaliser de nombreuses évaluations de fonctions objectifs, coûteuses en temps d'exécution. Ici, le simulateur utilisé nécessite parfois plusieurs minutes pour un cas de test. Cela dépend à la fois de la solution évaluée et de la taille de la trace d'exécution utilisée pour la simulation. Comme dans le cadre de l'optimisation d'un ordonnancement temps-réel (Section 2.3) notre stratégie a consisté à utiliser un algorithme MOEA, PAES, qui ne stocke que des solutions non dominées. Il n'évalue pas à chaque étape une population de solutions complète mais une seule solution courante.
- le fait que l'évaluation par simulation se base sur des données d'entrée (une trace d'exécution pour notre problème) qui n'est pas forcément celle qui sera utilisée dans le cadre réel. Cela est lié à la robustesse des solutions proposées vis à vis de ces données de simulation. Nous avons également été confrontés au problème dans le cadre de l'optimisation du routage de voiliers avec les incertitudes sur les données d'entrée (les prévisions météorologiques, voir Section 4.3).

Nous sommes dans le cadre des projets RESSACH et DISPEED (Section 5.2) à nouveau confrontés aux problématiques de coût d'évaluation des fonctions objectifs et de robustesse des solutions vis à vis des entrées. Une première piste envisagée est d'utiliser un MOEA économe en évaluations pour obtenir un front de Pareto approché [88]. Des approches par modèle de substitution [11] ou des simulations sur données réduites comme cela a été fait pour ce problème de stockage de mémoire Flash sont également possibles.

3.4 Placement d'objets dans le Fog avec MILP et approche *diviser pour régner*

3.4.1 Problématique

Les objets de l'IoT produisent et consomment des données en continu, données stockées soit sur un serveur Cloud (centralisé), soit sur divers équipements réseau locaux, dans le Fog. Le but est de minimiser la latence globale des accès aux données, en tenant compte de l'endroit où elles sont produites, stockées et consommées.

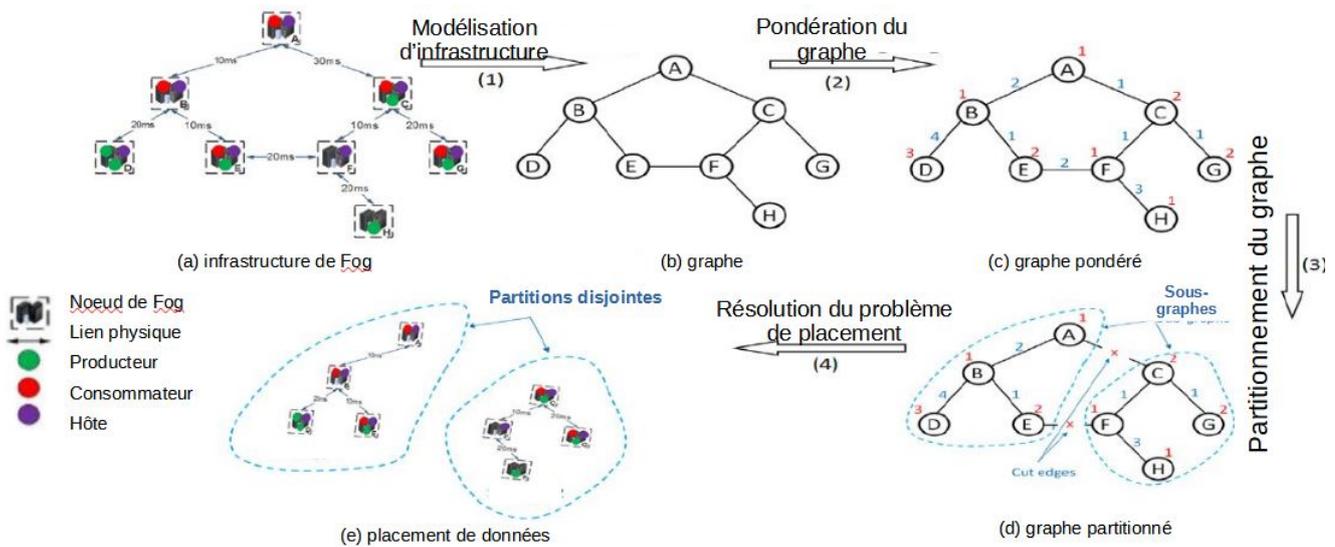


FIGURE 3.5 – Approche par partitionnement pour l'affectation d'objets dans le Fog[C15]

Nos travaux sur ce sujet correspondent à la thèse CIFRE de I. Naas [114], menés en collaboration avec Orange Labs à Rennes, et dirigée par J. Boukhobza.

Contributions Nous avons défini une approche exacte avec une formulation MILP du problème de résolution du placement d'un ensemble d'objets sur un ensemble de support de stockage disséminés dans le Fog (Fig. 3.5 (a)). Il est tenu compte des temps d'accès aux données en lecture (par les consommateurs de ces données) et en écriture (par les producteurs de ces données). La modélisation de l'infrastructure en graphe permet de déterminer les plus courts chemins entre paires de sommets. La latence sur ces chemins est ensuite évaluée par simulation [70]. Eviter la remontée des données sur un serveur central (Cloud) permet d'économiser des ressources de bande passante et d'améliorer la latence globale des accès. Cette approche en 2017 était parmi les premières à s'intéresser au placement efficace des données dans le Fog [C14]. Les travaux de l'époque utilisaient les capacités de stockage du Fog en relais d'un stockage dans le Cloud

[2] [145]. Une approche par partitionnement a aussi été proposée pour résoudre les instances de taille industrielle, avec plusieurs milliers de données à placer. Elle est basée sur une modélisation sous forme de graphe, auquel est appliqué une méthode de partitionnement (Fig. 3.5(d)). Les chemins les plus empruntés pour les transferts sont pondérés avec une valeur forte (Fig. 3.5(c)), afin d'éviter au maximum leur déconnexion lors du partitionnement. La résolution de la formulation MILP est alors appliquée à l'ensemble des sous-graphes générés pour obtenir un placement des données par partie (Fig. 3.5(e)).

Le problème de placement posé correspond à un *Generalized Assignment Problem* (GAP) [34], dont la formulation linéaire est la suivante :

$$\begin{aligned}
 & \text{Minimize} && \sum_{i=1}^n \sum_{j=1}^m v_{i,j} \cdot x_{i,j} \\
 & \text{Subject to} && \\
 & && \sum_{j=1}^m x_{ij} = 1 \quad \forall i \in I = [1..n] \\
 & && \sum_{i=1}^n s_{ij} \cdot x_{i,j} \leq c_j \quad \forall j \in J = [1..m] \\
 & && x_{i,j} \in \{0, 1\} \quad \forall i \in I, \forall j \in J
 \end{aligned}$$

où n objets (de taille s_i) sont à placer sur m endroits (de capacité c_j). Le coût v_{ij} associé au positionnement de i sur j correspond à la latence des accès entre le producteur et les consommateurs de l'objet i pour un placement à l'endroit j de i . $x_{ij} = 1$ dans ce cas et vaut 0 sinon. Un algorithme de plus court chemin [57] a été utilisé pour calculer la latence optimale des transferts entre points du réseau et en déduire les valeurs v_{ij} . Les valeurs de latence pour chaque segment de réseau sont déterminées grâce à l'environnement de simulation IFogSim [70] pour chaque cas de test. Un cas de test générique de *smart city*, avec une architecture de réseau hiérarchique (voir fig. 3.5) a été utilisé pour les expérimentations.

Plusieurs stratégies de placement de données peuvent ainsi être comparées avec les solutions obtenues par programmation linéaire [C16].

3.4.2 Résultats

Le problème IP GAP peut être résolu directement à l'aide d'un solveur MILP comme CPLEX [136], mais les temps de résolution augmentent exponentiellement avec la taille de l'instance (nombre de données et de localisations possibles). Une première approche de partitionnement du problème basée sur la hiérarchie du réseau a donc été expérimentée [C14].

Pour s'affranchir de topologies spécifiques du réseau, une méthode basée sur le partitionnement de graphes [86] a été ensuite développée et expérimentée [C15]. Le processus est illustré par la figure 3.5. Pour minimiser les déconnexions entre lieux de stockage et lieux de production/consommation lors du partitionnement, les arêtes du graphe sont pondérées par le nombre

de plus courts chemins qui les utilisent (étape (2) sur la figure, produisant le graphe (c)). Le poids des sommets correspond au nombre de services s'exécutant sur ce sommet (stockage, production, consommation). Le problème d'affectation est ensuite résolu pour chaque partie du graphe et une stratégie spécifique est ensuite utilisée pour router les données dont les lieux de production, stockage et consommation ont été déconnectés par le partitionnement.

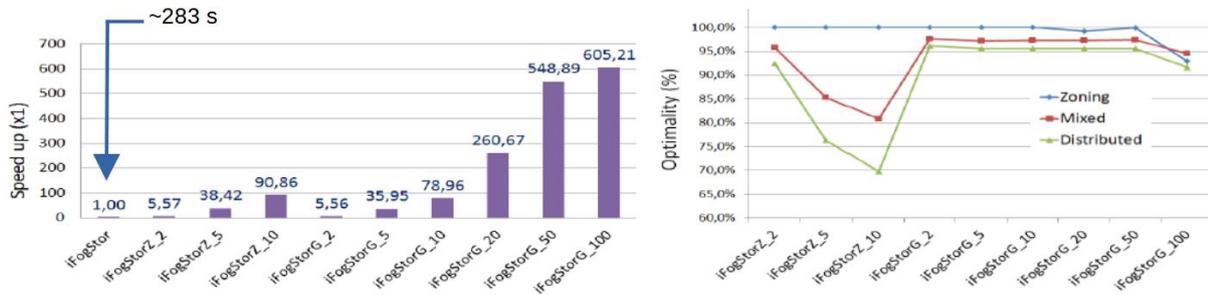


FIGURE 3.6 – Accélération des temps de résolution (à gauche) et perte d'optimalité (à droite) pour des approches par partitionnement du placement de données dans le Fog. Solution de référence exacte calculée par résolution IP en 283 secondes [C15]

La topologie de l'infrastructure réseau est hiérarchique et cette hiérarchie peut être exploitée pour un partitionnement basé sur ces informations métier. La figure 3.6 montre les résultats en termes de qualité et de temps de calcul pour différentes stratégies (zonage hiérarchique Z en 2, 5 et 10 zones, ou par partitionnement (G) jusqu'à 100 parts. Comme le montre la figure 3.6 à gauche, le partitionnement permet d'accélérer les calculs de manière supra linéaire. L'accélération est par exemple de 90 pour un partitionnement par zonage en 10, 260 pour le même partitionnement en 10 mais avec une approche par partitionnement de graphes. Pour ce même partitionnement, la perte par rapport à l'optimal est de moins de 5% avec l'approche par graphe, mais peut être de 30% pour le partitionnement par zones. Les informations métier de zonage ne permettent donc pas dans ce cas de faire mieux qu'une approche purement basée sur les propriétés des graphes.

Le travail s'est poursuivi avec la prise en compte du stockage des répliques de données, avec exigences variables sur la cohérence de ces répliques [R08]. La formulation linéaire du problème a été mise à jour. Avec les répliques, en plus d'une amélioration de la robustesse du stockage par redondance, l'accès aux données est amélioré, un producteur ou un consommateur exploitant la réplique la plus proche. Cependant, le protocole de mise à jour des copies génère lui-même du trafic à prendre en compte lors des calculs de latence par simulation.

Au bilan, ces travaux ont été publiés dans [R08][C14, C16, C15]. Le problème traité étant mono-objectif et assimilable à un GAP, une approche MILP était envisageable facilement pour

la résolution. De plus la flexibilité de l'approche MILP a permis son extension au problème de replication des données. Pour pallier aux problèmes d'extensibilité, nous avons exploité le fait que les données d'infrastructure pour le placement pouvaient elles-mêmes modélisées via un graphe. Nous avons déjà utilisé des techniques de partitionnement de graphe pour construire avec succès une approche de résolution de type *diviser pour régner* dans le domaine de la synthèse logique de circuits FPGA [T01].

3.5 Stockage dans le Cloud

3.5.1 Stockage hybride

Dans la section 3.3, nous avons vu comment optimiser la politique de gestion des accès pour un dispositif de stockage Flash. De nombreux types de support de stockage existent, avec des caractéristiques différentes de latence, de coût d'achat et de consommation énergétique. Ces différents types de support SDD (Flash) ou HDD (magnétiques) sont exploités par les services de Cloud pour stocker les objets de leurs utilisateurs. Ils fournissent à ceux-ci des accès, en allouant les espaces au plus juste en terme de coût, tout en satisfaisant une qualité de service ou *QoS* (latence d'accès, bande passante fournie, etc) qu'ils ont vendu à leurs clients. Ces garanties sont formalisées par contrat au travers de SLA (*Service Level Agreement*). Le non respect de ces SLA entraîne des pénalités financières pour le fournisseur [32].

Les objets stockés peuvent être des données brutes ou associés à des services, comme par exemple l'hébergement d'un serveur Web, avec ses exigences de réactivité (par ex., nombre de requêtes servies par heure). Ces objets correspondent en fait à des machines virtuelles (VM), associées à des ressources physiques (mémoire, stockage SSD ou HDD, calcul CPU ou GPU, réseau) [106].

Des politiques d'allocation des ressources permettent d'optimiser le coût de l'hébergement, en définissant les ressources réellement allouées aux VM, qui peuvent être inférieures à celles prévues au contrat. Le fournisseur, en se basant sur l'historique de consommation d'une VM, peut sous-allouer ses ressources par période, en prenant éventuellement le risque de violer ses SLA et donc de devoir payer des pénalités. Il doit aussi dans ce cas réallouer des ressources dynamiquement, ce qui peut être coûteux (par ex : migration/duplication d'une VM sur une machine physique plus puissante). Cette problématique se retrouve au niveau de chaque fournisseur. Ceux-ci peuvent également coopérer au sein d'une fédération, pour externaliser l'hébergement d'une partie de leurs clients chez d'autres fournisseurs [151].

Nous avons abordé le sujet dans le cadre de la thèse de Djillali Bouthekef [23], dirigée par J. Boukhobza en collaboration avec l'USTBH de Alger. Ces travaux concernaient le choix du type de stockage pour un ensemble d'objets. Djillali avait développé un modèle de coût [C13] pour le stockage différencié (entre SSD et HDD) d'un ensemble d'objets, en tenant compte des

SLA et du workload (dynamique de la charge en termes de différentes ressources associées à chaque objet). Nous l'avons aidé à mettre au point un algorithme génétique pour ce problème de stockage hybride et à calibrer ses paramètres pour le valider.

Ces travaux ont été publiés dans [R07].

Nous avons ensuite réalisé des travaux plus conséquents concernant le stockage hybride dans une fédération de Clouds avec la thèse de Amina Chickhaoui [31], toujours avec J. Boukhobza et en collaboration avec l'USTBH de Alger. Ces travaux sont présentés dans la section suivante.

3.5.2 Placement d'objets dans des Clouds fédérés avec MILP et objectifs multiples

Le Stockage en temps que Service (StaaS, [106]) est l'une des plus importante activités liées au Cloud pour la fourniture d'infrastructures Cloud [31]. Après avoir abordé la problématique du stockage efficace au niveau d'un fournisseur de service dans la section précédente, nous allons voir dans cette partie comment optimiser la coopération entre plusieurs fournisseurs, en se plaçant du point de vue de l'un d'entre eux.

Problématique

Pour améliorer la latence d'accès et la qualité de service apportée aux clients, un hébergeur peut intégrer une fédération de fournisseurs et mutualiser ainsi leurs ressources. Il peut ainsi externaliser le stockage des objets de certains de ses clients, en payant un loyer à ses partenaires, pour économiser ses propres ressources et rapprocher les données du lieu où elles sont utilisées. A l'inverse, il peut tirer profit de ses propres ressources sous-utilisées en hébergeant des données d'autres partenaires de la fédération. Cette approche facilite aussi le passage à l'échelle de l'hébergeur et augmente la disponibilité des données.

Le placement d'objets dans une fédération de Clouds a été optimisé en se plaçant du point de vue de l'un des membres de la fédération. Le coût global d'un placement est complexe. Il prend en compte le placement actuel des objets (vont-ils migrer dans le nouveau placement ?), les informations sur les clients (objets associés, trafic, pénalités associées aux SLA sur la QoS), les informations sur les objets (taille, trafic), les coûts de stockage interne (capacité, performance, prix des différentes classes de stockage) et externe (capacité, performance, prix de stockage, prix de transfert). Ces informations sont utilisées pour produire un nouveau placement des objets. L'approche et les valeurs choisies ont été validées dans [33]. 3 métriques sont mises en place pour évaluer l'intérêt d'un nouveau placement :

- le coût du stockage dans la nouvelle solution, calculé suivant que les objets sont hébergés localement ou pas

- le coût de migration, prenant en compte le fait qu’un objet soit déplacé ou non par le nouveau placement, avec les coûts de transfert éventuels de/vers un hébergeur externe
- la latence associée aux objets en liaison avec ce nouveau placement et évaluée par rapport au trafic sur ces objets avec les clients associés. Cette latence est traduite en coût par les pénalités éventuelles liées à la non satisfaction des exigences de QoS des clients.

Ces facteurs sont souvent contradictoires. Un stockage en interne est a priori moins coûteux, mais peut nécessiter des relocalisations de données et dégrader la latence globale des clients de l’hébergeur si ses serveurs locaux se retrouvent trop sollicités.

Contributions Ces métriques et les contraintes associées pour le placement d’objet dans un Cloud fédéré ont été modélisées par une formulation MILP multi-objectif, exploitée dans une matheuristique utilisant NSGA II. Pour la phase MOEA, nous avons développé un opérateur de crossover spécifique (avec réparation de solutions invalides), pour une exploration efficace de l’espace des solutions. Le travail a ensuite été étendu pour la réplication de données. Comme présenté dans [R03], de nombreux travaux ont contribué au problème du placement dans les Clouds centralisés et distribués (par ex. section précédente et [95] pour les Clouds centralisés, [69][81] pour les Clouds distribués). L’état de l’art considère de nombreux objectifs et contraintes différentes, y compris avec des approches MOO. Cependant, les approches restreignent les métriques le plus souvent au volume de données, plus rarement aux performances en I/O. Les coûts liés aux SLA sont également peu pris en compte.

Approche

Le problème de placement a été exprimé sous la forme d’un programme linéaire en nombres entiers, avec les fonctions objectifs et catégories de contraintes suivantes pour un placement x :

Minimize	$storeCost(x)$	
	$latencyCost(x)$	
	$migrationCost(x)$	
Subject to		
	interne & externe	$storeCapacityConstraints(x)$
	interne & externe	$networkCapacityConstraints(x)$
	Tolerance clients sur les performances	$SLAPerformanceConstraints(x)$
	Unicité du placement	$validPlace(x)$

Les contraintes de capacité de stockage se rapportent aux ressources en disques des hébergeurs, alors les capacités réseau sont liés aux débits disponibles entre hébergeurs. Les contraintes de SLA vérifient la conformité du contrat sur la qualité de service délivrée aux clients.

Une formulation MILP a été produite et utilisée pour une résolution directe d’instances

de petite taille (15 objets placés sur 4 emplacements possibles en 2886 secondes dans [31]) et en produisant des solutions supportées du problème multi-objectif par somme pondérée. Une approche MOO directe utilisant NSGA II ([42], voir section 1.2.3) a aussi été mise en œuvre, avec des opérateurs spécifiques. Une des originalités est l'introduction d'un opérateur de réparation qui permet de rendre valide des solutions invalides produites par un crossover classique. L'implantation de la métaheuristique a été faite en utilisant le framework MOO *MOEA* [59].

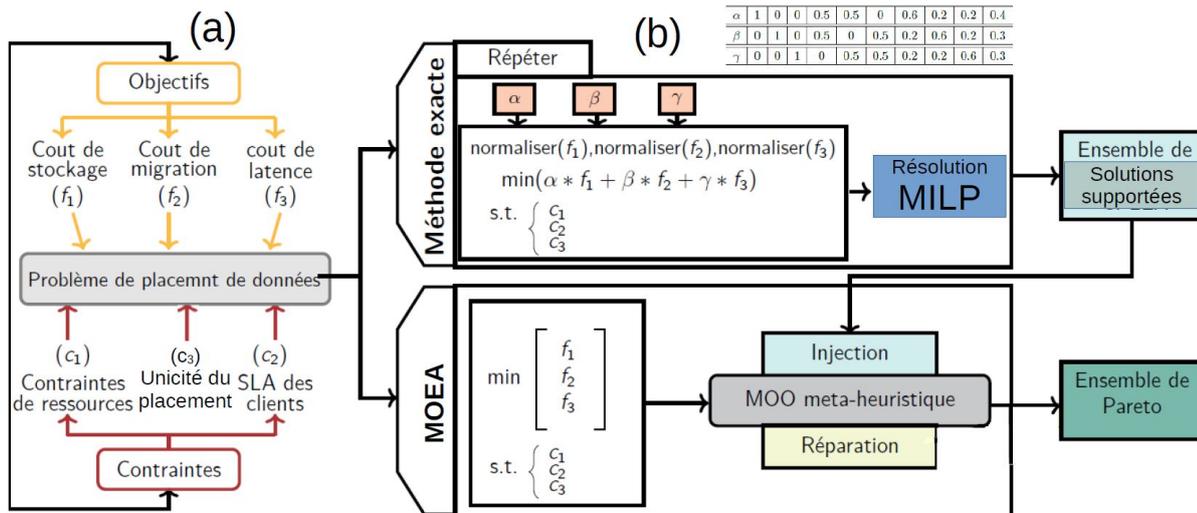


FIGURE 3.7 – Matheuristique pour le placement d'objets dans un Cloud fédéré [31]. A partir du modèle (a), un ensemble de MILP sont résolus en variant les poids (b) pour fournir les solutions supportées de départ pour l'exécution de la partie MOEA (*injection*). Un opérateur de *réparation* est inclus dans le cycle de l'algorithme évolutionnaire.

Les deux approches ont ensuite été combinées en une matheuristique comme le montre la figure 3.7. Un certain nombre de solutions sont calculées en pondérant les 3 objectifs en une seule fonction $\alpha.f_1() + \beta.f_2() + \gamma.f_3()$. Chaque solution se différencie des autres par les valeurs (α, β, γ) utilisées (voir le tableau d'exemple de 10 combinaisons dans la figure 3.7(b)). Ces solutions supportées du front de Pareto sont ensuite injectées dans la population initiale de la partie évolutionnaire de la matheuristique. Ce algorithme peut également être exécuté sans injection, en initialisant la population aléatoirement. De plus, une fonction de réparation de solutions peut être utilisée afin d'explorer plus efficacement l'espace de recherche : les objets qui ne respectent pas les contraintes de placement sont examinés un par un et déplacés à un endroit valide si il en existe un. Dans le cas contraire, la solution reste infaisable. L'ajout de cette fonction de réparation permet d'utiliser des opérateurs de crossover standard comme SBX [39], quitte à devoir vérifier l'invalidité et réparer les solutions infaisables. Cette fonction de réparation examinant les objets invalidant la solution un par un et de manière indépendante, le surcoût en temps d'exécution n'est pas significatif par rapport au temps global d'exécution de

l'algorithme comme le montre les temps indiqués par les colonnes bleues (I) et vertes (IR, ajout de la réparation) quasi-identiques sur la figure 3.8.

Résultats

L'approche matheuristique pose des questions en terme de qualité des résultats produits et en terme d'effort de calcul induit. Les expérimentations présentées dans [R03] ont été réalisées sur un ensemble de jeux de tests comportant entre 15 et 25 fournisseurs et de 100 à 1000 objets (machines virtuelles) placées, avec deux traces de trafic différentes. La qualité des solutions est mesuré par le cumul des hypervolumes. L'outil de résolution MILP utilisé est CPLEX [136], avec ses options par défaut. Le MOEA est NSGA-II, dans la version fdournée par le framework MOEA [59], avec là encore des opérateurs (SBX pour le crossover) et options par défaut.

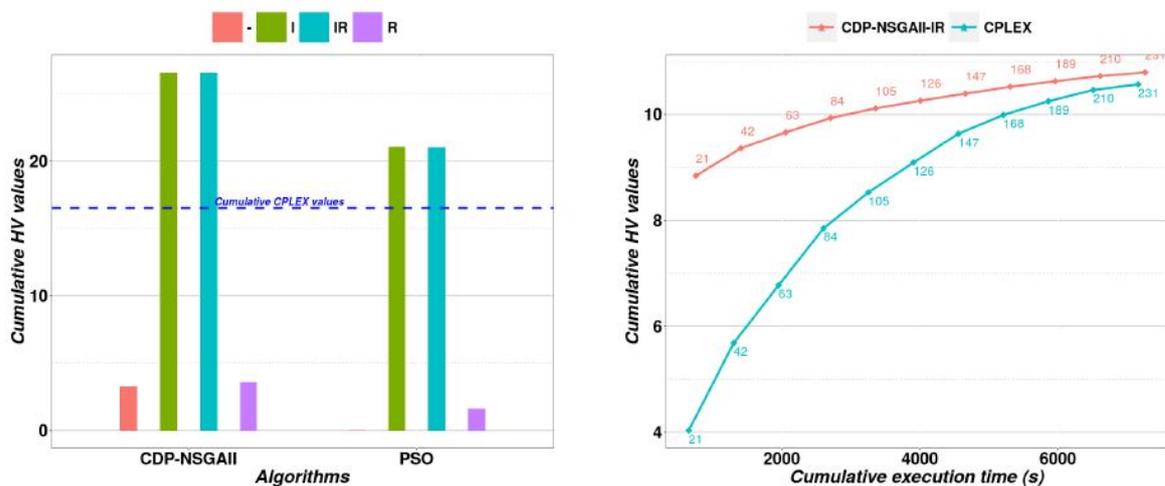


FIGURE 3.8 – Comparaison des résultats (hypervolume, HV) et des temps de calcul (à droite) pour plusieurs approches de résolution. Résultats cumulés pour un ensemble de cas de test [C21]. I et R font référence au fait d'injecter (I) les solutions calculées par la méthode exacte et d'inclure l'opérateur de réparation (R). 2 MOEA sont comparés : NSGA II et PSO (une version de l'algorithme *Particle Swarm Optimisation*). L'hypervolume de référence est celui correspondant à un ensemble de 10 solutions supportées calculées par résolution exacte MILP (trait pointillé bleu à gauche, courbe bleue à droite). A droite, les chiffres sur les courbes permettent de comparer, pour un nombre donné de solutions calculées avec CPLEX, les temps d'exécution et la qualité du front obtenus en injectant ces mêmes solutions dans la matheuristique NSGAI-IR.

Les algorithmes suivant ont été comparés qualitativement (Fig. 3.8) en terme d'hypervolume (HV) cumulé des cas de test : (CPLEX) calcul de 11 à 231 solutions supportées correspondant à un nombre variable de pondérations différentes des 3 fonctions objectifs ; (-) pour la métaheuristique seule, NSGA II ou PSO ; (-I) avec l'injection dans la population initiale des solutions produites par (CPLEX) ; (-R) pour l'ajout de la fonction de réparation ; (-IR) pour l'inclusion

de la fonction de réparation dans la matheuristique (-I).

En terme qualitatif, à la fois pour l'algorithme NSGA II et pour la métaheuristique PSO (version MPSO [102]), l'injection des solutions supportées (option -I ou -IR) permet d'améliorer la qualité du front obtenu par la résolution MILP seule en 10 combinaisons de 3 fonctions objectifs : l'hypervolume correspondant aux 10 solutions supportées trouvées par l'approche MILP est marqué par le trait horizontal à gauche dans la figure 3.8. Cela se constate aussi sur la partie droite de cette même figure par l'écart entre les deux courbes (au dessus pour la matheuristique, au dessous pour l'approche MILP seule).

En ce qui concerne les temps d'exécution, la figure 3.8 à droite montre comment l'effort de calcul peut être modulé, en choisissant le nombre de solutions supportées calculées (chiffres sur les deux courbes). Le temps de calcul de l'approche MILP seule (courbe bleue) est augmenté de la partie métaheuristique (courbe rouge pour la matheuristique). On voit que la partie heuristique permet un gain qualitatif important pour un petit nombre de solutions injectées. Pour 21 solutions, l'hypervolume est multiplié par 2.2 par rapport à la valeur du front MILP initial. Il faut avec CPLEX 100 solutions injectées, avec un temps de calcul bien plus long, pour obtenir la même qualité de front.

L'ajout de répliques a également été prise en compte dans la suite de ce travail. Cela permet à la fois d'améliorer la sécurité des données (redondance) et les temps de latence d'accès en lecture (copie la plus proche). Cependant, ces répliques nécessitent un protocole de mise à jour dont le coût est intégré à la fonction de migration [C21].

Ces travaux ont été publiés dans [R03][C21]. Comme nous le verrons dans la section 4.2.2, l'application d'une méthode de résolution exacte à partir d'une formulation IP peut s'avérer très coûteuse en temps de calcul. Nous avons fait le choix dans ce travail réalisé postérieurement de limiter la recherche de points du front de Pareto à un ensemble restreint de solutions supportées. Ce front partiel est utilisé pour la construction d'une matheuristique et la modulation de l'effort de calcul. Il est aussi utile pour évaluer qualitativement l'approche métaheuristique, comme le montre la comparaison des hypervolumes sur la figure 3.8(a), avec la valeur de référence (trait bleu) associée à ce front partiel. De plus, la mise en œuvre du calcul avec une résolution mono-objectif sur somme pondérée est peu coûteuse à mettre en œuvre. Quand au choix de NSGA II et d'une approche PSO pour la patir métaheuristique de la matheuristique, ils ont été motivés pour la première par la popularité de NSGA II et sa disponibilité au sein de l'environnement MOEA [59] et pour la seconde par l'existence de travaux sur le placement pour clouds fédérés utilisant cette approche [102].

3.6 Conclusion

Nous avons présenté dans ce chapitre des travaux liés au transfert de données et à leur stockage dans le Fog ou dans des Clouds, avec des métriques associées notamment aux temps de réponse des systèmes étudiés.

Grâce à une approche adhoc originale basée sur la théorie des graphes, le premier problème traité (QoS dans les réseaux clos) se résout en temps polynomial. Pour les autres, les temps de calcul pour les instances de taille utile par rapport aux applications visées sont prohibitifs. Une approche MOEA et une parallélisation maître esclave ont été proposées pour l'optimisation de la politique de gestion de mémoire Flash. Les autres travaux présentés dans ce chapitre correspondaient à 3 thèses que j'ai coencadrées. Le placement d'objets dans le Fog et leur réplication avec Islam, ainsi que le placement d'objets dans les Clouds fédérés avec Amina ont été modélisés par des MILP. La difficulté réside toujours dans la taille des instances à traiter, qui ne permet pas une résolution directe des modèles MILP proposés. Avec Islam, nous avons utilisé une approche diviser pour régner sur les données pour réduire la taille des problèmes de GAP à résoudre. Cette approche est similaire à celle que j'avais proposée dans ma propre thèse sur la synthèse logique de circuits FPGA [T01] et basée sur la résolution d'un problème de partitionnement de graphes. Avec Amina, le problème était multi-objectif, et les temps de calcul ne permettaient pas une résolution exacte. Une mathheuristique flexible en termes d'efforts de calcul a donc été proposée. Enfin la thèse de Djillali utilisait une approche évolutionnaire plus classique.

Les 3 thèses ont été réalisées dans le cadre de projets collaboratifs. Islam a fait une thèse CIFRE avec Orange Labs Rennes. Amina et Djillali venaient de l'USTBH, Alger, dans le cadre d'une collaboration de longue date de J. Boukhobza (UBO) et K. Boukhalfa (USTBH), supportée par un PHC Tassili.

L'ensemble de travaux a donné lieu à la publications de 3 revues et 5 conférences avec les doctorants et à 2 revues et 2 conférences pour les travaux menés uniquement avec mes collègues du Lab-STICC.

Des nouveaux problèmes liés au stockage vont apparaître, par exemple avec l'avènement des mémoires non volatiles NVM [25]. L'embarquement des calculs d'optimisation est aussi un champ de recherche prometteur, en particulier dans le domaine des drones autonomes. Cela pose non seulement des contraintes de temps de réponse sur le résultat des calculs, mais aussi des contraintes d'empreinte mémoire et de consommation énergétique.

RÉSEAUX DE TRANSPORT, ROUTAGE ET ROUTAGE EMBARQUÉ

4.1 Introduction

En recherche opérationnelle, le routage est souvent lié à la notion de plus court chemin et les algorithmes de calcul de la théorie des graphes sont connus. Un problème de routage peut également être formulé et résolu par programme linéaire. Cependant, dans de nombreux cas, des contraintes et/ou objectifs additionnels rendent le problème à résoudre plus complexe, et les algorithmes utilisés non polynomiaux. Le problème MOSPP (*Multi Objective Shortest Path Problem*), même sans valuations dynamiques, est déjà NP-difficile [73]. Dans le cas de routage avec des vitesses variables pour les véhicules en fonction du temps, les graphes utilisés pour la modélisation sont dynamiques : la pondération associée à un arc n'est pas fixe et dépend du moment où l'on transite sur celui-ci. Nous avons été confrontés à ce type de problème dans le cadre de travaux sur le routage météo de voiliers de course : la vitesse du bateau dépend de l'angle et de la force du vent, par nature variables dans le temps. La qualité d'un parcours peut également être évaluée selon des objectifs multiples. Dans une modélisation des routes basée sur les graphes, elle n'est pas forcément calculable en tant que somme ou maximum des poids des arcs qui le composent, comme envisagé dans de nombreux travaux sur les plus courts chemins multi-objectifs [61]. Toujours pour le routage météo, la qualité se mesure parfois au nombre de manœuvres qu'impliquent les changements de route, ce qui est lié à la géométrie des chemins empruntés et non pas à leur longueur. Dans le cadre de missions de supervision par un drone, en plus d'une classique métrique distance/temps/énergie, la surface supervisée par le drone à l'aide d'un capteur (Lidar), ne dépend pas que du parcours, mais aussi des obstacles le parsemant. Dans tous ces cas, la théorie des graphes reste utilisable avec des modélisations de routes par des chemins pondérés, mais les algorithmes classiques de calcul plus court chemin sont insuffisants pour résoudre les problèmes posés.

Ces problèmes nécessitent d'utiliser des approches MILP ou métaheuristiques multi-objectifs adaptées. Nous avons commencé à travailler en 2014 sur les problèmes de routage météo de voi-

liers avec G. Guillou et J.P. Babau au Lab-STICC. Ces travaux furent les premiers pour lesquels je me suis penché sur les approches multi-objectifs, en profitant d'un séjour de plusieurs mois au CEMPS de Exeter (GB). J. Fieldsend et R. Everson m'ont aidé à monter en compétence sur le sujet. Ces compétences en routage multi-objectif m'ont permis ensuite de collaborer avec J.P. Diguët à l'UBS. J'ai ainsi participé à l'encadrement de Hand Ouelmokhtar à partir de 2019, en collaboration avec l'université de Bourmedès, Algérie. L'étudiant a développé dans sa thèse deux approches complémentaires (MILP et métaheuristique) pour la résolution d'un problème de supervision de ports par un drone.

La première section est dédiée à la présentation brève de 2 collaborations que j'ai portées. Elles ont plusieurs points communs :

- il s'agit de construire un réseau de transport, et non pas du routage
- une formulation MILP a été proposée et exploitée
- ce sont des collaborations internationales sur plusieurs années, la première avec une chercheuse de l'université de Exeter, l'autre avec un enseignant-chercheur de Dalarna, Suède

Dans la section 4.3, nous détaillons nos travaux sur le routage météo de voiliers. La problématique de la supervision de ports par un drone et l'approche proposée dans les travaux de Hand font l'objet de la section 4.4, avant la conclusion de ce chapitre.

4.2 Réseaux de transport

4.2.1 Réticulum endoplasmique avec MILP et Branch & Cut

En coopération avec Congping Lin à l'université de Exeter, GB, et avec la participation de R. Euler, Lab-STICC/UBO, nous avons simulé in silicio la création du *réticulum endoplasmique* à la surface des cellules de feuille de tabac, qui sert à faire circuler les nutriments vers l'intérieur de celle-ci. Comme le montre la figure 4.1, il existe des points de jonction fixes, avec des propriétés topologiques particulières. En connaissant la localisation de ces points, les biologistes ont émis l'hypothèse que le réseau est construit en minimisant sa longueur totale [R06]. Cela rapproche le problème d'un problème d'arbre de recouvrement [121]. Des contraintes supplémentaires de planarité du réseau et de respect du degré de connexité des points sont également à prendre en compte. Certains points doivent être connectés à 3 autres impérativement, en étoile. D'autres peuvent avoir une ou deux connexions.

Contributions L'ensemble des contraintes peut s'exprimer à l'aide de la programmation MILP et la fonction objectif à minimiser est la longueur totale du réseau. Cependant, les contraintes sont trop nombreuses pour être intégrées directement. Une résolution LP itérative de type *branch and cut* permet d'ajouter différents types de contraintes d'intégrité, de connexité

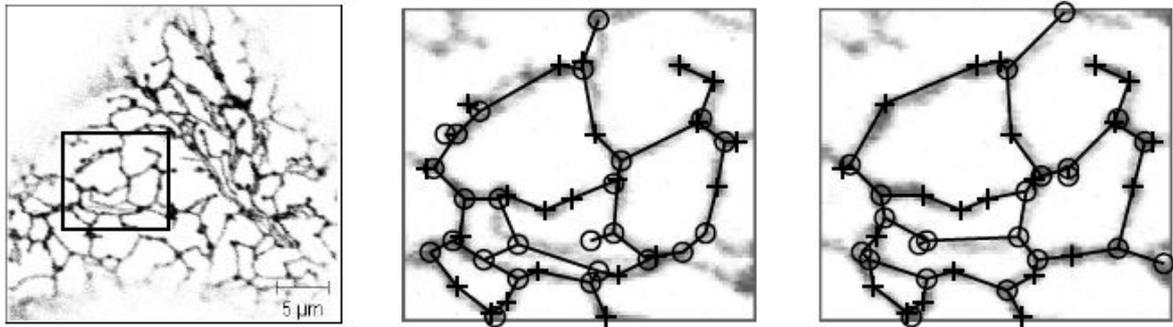


FIGURE 4.1 – Un réseau vivant de transport de nutriments : le réticulum endoplasmique (à gauche) et des simulations de sa construction (2 images de droite)

et de planarité. Cela permet de résoudre à l'optimal des instances de grande taille (problème de 500 nœuds résolu en 40 s, [C08]). Les travaux ont ensuite intégré la dynamique d'évolution du réseau, en prenant en compte les forces physiques déplaçant les points de jonction. Les études [67][99] montraient la ressemblance de la construction avec des arbres de Steiner ou des arbres de recouvrement minimum, mais à notre connaissance, aucune approche MILP pour prendre en compte les contraintes topologiques du réticulum en minimisant sa longueur n'avaient été publiée.

Ces travaux ont été publiés dans [C08, R06]. Ils illustrent les possibilités offertes par les approches MILP pour la modélisation de problèmes variés mais aussi l'expertise et le temps nécessaires à leur exploitation effective. Les modèles MILP proposés nécessitent un savoir-faire que l'on trouve chez les praticiens de la recherche opérationnelle mais sans doute plus rarement chez les biologistes. De plus, la méthodologie de résolution LP a nécessité un travail bibliographique ainsi que l'implémentation (programmation via l'interface en C de CPLEX) de 4 méthodes de séparation différentes (recherche de coupes [C08]) et de manipulation du graphe sous-jacent du problème. La modélisation IP permet de résoudre directement un jeu de cas de tests en 102.9 s environ par cas. Ce travail complémentaire avec l'approche LP réduit le temps de résolution moyen à 0.33 s.

4.2.2 Transport multi-modal avec MILP et méthode MOO ϵ -constraints

Nos premiers travaux en collaboration avec Pascal Rebreyend à l'université de Dalarna en Suède ont porté sur la résolution de problèmes (mono-objectif) de P-median [125] de grande taille à l'aide d'une métaheuristique couplée à une recherche locale [C10, CN02]. Dans le cadre du déploiement d'infrastructures (écoles, centres de secours, etc, *centres* par la suite), un problème de localisation avec transport multi-modal multi-objectif a ensuite été modélisé : il s'agissait de placer un ensemble de centres en prenant en compte les moyens d'arriver à ces centres : (i) à

pied si le centre est assez près du lieu d’habitation, (ii) en transport si le centre est trop loin. Les métriques associées étaient la distance moyenne parcourue par ceux se déplaçant à pied et le nombre de personnes se déplaçant en transports.

Contributions Deux approches ont été proposées pour la résolution du problème : (a) un algorithme MOMILP standard (méthode [28], avec une implantation basée sur AIRA [117]) a été utilisé pour trouver l’ensemble de Pareto, jeu de solutions représentant les compromis intéressant entre les objectifs [35] pour des cas de test jusqu’à 200 nœuds et 5 centres ; (b) une approche MOEA couplée à une recherche locale pour des instances à plusieurs milliers de sommets, correspondant à des données réelles sur l’ensemble de la Suède. PAES [87] et NSGA II ont été testés comme base pour la métaheuristique. L’algorithme a été parallélisé, sur le modèle de nos travaux temps réel ([22], voir Section 2.3). Le problème traité et les approches MOO choisies sont similaires à [6] qui prend en compte un coût pour les centres choisis ou [130] qui traite le problème de la dispersion maximale des centres.

Ces travaux ont été publiés dans [OA01, W05]. Plusieurs heures sont nécessaires pour résoudre des cas de test jusqu’à 200 nœuds avec notre approche exacte. Nous nous sommes donc orientés vers une approche métaheuristique pour les cas couvrant toute la Suède. Cependant, d’autres méthodes exactes seraient envisageables comme une approche 2 phases [139], avec un travail de spécialisation de l’algorithme pour la phase 2 de recherche des points du front de Pareto non supportés. Ce travail de spécialisation pourrait s’avérer coûteux et les gains en termes d’applicabilité de la méthode seraient à évaluer.

4.3 Routage de voiliers avec MOEA

Les thèses au Lab-STICC de Goulven Guillou [68] et de Emilien Lavigne [93] ont été réalisées dans le domaine des écuries de course au large, notamment avec la société MerConcept, basée à La Forêt Fouesnant. Un des besoins identifiés pour la course au large concerne l’assistance au routage du bateau, point clé de la course. Les choix de route peuvent être décidés à bord ou effectués par un routeur à terre, qui s’appuie sur des logiciels spécialisés. Les données d’entrée sont le point de départ, le point d’arrivée, les prévisions météo (nous nous restreignons à la force et la direction du vent ici), et les *polaires* des voiliers en course. Une polaire est une table qui indique, pour un angle et une force du vent par rapport à la direction souhaitée, la vitesse maximale que peut atteindre le navire. Intuitivement, la vitesse sera plus élevée avec un vent fort dans le dos (angle de 180^0) qu’avec un vent faible de face.

L’algorithme traditionnel de routage est le calcul des isochrones [72]. Il a pour objectif de déterminer un chemin minimisant l’ETA (*estimated time to arrival*), i.e. trouver le chemin permettant d’arriver en tête. Il est illustré sur la figure 4.2. En définissant un laps de temps Δt ,

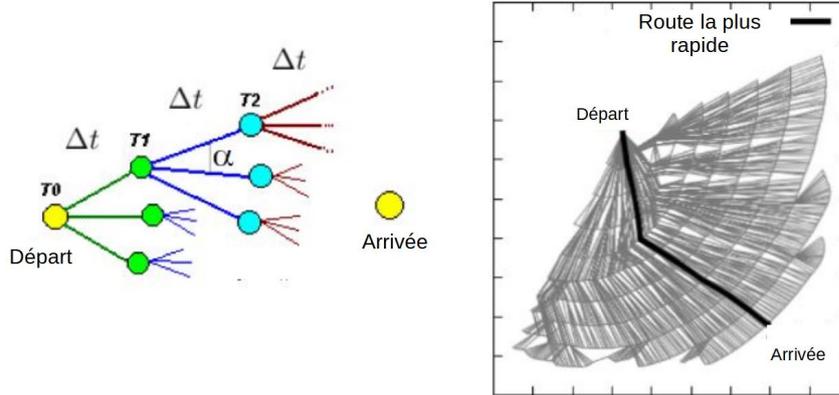


FIGURE 4.2 – Routage par isochrone, principe et résultat [C17]

on calcule suivant une série de directions choisies (avec un pas d'angle α entre elles) les points atteints en Δt unités de temps. L'ensemble des points atteints forme une isochrone. Le processus est répété jusqu'à ce que l'isochrone obtenue soit voisine de la destination finale.

En plus du temps de parcours, d'autres objectifs parfois contradictoires sont à prendre en compte. Par exemple, la notion de risque est liée au mauvais temps éventuel, qui peut entraîner des avaries matérielles et de la fatigue humaine. Les bateaux sont de plus en plus performants mais aussi fragiles (par ex. les foils des grands multi-coques de la classe Ultim). Les routes choisies peuvent ou non générer un nombre important de manœuvres, changement de voiles ou d'allure, et donc être plus fatigantes pour un skipper en solitaire. La *robustesse* des routes est aussi un facteur important. Plusieurs modèles de prévision coexistent et donnent parfois des résultats divergents. La route suivie est considérée comme robuste si elle est résiliente en cas d'écart entre la météo annoncée et celle réellement subie.

Contributions Pour répondre à ces problématiques, nous avons développé une approche MOEA qui prend en compte, en plus du temps ETA, les facteurs de risque humains et matériels. Une fonction dite de *stress* a été définie en combinant 6 facteurs liés aux manœuvres et à la force du vent en une somme pondérée. Ces facteurs ne sont pas généralement pris en compte dans les approches MOO existantes qui se concentrent sur le routage météo de cargo [143]. De manière générale, le routage en mer (voiliers, drones) correspond à la définition d'une suite de points (*waypoints*), éventuellement contraints par une grille et donc un graphe de déplacement sous-jacent [30].

Dans notre cas, une approche par routage dans un graphe ne convenait pas (objectifs non modélisables par des poids sommables pour définir la qualité d'un chemin).

L'algorithme MOEA choisi est PAES [87]. Un opérateur de mutation multi-formes spécifique

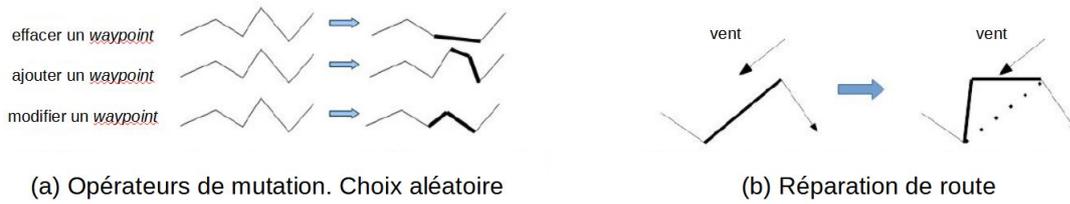


FIGURE 4.3 – (a) Opérateurs de mutation pour le routage de voiliers de course et (b) tirage de bord pour routage face au vent [C17]

a été développé, avec une fonction annexe de réparation de solution infaisable. Une route devient irréaliste si le bateau ne peut plus avancer car il est face au vent. Dans ce cas, il faut tirer un bord, comme illustré par la figure 4.3(b). Pour initialiser la résolution, une première solution courante est fournie via la méthode des isochrones.

Pour des courses classiques (traversée de l’Atlantique Nord sur un bateau de la classe IMOCA 60), les résultats obtenus ont montré que la route la plus efficace en temps (fonction objectif ETA) était effectivement également la plus stressante (fonction stress). Ce travail a été publié dans [C17].

Ces premiers travaux ont montré la faisabilité d’une modélisation des aspects liés aux manœuvres et au risque pour le routage de voiliers de course. Nous avons montré par l’utilisation d’une métaheuristique multi-objectif l’intérêt de produire un ensemble de routes représentant des compromis *stress vs temps*. La difficulté de définir des opérateurs de croisement pertinents entre routes a guidé au départ le choix de PAES, qui ne nécessite que de définir un opérateur de mutation. L’algorithme travaille, rappelons le, sur une seule solution courante et non pas une population de solutions.

La validation de l’approche nécessite un travail plus approfondi, avec notamment :

- la création d’un ensemble de cas de tests de référence, éventuellement évalués par un routeur professionnel,
- l’exploitation de ce jeu de test par plusieurs approches métaheuristicques pour par exemple comparer les résultats obtenus avec PAES et ceux d’une autre métaheuristique comme NSGA II. Ces résultats pourraient eux-mêmes être confrontés à ceux d’une approche par calcul de plus courts chemins multi-objectifs [142][30].
- une prise en compte de la robustesse des solutions, en fonction des variations entre ensembles de prévisions météorologiques [76].

Ces travaux vont faire l’objet d’une thèse débutant en octobre 2023 (cf. Section 5.3).

Ce travail a inspiré la version MOEA du problème de routage de drones présenté dans la section suivante, en reprenant les opérateurs de mutation.

4.4 Routage de drones avec MILP et Branch & Cut

Des travaux pour le routage d'un drone de supervision d'une zone maritime ont été réalisés dans le cadre de la thèse de Hand Ouelmokhtar [116], que j'ai co-encadrée avec J.P. Diguet du Lab-STICC. Le drone est muni d'un lidar. Le lidar lui permet de couvrir les points à sa portée sans s'y rendre. Ces travaux permettent de définir le trajet du drone, tout en faisant un compromis entre l'énergie dépensée et la portion de la surface couverte lors de ce trajet. L'énergie dépensée est calculée à l'aide d'un modèle énergétique [116] et dépend principalement de la distance couverte.

Contributions Nous avons proposé une modélisation MILP et une approche métaheuristique du problème. Cette fois-ci le problème est discrétisé en représentant l'espace maritime par une grille comme le montre la figure 4.4. Un modèle IP a été proposé, et sa résolution fait appel aux méthodes de coupe des sous-tournées pour le Voyageur de Commerce [111]. Pour résoudre les instances de taille supérieure à une quinzaine de sommets, une approche MOEA, avec ses deux déclinaisons (une basée sur PAES, l'autre utilisant NSGA II) ont été comparées. Celle-ci inclut une méthode de calcul originale de la longueur des chromosomes représentant les solutions.

Comme constaté dans [R10] beaucoup de travaux sur la planification de missions de drones considèrent uniquement son routage. Les caractéristiques de celui-ci (longueur, coût énergétique, risque, fluidité, etc) sont souvent pris en compte, y compris avec des approches multi-objectifs (par ex. [115][105]). Peu, comme dans [120], y ajoutent une problématique de couverture.

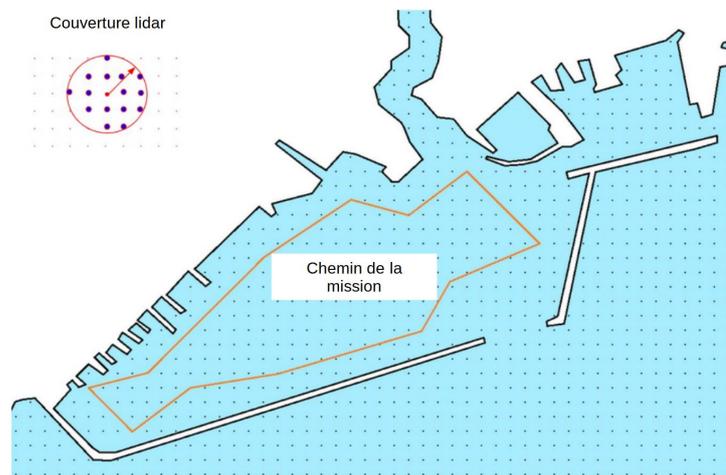


FIGURE 4.4 – Un circuit de routage d'un drone de supervision dans le port de Brest et les points couverts par son lidar à partir d'une position et d'une portée donnée (en médaillon) [W08]

Un modèle MILP, mélange de CSP (*Covering Salesman Problem*) et de TSPP (*Travelling Salesman Problem with Profits*) [53] a été développé, avec la formulation linéaire suivante :

Minimize

$$\text{energy} : \quad \sum_{e \in A} c_e \cdot x_e$$

$$\text{covered} : \quad n - \sum_{v \in V} z_v$$

Subject to

$$\text{tour} : \quad \sum_{e \in \delta(v)} x_e = 2y_v \quad \forall v \in V, v \neq v_1$$

$$\text{subtour} : \quad \sum_{e \in \delta(S)} x_e \geq 2y_v \quad \forall v \in S \subset V, S \neq \emptyset, v_1 \notin S$$

$$\text{cov} : \quad z_v \leq \sum_{v' \in \gamma(v)} y_{v'}, \forall v \in V$$

$$x_e, y_v, z_v \in \{0, 1\}, \forall v \in V, e \in E$$

avec $G = (V, E)$ le graphe correspondant à la grille choisie (avec un voisinage cardinal pour les n sommets), v_1 le sommet de départ de la tournée, $\delta(S)$ l'ensemble des arêtes sortantes du sous-ensemble de sommets $S \subset V$ et $\gamma(v)$ l'ensemble des sommets couverts par le lidar depuis le sommet v . Les poids d'entrée c_e sont calculés avec un modèle de coût du parcours de l'arête e [R09], ce qui permet de calculer le coût de la tournée *energy*. Le nombre de points de la grille non couverts ($z_v = 0$ dans la solution) est exploitée dans la seconde fonction objectif *covered*. La cohérence des tournées est donnée par les contraintes *tour* (un sommet v est dans la tournée ssi $y_v = 1$ et dans ce cas il y a deux arêtes incidentes sélectionnées), et les sous-tournées sont éliminées grâce aux contraintes *subtour*.

La résolution multi-objectif est faite en optimisant une somme pondérée des deux objectifs, énergie consommée *energy* et surface couverte *covered*, pour trouver des solutions supportées de l'ensemble de Pareto. La résolution se fait en utilisant directement le solveur IP, et en coupant les sous-tournées qui apparaissent par l'ajout de la contrainte *subtour* adéquate. L'approche simpliste de résolution IP avec coupe des tournées ne permet de résoudre que les instances de petite taille (moins de 100 sommets), même en utilisant une méthode de *warmstart* [135] pour recommencer la résolution en modifiant les pondérations de la fonction objectif lors du calcul d'une série de solutions supportées [R10].

Une métaheuristique exploitant les opérateurs génétiques de la section 4.3 a par la suite été développée pour résoudre les cas de taille réelle, représentant par exemple le port de Brest avec un maillage de 2000 points. PAES et NSGA II ont été comparés comme algorithmes support pour la résolution. Etant contraint par la grille de routage, les waypoints sont des sommets du graphe et le nombre de waypoints est ajustable. Une tournée est construite en visitant les waypoints choisis en séquence, et en passant de l'un à l'autre par le plus court chemin entre eux dans le graphe. L'originalité de ce travail consiste dans un calcul dynamique de la taille des chromosomes utilisés (nombre de waypoints) pour la représentation d'une tournée. On prend en compte le nombre moyen *avgDiscovered* de sommets découverts lors du passage par un sommet v quelconque (i.e. non encore couverts quand on arrive à v). La taille utilisée pour l'encodage est alors $N = \frac{n}{\text{avgDiscovered}}$, avec n le nombre de sommets du graphe [R09].

La figure 4.5 montre, pour la couverture presque complète ($\text{covered} \geq 95\%$) du cas de test

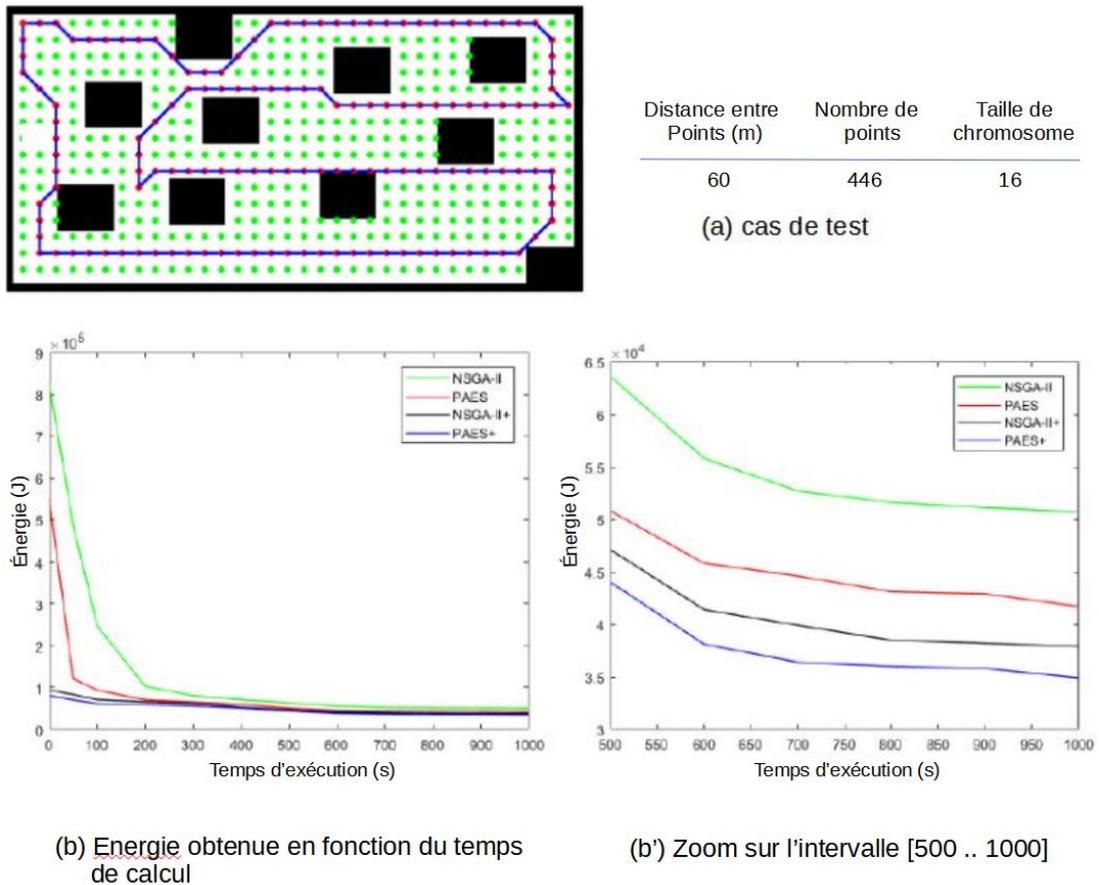


FIGURE 4.5 – (a) Un cas de test avec sa grille de sommets et des obstacles statiques [R10] et ses caractéristiques, (b) et (b') valeur de fonction objectif *énergie* associée aux solutions avec taux de *couverture* $\geq 95\%$ en fonction du temps de calcul pour les 4 algorithmes PAES, PAES+, NSGA et NSGA+

(a), l'évolution dans le temps de la seconde fonction objectif *energy* avec les algorithmes PAES, PAES+, NSGA II et NSGA+. Les algorithmes PAES+ et NSGA+ correspondent à l'utilisation du mécanisme de réduction de la taille des chromosomes utilisés (qui peuvent par défaut comporter jusqu'à n valeurs pour un graphe de taille n). Dans les deux cas, la réduction de la taille des chromosomes permet d'améliorer la convergence des métaheuristiques PAES et NSGA II.

Ces travaux ont été publiés dans [R09][R10].

La validation de l'approche métaheuristique choisie (PAES) a été réalisée par comparaison avec une seconde métaheuristique, NSGA II. Comme pour le routage de voiliers (Section 4.3), la difficulté de définir des opérateurs de croisement pertinents entre routes a guidé au départ le choix de PAES. La modélisation MILP proposée pourrait permettre de mieux évaluer l'efficacité de l'approche métaheuristique, en fournissant des valeurs optimales pour des instances de taille

supérieure à celles pour l’instant résolues. Pour cela, des coupes plus efficaces doivent être utilisées pour l’approche *Branch & cut*. Une bascule vers une résolution en nombres réels avec des coupes linéaires associées serait nécessaire. Les travaux liés à la résolution de formulations LP pour le TSPP [28][18] pourraient sans doute être adaptées pour la résolution de notre problème de supervision de ports.

4.5 Conclusion

Nous avons présenté dans ce chapitre des travaux principalement liés au routage. Au moins une des fonctions objectifs des problèmes étudiés est encore une fois liée au temps, sauf, par exception, pour la simulation du réticulum endoplasmique.

Nos premiers travaux envisageant une problématique multi-objectif concernaient la routage météo de voiliers, au CEMPS à Exeter. Ils n’ont pas été concrétisés immédiatement, mais nous ont permis de développer et d’exploiter notre savoir-faire en matière de MOEA. Les travaux finalement publiés sur le routage de voiliers font une utilisation directe de l’algorithme PAES [C17], avec des opérateurs de mutation adhoc. Dans le même temps, les travaux sur le réticulum endoplasmique ont été l’occasion de développer un savoir faire dans le domaine de la modélisation et de la résolution de problèmes MILP, avec des méthodes de coupe LP diverses. La résolution de problème de P-median multi-objectif a aussi été réalisée dans le cadre d’une collaboration internationale, avec une approche ϵ -constraints.

Ce savoir-faire nous a ensuite permis de développer un modèle MILP et une approche mathématique pour le routage de drones avec les travaux de thèse de Hand Ouelmokhtar. Cependant, cette approche simple avec élimination des sous-tournées sur le programme en nombre entiers ne permet d’exploiter celle-ci que pour des problèmes de très petite taille, avec un maillage de l’ordre d’une quinzaine de points. La encore, on se heurte à la difficulté liée aux temps de résolution, mais aussi au temps de recherche et développement nécessité par une approche plus sophistiquée, pour définir et implanter des méthodes de coupes efficaces. Cela nécessite du temps et une expertise difficile à acquérir, en plus des compétences métier d’un domaine applicatif dans le temps d’une thèse.

Les travaux menés avec mes collègues dans le cadre des projets MOO (avec Exeter) et P-median (avec Dalarna), ainsi que le projet sur le routage de voiliers avec mes collègues du Lab-STICC ont donné lieu à 1 publication en revue, 5 conférences, 1 workshop et 1 revue en open access. La thèse de Hand, en collaboration avec l’Université de Bretagne Sud (J.P. Diguët, Lab-STICC) et l’université de Bourmedès, (Y. Benmoussa), initiée par J.P. Diguët a été valorisée dans 2 revues.

De nombreuses perspectives de recherche sont ouvertes, tant au niveau des aspects métier

(course au large, routage de voiliers, routage de drones et planification de mission, routage embarqué) et aux méthodes (par exemple exploitation des algorithmes de plus court chemin dans les graphes dynamiques).

Nos travaux en cours concernent notamment le routage de drones. Ces perspectives pour le routage, mais aussi dans d'autres domaines comme le temps réel embarqué font l'objet du chapitre suivant.

BILAN ET PERSPECTIVES DE RECHERCHE

Ce chapitre présente tout d'abord un bilan synthétique de mes actions de recherche et de leur valorisation, avant de détailler mes perspectives dans les sections suivantes.

5.1 Bilan des actions de recherche

Ce manuscrit couvre mes actions de recherche depuis 2012. Mes thématiques de recherche concernent principalement l'optimisation combinatoire appliquée dans le domaine des systèmes embarqués. Les 3 grands domaines applicatifs sont les systèmes temps réel, le stockage dans le Fog et le Cloud et le routage. Depuis 2014, une montée en expertise sur la thématique de l'optimisation multi-objectif m'a permis d'apporter des contributions dans les domaines précités en introduisant les approches MOO dans ces domaines, pour résoudre des problèmes variés.

Le temps est un aspect central des problèmes traités :

- Dans de nombreux cas où le temps est important, c'est une fonction objectif, mais pas la seule. Dans les sections 2.2, 2.3, 2.4, 3.3, 3.4, 3.5, 4.2.2, 4.3 et 4.4, nous avons vu comment modéliser certains problèmes d'optimisation et les résoudre avec des approches MOO.
- C'est un facteur limitatif pour l'exécution des algorithmes d'optimisation. Nous avons proposé des stratégies pour utiliser au mieux ce temps et limiter les durées d'exécution des algorithmes, afin de pouvoir traiter des cas de taille utile, avec des approches en 2 phases (sections 2.2, 2.4), branch & cut (section 4.2.1), diviser pour régner (section 3.4), matheuristiques (section 3.5) ou parallèle (sections 2.3, 3.3, 4.2.2).
- Le temps de modélisation du problème et de conception d'une approche pour le résoudre est aussi important. La modélisation MILP permet une approche relativement rapide, avec peu de temps de développement, mais une expertise nécessaire importante pour la modélisation. La qualité des solutions est alors garantie au moins par une borne, mais au prix de temps d'exécution importants. Les approches évolutionnaires sont plus flexibles, nécessitent éventuellement moins d'expertise et ont des temps d'exécution plus abordables, mais au prix d'une précision moindre.

Bilan chiffré Tous ces travaux ont été réalisés en collaboration avec d'autres chercheurs et doctorants. En terme d'encadrement, j'ai supervisé une dizaine de stages de master depuis 2012 et j'ai participé à la supervision de 6 thèses. Les 6 thèses ont été soutenues, dont 3 en 2022. J'ai aussi participé antérieurement à 2014 aux travaux scientifiques de 2 autres thèses. Enfin, 2 autres thèses ont été entamées, chacune avec un étudiant en poste à l'étranger (Houria Dhamani à Alger, Isaac Armah-Mensah à Cape Coast, Ghana), qui n'ont pu dégager suffisamment de temps pour leurs travaux de doctorat et ont renoncé.

Pour les publications, je suis co-auteur de 10 revues internationales (7 sélectives Q1-Q2 de SJR, 3 non sélectives Q3-Q4 de SJR), 1 chapitre d'ouvrage, 21 conférences internationales et 8 workshops internationaux. J'ai aussi donné 4 séminaires invités sur le multi-objectif.

En ce qui concerne les partenariats étrangers, j'ai participé à des collaborations de recherche (intégrant parfois une composante d'enseignement) avec le Ghana (montage de formation sur place, obtention d'un financement de doctorat), l'Algérie (projets PHC et encadrement de thèses avec USTBH et UBMB), la Tunisie (encadrement de thèse), la Croatie (projet et encadrement de stagiaires, colloque de formation), la Suède (projets, cours, encadrement de stagiaires), le Royaume Uni (projet, chercheur invité 8 mois), le Portugal (projet PHC),

Pour ce qui concerne les projets, j'ai été porteur de 4 projets (2 pour des coopérations académiques internationales financées localement (3K€ et 8K€), 1 avec Thalès/Latéral pour 6K€, 1 projet SEA-EU pour 8K€), et j'ai par ailleurs participé à 2 projets PHC, 2 contrats industriels (Orange et ISIS-MPP), et un autre projet SEA-UE (consortium d'universités).

Perspectives de recherche Mes perspectives de recherche sont centrées sur l'adaptabilité et l'adaptation des solutions, en fonction de données d'entrée initiales et de l'évolution de ces données dans le temps dans le cadre de leur utilisation embarquée. Nous proposons :

- lors des calculs initiaux, de prendre en compte les besoins en termes de précision du modèle de calcul et de l'optimalité du résultat, notamment lorsque il est connu que ces données ou ce modèle sont appelés à évoluer. Par exemple, utiliser des modèles approximatifs et raffiner les solutions sur les données effectives d'une mission.
- de gérer l'incertitude sur les données d'entrée, et donc les solutions les plus adaptées, suivant plusieurs stratégies :
 - fournir des solutions robustes, toujours valides et performantes en cas de changement des données d'entrée,
 - produire des solutions multiples, qui peuvent être substituées les unes aux autres suivant les données d'entrée,
 - ou générer des solutions adaptables, que l'on peut mettre à jour avec un effort de calcul réduit
- d'adapter les méthodes et l'effort de calcul aux ressources disponibles, et notamment

lorsque des calculs embarqués ou en ligne sont requis

- après le choix d’une solution parmi un ensemble de compromis, d’exploiter les autres solutions pour prendre en compte la dynamique de l’environnement

Nous verrons dans les sections suivantes sur les projets en cours et les perspectives comment l’exécution, pour un système embarqué, d’un algorithme d’optimisation, peut avoir lieu à différents moments, dans différents scénarios. Cela amène des problématiques d’embarquement des calculs, d’adaptabilité des algorithmes et de temps de réponse de ceux-ci, par exemple dans le cadre de missions de drones.

5.2 Projets en cours

Je participe actuellement à 4 projets financés. Le premier, MEDORAA, a commencé l’an dernier et son financement s’achève en septembre 2023. Il a donné ses premiers résultats publiés [P01] [W08]. Le second, IA-AUV, est une collaboration avec l’université de Split, et son financement s’achève en juin 2023 [P03]. DISPEED, lui, a démarré effectivement fin 2022. La partie sur les aspects MOO du projet seront abordés à partir de septembre 2023. Enfin, le dernier, RESSACH, démarre en mars 2023.

5.2.1 MEDORAA : Décision embarquée exploitant un ensemble de Pareto

Dans le cadre de 2 projets étudiants successifs financées par le laboratoire commun Lab-STICC/Thalès (Evan Flecheau en 2022, Kilian Le Gall en 2023), nous avons commencé à étudier avec ma collègue C. Dezan la possibilité d’exploiter en ligne les résultats du calcul préalable hors ligne d’un ensemble de Pareto. Le projet MEDORAA (*Multi-objective optimization at the EDge for Online and Real-Time self Adaptation of Autonomous vehicles*) utilise l’approche générale illustrée Fig. 5.1.

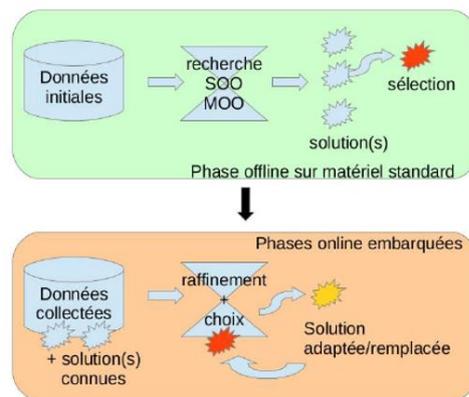


FIGURE 5.1 – Principe de l’approche MEDORAA

Le problème posé dans la thèse de Hand Ouelmokhtar (voir Section 4.4) est de définir la tournée d'un drone marin équipé d'un lidar pour superviser une zone prédéfinie comme par exemple un port. La dynamique de l'environnement considérée dans la thèse est la variation des courants de marée, qui influence la consommation énergétique. Elle ne tient pas compte des obstacles éventuels pouvant survenir après le début de la mission. L'idée, avec l'approche MEDORAA, est d'adapter les solutions obtenues hors ligne pour éviter ces obstacles, de les ré-évaluer, et de choisir parmi ces solutions adaptées au nouvel environnement comment va se poursuivre la mission. Ces premiers travaux ont donné lieu à 2 publications [P01] [W08]. Nous avons montré que pour un ensemble synthétique de 1000 cas de test, 75% des poursuites de missions se faisaient en utilisant une autre solution s' que la solution initiale s_0 mise à jour par un simple contournement de l'obstacle par un plus court chemin. L'utilisation de l'ensemble de Pareto précalculé est donc efficace. Cependant, de nombreux points restent à étudier :

- les temps de calcul sont 100 fois plus importants pour calculer s' par rapport à s_0 . Suivant les cas d'utilisation et le temps disponible pour les calculs, un embarquement d'une sous-partie de l'ensemble de Pareto est possible. L'impact sur la qualité des solutions a été pré-évalué mais reste à mesurer plus précisément en fonction des cas de test.
- L'embarquement du logiciel est à effectuer, avec les contraintes en terme d'empreinte mémoire (sur ARM ou sur FPGA par exemple)
- La généralisation de l'approche pour résoudre des problèmes liés aux drones aériens qui intéressent plus spécifiquement Thalès doit aussi être abordée.

5.2.2 IA-AUV : TSPP pour la définition de missions d'un ROV

Une collaboration avec l'université de Split sur l'optimisation des missions d'un ROV est en cours (Projet IA-AUV) [P03].

L'objectif de cette collaboration est d'examiner les problèmes liés à l'embarquement de composants logiciels et matériels utiles dans le cadre de l'exécution de missions d'un ROV filo-guidé construit par le partenaire croate. La caractérisation du coût en ressource CPU et mémoire, ainsi que le coût énergétique des composants (par exemple un modèle de caméra et un composant de traitement d'image plus ou moins performant) peuvent être pris en compte lors de la définition de missions. Une première modélisation de ces missions basée sur une représentation de type *Travelling Salesman Problem with Profit (TSPP)* a eu lieu (avec 2 stagiaires de M2 croates à Brest en 2021). La venue des croates fin 2022 pour un séminaire de formation et de recherche que j'ai coorganisé a renforcé leur intérêt pour les approches MOO. J'ai aussi effectué deux missions d'enseignement sur place (2021, 2022) avec des cours/séminaires sur l'optimisation multi-objectif. L'approche TSPP sert d'exemple lors de ces séminaires.

Notre partenaire croate est actuellement en train de construire un ROV autonome, ce qui accroît les contraintes, notamment en matière de consommation énergétique. Nous prévoyons

de poursuivre notre collaboration sur la définition des composants logiciels et matériels à embarquer pour la réalisation de missions de ROV. Des demandes de financement du consortium d'universités SEA_UE auquel nos deux universités appartiennent sont possibles.

En parallèle, des étudiants de Licence 3 ont été recrutés en projet (avr-mai 2023) pour faire du développement autour du projet (visualisation de la dynamique de l'algorithme de résolution du TSPP basée sur PAES, substitution par une autre métaheuristique). L'outil développé servira à la fois en séminaire et aussi comme outil annexe ou d'exemple pour nos autres projets sur la thématique MOO.

5.2.3 DISPEED : cybersécurité embarquée pour meute de drones

Le projet DISPEED bénéficie d'un financement AID de 12 mois de post-doc. Il a démarré début 2022. Nous l'avons monté avec J. Boukhobza (ENSTA) et D. Espès (UBO). *DISPEED - Détection d'Intrusion et compromis Sécurité/Performance/Energie, Etude pour les meutes de Drones* a comme objectif de proposer un modèle et des stratégies pour l'exécution sur un ensemble de drones de systèmes de détection d'intrusion embarqués sur des architectures hétérogènes et qui fournissent un compromis pertinent selon la criticité de l'attaque et l'état du système entre rapidité de détection / consommation énergétique. Ces compromis seront obtenus soit grâce à des approches MOO soit avec des méthodes IA. La première phase du projet a consisté à définir et caractériser les différents aspects de la détection en ligne des intrusions :

- modéliser la mission elle même pour une meute de drones (métriques et contraintes sur la qualité de la détection),
- examiner un ensemble de méthodes de détection IDS, basées sur l'apprentissage (forêts aléatoires, réseaux de neurones),
- définir un ensemble de plate-formes d'exécution possibles pour les algorithmes d'IDS (CPU, GPU, FPGA),
- mettre en œuvre des implantations alternatives pour les algorithmes IDS choisis sur les plate-formes sélectionnées.

Chaque solution doit être qualifiée (qualité de la détection, empreinte mémoire, consommation énergétique).

Une première analyse des solutions alternatives (une petite dizaine) permet de définir sans utiliser de méthode MOO les implantations les plus indiquées dans différents cas. Ce traitement manuel ne tient cependant pas compte des réglages paramétriques possibles des différents algorithmes (par exemple, le nombre ou la profondeur des arbres dans une implantation de type Random Forest). Nous devons étudier comment évaluer ces *sous-solutions* alternatives, qui peuvent elles être bien plus nombreuses. L'implantation des solutions est un processus long et souvent manuel. Il n'est pas envisageable à ce stade de les réaliser toutes, mais des méthodes

d'évaluation approximatives (*surrogate*) sont peut être utilisables (voir Section 5.4.2 ci-après). Un mécanisme de décision sera également à définir et à embarquer.

5.2.4 RESSACH : réseau de surveillance acoustique

Le projet ANR ASTRID RESSACH (*RESeau de Surveillance ACoustique Hétérogène et auto-adaptatif*) vient de démarrer en mars 2023. Il a pour partenaires l'Ecole Navale, le SHOM (service hydrographique de la Marine) et le Lab-STICC. Il a pour but la construction et le déploiement d'un réseau de surveillance acoustique sous-marin. Les composants d'écoute et d'analyse et le positionnement du réseau doivent être définis en tenant compte des caractéristiques de propagation et de la dynamique de l'environnement.

Les verrous scientifiques sont les suivants :

- Établissement de règles d'apprentissage de l'environnement physique par intelligence artificielle afin de prédire l'impact de ce dernier sur les performances de détection
- Prise en compte de l'environnement pour la recherche d'une configuration optimale du réseau
- Mise en place de méthodes de classification automatiques et d'aide à la décision pour l'évaluation du niveau d'alerte associé
- Choix des méthodes et leur implémentation sur des systèmes embarqués pour garantir une capacité de calcul avec la frugalité énergétique requise pour une application opérationnelle

Nous allons intervenir fin 2023/début 2024 sur la phase de configuration optimale du réseau, en tenant compte des facteurs d'efficacité de la détection, de coûts (financier, énergétique, etc) du déploiement et de la robustesse en fonction de la dynamique de l'environnement. Le support d'un post-doctorant pour un an est acquis.

5.3 Projet à court terme : OMER, routage de voiliers

Les projets MEDORAA et IA-AUV, dont la période financée est en cours d'achèvement pourraient se poursuivre. De plus, je suis à l'initiative d'un dépôt pour 2023 avec Goulven Guillou et Valérie-Anne Nicolas d'une demande de financement de thèse (Contrat Doctoral Etablissement et financement régional) concernant l'Optimisation Multi-objectif pour le Routage en mer (OMER) pour le routage maritime multi-objectifs et l'aide à la décision pour des voiliers de course. Le financement a été validé et le doctorant, Killian Le Gall, commencera sa thèse en octobre 2023.

Le routage maritime est utilisé pour déterminer la trajectoire de bateaux ou de drones à moteur, mais aussi celles de voiliers, notamment de course. Ce routage est dit *météo* car les

conditions de mer influent de manière déterminante sur le coût en énergie et le temps de parcours et doivent donc être prises en compte (courant, vent, hauteur des vagues, marées, etc).

Dans le domaine de la course au large, d'autres facteurs comme le risque associé au passage dans des zones avec des mauvaises conditions météorologiques, les incertitudes sur les prévisions, des exigences quant aux manœuvres à effectuer suivant la configuration de la route influencent aussi la prise de décision. Le calcul du routage doit donc prendre en compte des critères multiples en tenant compte de l'incertitude liées aux données et à leur dynamique. De plus, dans le cadre de la course à la voile, des facteurs stratégiques externes au processus d'optimisation sont aussi à prendre en compte, comme par exemple la position d'un concurrent dans une course. Enfin, et toujours par rapport à la prise de décision et à la dynamique de l'environnement, la problématique de l'aide à la décision et de l'ergonomie des outils associés, utilisés dans des conditions difficiles lors de la course au large, sont aussi à considérer.

Verrous scientifiques La modélisation du problème de routage doit se faire en modélisant l'environnement et le calcul des caractéristiques d'une route. Des modèles existent pour le calcul du temps de parcours et son optimisation, avec des algorithmes par exemple de calcul d'isochrones basés sur des fichiers météo et des fichiers dit *polaires* caractérisant les performances des bateaux en fonction de la force et de la direction du vent. Mais en ce qui concerne la prise de risque, les modèles sont à définir et valider. De même, pour ce qui concerne le coût humain en terme de fatigue des manœuvres, un travail de modélisation est aussi à réaliser.

Les méthodes d'optimisation de route, basées sur des métriques définies en levant le verrou précédent sont également à explorer. Est-ce que des méthodes multi-objectifs standard peuvent suffire ? Il faut être capable de gérer l'incertitude liée aux données d'entrée soit en définissant des métriques reflétant cette incertitude, soit en utilisant des algorithmes prenant en compte nativement le fait que les données d'entrée soient incertaines, et fournissant ainsi des résultats robustes.

En liaison avec la notion de robustesse, il faut réfléchir à des indicateurs de fiabilité des solutions, permettant d'estimer les intervalles de valeurs d'entrée, notamment météo, pour lesquelles les solutions restent applicables.

La dynamique des données d'entrée, et donc la validité des solutions, vont impacter la politique de mise à jour des routes. Il faut définir une fréquence de calcul, en liaison avec le rafraîchissement externe (fichiers météo reçus) et local (mesures embarquées) des données de calcul. Le coût de ces calculs (en temps et en énergie) va aussi influencer la fréquence de mise à jour. Les besoins du skipper, les capacités de calcul embarquées et la précision jugée utile auront aussi un impact sur le coût et la qualité des routes produites.

La problématique de la validation des méthodes et approches utilisées est aussi à aborder. Une méthodologie et des données de test doivent être définies pour mesurer l'efficacité des algorithmes proposés et des solutions calculées. Les jeux de test doivent être construits en utilisant des données les plus réalistes possibles, tant au niveau de la météo que au niveau des caractéristiques des bateaux. Hors ces dernières, en particulier sur les bateaux de compétition, ne sont pas disponibles publiquement. Un travail de collaboration avec les acteurs du domaine est donc nécessaire.

Le problème de l'extensibilité des méthodes et outils est à explorer, pour estimer dans quelle mesure les problématiques de routage météo de voiliers en course peuvent être adaptées pour le pilotage d'autres navires, comme les voiliers cargo ou les drones marins, en substituant par exemple aux métriques spécifiques à la conduite d'un voilier des métriques liées à son inclinaison ou à sa consommation énergétique.

Contributions Cette thèse se propose d'explorer ces problématiques en utilisant des approches issues de l'optimisation combinatoire multi-objectif. Les contributions attendues sont concernent à la fois la modélisation et l'optimisation du routage d'un coté, et les aspects logiciels et outils de l'autre. Pour les premiers, on a :

- Compléter le travail fait dans [C17] et qui utilisait une première approche basée sur un algorithme évolutionnaire multi-objectif (MOEA) et une métrique de gestion du stress liée aux manœuvres. Des solutions alternatives MOEA existent [35], et d'autres, liées aux approches multi-objectifs sur des calculs de plus courts chemins dans des graphes multi-valués sont possibles [60]. Dans le même domaine de la théorie des graphes, les travaux sur les plus courts chemins dans des graphes dynamiques seraient utilisables [38, 30], et ces approches sont combinables comme dans [142, 1]. Cela permettrait d'évaluer comparativement les deux classes d'approches, métaheuristiques et basées sur la théorie des graphes.
- Produire un modèle affiné des objectifs d'optimisation liés à la gestion du stress et des manoeuvres, notamment par une étude bibliographique poussée et une interaction avec les acteurs du domaine, pour prendre en compte au mieux les informations du domaine métier.
- Etudier et proposer des méthodes de gestion de l'incertitude sur les données d'entrée, avec soit l'exploitation de méthodes de calcul basées sur la notion de scénario météo [79, 1], soit sur des méthodes d'optimisation prenant en compte nativement le caractère dynamique des données [54].

Pour la partie validation et outils, les contributions attendues sont les suivantes :

- Construire des scénarios d'utilisation des méthodes, avec une définition de la fréquence de mise à jour des routes, automatisée ou sur demande, en exploitant les calculs déjà effectués, suivant une précision et dans des délais de calcul prédéfinis. Ceci doit être fait également en fonction des capacités de calcul embarqué ou distant.
- Concevoir et mettre à disposition des outils de restitution des résultats, pour l'aide à la décision [90, 9]. L'aspect ergonomique des outils serait à discuter avec les acteurs du secteur.
- Mettre au point un jeu de tests et de validation des méthodes, et le rendre disponible pour la communauté. La généralisation des méthodes et outils élaborés permettrait d'étendre le spectre d'utilisation des résultats obtenus, pour passer par exemple du domaine de la course à la voile à celui de la navigation des drones marins.

5.4 A moyen terme

Les thématiques suivantes ne sont pas encore lancées, mais se situent en articulation avec des projets terminés. Ces thématiques pourraient faire l'objet de demande de financement de thèse.

5.4.1 Résolution exacte pour la définition de missions de drone marin

En matière de routage de drones, je voudrais revenir sur les problématiques de résolution efficace du MILP associé à la thèse de Hand Ouelmokhtar. La première approche présentée dans [R10] utilisait une formulation IP et des coupes de sous-tournées uniquement. Avec une résolution LP, bien plus rapide, des méthodes de coupes des solutions non intégrales, utilisées dans le projet sur le réticulum endoplasmique [47, 71, 10] (cf section 4.2.1). Des techniques similaires en provenance du champ voyageur de commerce avec profits (TSPP, par exemple les travaux de C. Filippi [55, 56] et ceux de J.F. Bérubé [28]), pourraient être adaptées et permettre une résolution efficace pour des problèmes de taille raisonnable.

De plus, des méthodes existent pour la résolution de BOMILP (Bi-Objective MILP, [140, 29, 118, 28]). A part pour le problème du transport multi-modal, je ne les ai pas beaucoup exploré, car leur applicabilité pour les problèmes de grande taille reste à évaluer. Je compte également profiter des résultats engendrés sur la thématique *routage de voiliers* (Projet OMER, section 5.3) pour généraliser l'utilisation des méthodes de routage dynamique (dépendant de l'état de la mer) pour l'adapter aux spécificités des drones marins : leur routage dépend également de la météo marine, mais avec un impact plus conséquent de la houle et des courants. Le périmètre de la thèse de Hand Ouelmokhtar se limite à la prise en compte d'un modèle statique de courants, en appliquant l'algorithme d'optimisation à des dates précises, avec des courants constants. Ces travaux rejoindraient la thématique plus large du routage de bateaux à moteur,

et notamment de cargos et porte-conteneurs, pour lesquels la consommation de carburant est un objectif d'optimisation majeur (voir par exemple [64] pour des travaux récents). Nos partenaires algériens sont prêts à continuer la coopération sur ce thème avec un projet CampusFrance Tassili et des financements de thèse.

5.4.2 Méthodes par substitution de modèle pour l'exploration d'architectures temps réel

Je voudrais investiguer le domaine des modèles et méthodes de substitution (méthodes *surrogate*). Une littérature abondante existe sur le sujet [58]. Des applications dans le domaine de l'avionique par exemple permettent de construire des fonctions objectives approximatives par apprentissage et de les utiliser dans un cadre multi-objectif [11]. Je n'ai pas encore exploité ces approches en terme de recherche. Plusieurs axes d'utilisation seraient possibles, appliquées aux thématiques de recherche passées ou futures.

Dans un premier temps, et pour ce qui concerne les problèmes pour lesquels la fonction d'évaluation des solutions est coûteuse, il faut explorer les mécanismes de substitution à mettre en jeu lors de l'exploration d'un espace de recherche : à quel moment, et pour quel effort de calcul basculer d'une évaluation approximative à une évaluation fine ? Des mécanismes généraux sont-ils utilisables ? Des mécanismes spécifiques doivent être mis en place ? etc.

Un domaine applicatif concerne l'ordonnancement temps réel : dans les travaux de R. Bouaziz [22] et I. Atchadam [7], une solution est caractérisée suivant une ou plusieurs fonctions objectifs (nombre de préemptions, nombre d'échéances ratées, etc) via simulation. Hors ces simulations s'avèrent parfois très coûteuses en temps de calcul, surtout lorsque elles doivent avoir lieu sur un intervalle de faisabilité long pour obtenir des résultats garantis [66] (c'est nécessaire notamment pour les systèmes temps réel critiques). Réduire l'intervalle de simulation peut se faire dans certains cas en dégradant de manière minimale la fiabilité des résultats. Un modèle de substitution pourrait alors permettre de survoler rapidement et efficacement l'espace de recherche, puis de rebasculer vers des évaluations à résultat garantis pour exploiter les zones les plus prometteuses.

Dans un second temps, d'autres aspects de l'utilisation des méthodes de substitution sont à explorer. Elles peuvent être utilisées pour diminuer les temps de calcul quand la précision n'est pas essentielle. Ces méthodes comme sont aussi des alternatives pour les calculs embarqués, en nécessitant potentiellement moins de ressources.

Comme pour la thèse de I. Atchadam, un financement mixte Contrat doctoral/financement régional pourrait être sollicité. Une alternative au financement si le domaine applicatif est l'ordonnancement temps réel, serait de solliciter un des programme commun Thalès/Lab-STICC.

5.4.3 Répartition d'équipes sportives en poules de championnat

Un petit projet sur l'affectation des poules de championnats a été mené à bien sur les années 2019-2021, avec la Fédération de Football de Loire-Atlantique. Ce projet a été initié à l'occasion d'un premier stage de L3, et prolongé par un second stage l'année suivante au sein de la Fédération.

Des solutions MILP et métaheuristique ont été développées et partiellement testées mais le problème de l'évaluation de la qualité des solutions générées est posé. Des problématiques similaires ont été soulevées et des approches MILP sont envisagées [124]. Nous envisageons également des approches s'inspirant de problèmes liés à la recherche de cliques dans un graphe [119], sachant que des contraintes particulières sur la constitution des poules seraient à prendre en compte (mixité des districts, rencontres exclues, etc).

De plus, le mécanisme de préchargement de solutions comme base pour l'algorithme du simplexe ([135]) dans le solveur MILP pourrait permettre de combiner une approche heuristique de type algorithme génétique, avec une résolution accélérée du MILP.

Enfin, une dernière piste de recherche pour les problèmes de grande taille serait d'utiliser des approches diviser pour régner, soit en partitionnant l'espace géométriquement, soit partitionnement le graphe des distances, avec la recherche d'une méthode de raffinement des solutions. Nos contacts à la ligue de football des Pays de Loire se sont montrés réceptifs aux financements de stages de Licence et de Master, voir de contrats de prestation courts. Il est cependant trop tôt pour envisager avec eux un financement de thèse CIFRE.

5.5 A plus long terme

La recherche de compromis qualité du résultat/effort de calcul m'intéresse. Le mélange entre méthodes d'optimisation combinatoire classique et intelligence artificielle est également certainement un aspect à développer.

Le problème de la précision des solutions se pose en permanence. L'effort de calcul consacré à un problème doit pouvoir être modulé en fonction du temps de calcul disponible, mais aussi en tenant compte de la pertinence des données d'entrée. A quoi bon résoudre à l'optimal un problème dont les entrées ne reflètent que très imparfaitement le problème réel. Par exemple, la collecte des temps de latence utilisés pour l'optimisation du placement des objets dans la section 3.4 est basée sur une simulation du réseau, par nature incomplète et donc inexacte.

Les liens entre IA et recherche opérationnelle sont multiples mais peu exploités pour le moment (voir par exemple la revue [83] ou le chapitre consacré à ce sujet dans [104] ou [85]). L'utilisation des méthodes de l'apprentissage automatique et de l'apprentissage profond pourraient

permettre d’optimiser les nombreux paramètres régissant les métaheuristiques. Par exemple, dans la thèse de I. Atchadam, un algorithme à deux phases (gros grain et grain fin) est utilisé pour réaliser l’exploration de l’espace de recherche. Le nombre d’itérations dédiée à chaque phase est décidé arbitrairement. Ce paramètre, ainsi que le nombre total d’itérations à effectuer, pourraient être calculés en fonction du profil du problème. Des travaux comme [17] ont déjà abordé le problème, et les conférences spécialisées comme LION [100] y sont dédiées. Des conférences plus généralistes comme EVO consacrent également des sessions au sujet (voir le programme 2022 [50]). J’aimerais aussi me pencher sur le domaine de l’aide à la décision finale, i.e., comment après avoir exploré un ensemble de solutions, par exemple avec un algorithme multi-objectif, intégrer des critères non formalisés dans la recherche pour décider de la solution retenue (voir par exemple les méthodes de type Topsis [90]).

CONCLUSION

Ce manuscrit couvre une douzaine d'années de travaux dans le domaine de la Recherche Opérationnelle appliquée, principalement dans le domaine des systèmes embarqués. Après la décennie des années 2000, où les méthodes d'optimisation multi-objectif se sont multipliées, nous les avons exploitées dans de nombreuses applications à partir de 2013. Les travaux concernent l'ordonnancement temps réel pour 3 doctorants, le stockage dans le Cloud et dans le Fog pour 2 autres étudiants, le routage de drones pour un sixième doctorant. J'ai aussi collaboré avec des collègues, parfois à l'étranger, pour des problèmes de QoS dans les réseaux, de P-median, de réseau biologique, de mémoire Flash, etc.

Mes outils de prédilection sont la programmation linéaire, la théorie des graphes et les méta-heuristiques, notamment MOEA. Comme déjà souligné dans l'introduction, l'utilisation de ces outils et méthodes est rarement directement possible efficacement. Nous avons donc déployé des approches plus ou moins sophistiquées, à la fois pour la modélisation des problèmes posés mais aussi pour la résolution des instances intéressantes.

Les méthodes MOO sont maintenant assez largement connues et répandues. Pour les domaines métier que j'ai pratiqué, le temps reste toujours une métrique intéressante, à considérer en même temps que d'autres. Le problème de l'extensibilité des méthodes et du compromis temps de calcul/précision des solutions perdure également. Il n'est cependant plus maintenant aussi novateur d'introduire ces méthodes MOO dans ces domaines métier. Par contre, comme souligné au chapitre 5, les thématiques de la robustesse des solutions, de la prise en compte de la dynamique de l'environnement et de l'embarquement des processus de calcul eux mêmes, sont les directions de recherche que je veux explorer avec mes collègues, qu'ils soient chercheurs ou doctorants.

BIBLIOGRAPHIE

Publications

Revues internationales avec comité de lecture sélectives (Q1-Q2 du classement SJR)

- [R10] Hand Ouelmokhtar, Yahia Benmoussa, Djamel Benazzouz, Mohamed Abdessamed Ait-Chikh and **Laurent Lemarchand**. Near-Optimal Covering Solution for USV Coastal Monitoring using PAES. *Journal of Intelligent & Robotic Systems*, Q2, Springer, 106 (1), pp 24, 2022.
- [R09] Hand Ouelmokhtar, Yahia Benmoussa, Djamel Benazzouz, Mohamed Abdessamed Ait-Chikh and **Laurent Lemarchand**. Energy-based USV maritime monitoring using multi-objective evolutionary algorithms. *Ocean Engineering*, Q1, Elsevier, 253, pp 111-182, 2022.
- [R08] Mohammed Islam Naas, **Laurent Lemarchand**, Philippe Raipin Parvedy and Jalil Boukhobza. IoT Data Replication and Consistency Management in Fog computing. *ACM Journal of Grid computing*, Q2, 13(3), pp. 33, 2021.
- [R07] Djillali Boukhelef, Jalil Boukhobza, Kamel Boukhalifa, Hamza Ouarnoughi and **Laurent Lemarchand**. Optimizing the cost of DBaaS object placement in hybrid storage systems. *Future Generation Computer Systems*, Q1, Elsevier, 93 :176–187, April 2019.
- [R06] Congping Lin, **Laurent Lemarchand**, Reinhardt Euler and Imogen Sparkes. Modeling the geometry and dynamics of the Endoplasmic Reticulum network. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Q2, 15(2) :377–386, 2018.
- [R05] Rahma Bouaziz, **Laurent Lemarchand**, Frank Singhoff, Béchir Zalila and Mohamed Jmaiel. Multi-Objective Design Exploration Approach for Ravenscar Real-time Systems. *Real-Time Systems*, Q2, Springer Verlag, 54(2) :424–483, 2018.
- [R04] **Laurent Lemarchand** and Jean-Philippe Babau. Bandwidth Allocation for Sharing Video in a Closed Domestic Network. *IEEE Journal of Communications and Networks*, Q2, KICS, 20(2) :168–181, April 2018.

Revues internationales avec comité de lecture non sélectives (Q3-Q4 ou non classées SJR)

- [R03] Amina Chikhaoui, **Laurent Lemarchand**, Jalil Boukhobza and Kamel Boukhalifa. Multi-objective Optimization of Data Placement in a Storage-as-a-Service Federated Cloud. *ACM Transactions on Storage*, Q3, 17(3) :1–32, 2021.

-
- [R02] Jalil Boukhobza, Pierre Olivier, Stéphane Rubini, **Laurent Lemarchand**, Yassine Hadjadj-Aoul and Arezki Laga. MaCACH : An adaptive cache-aware hybrid FTL mapping scheme using feedback control for efficient page-mapped space management. *Journal of Systems Architecture*, Q3, Elsevier, 61(3-4) :157–171, March 2015.
- [R01] **Laurent Lemarchand**, Alain Plantec, Bernard Pottier and Sakina Zanati. An object-oriented environment for specification and concurrent execution of genetic algorithms. *SIG-PLAN OOPS Mess.*, 4(2) :163–165, December 1992.

Chapitre d'ouvrage

- [CH01] Damien Massé, Reinhardt Euler and **Laurent Lemarchand**. On the Connected Spanning Cubic Subgraph Problem. In K.A. Adiprasito, I. Bárány and C. Vilcu, editors, *Proceedings in Mathematics and Statistics*, volume Convexity and Discrete Geometry Including Graph Theory of *Proceedings in Mathematics and Statistics*, pages 109–136. Springer, 2016.

Conférences internationales avec comité de lecture

- [C21] Amina Chikhaoui, **Laurent Lemarchand**, Kamel Boukhalfa and Jalil Boukhobza. StornIR , a multi-objective replica placement strategy for cloud federations. In *SAC '21 : The 36th ACM/SIGAPP Symposium on Applied Computing*, pp 50–59, Virtual Event Republic of Korea, France, March 2021. ACM.
- [C20] Mohamed Handaoui, Jean-Emile Dartois, **Laurent Lemarchand** and Jalil Boukhobza. Salamander : a Holistic Scheduling of MapReduce Jobs on Ephemeral Cloud Resources. In *CCGRID 2020 - 20th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp 1–10, Melbourne, Australia, November 2020. IEEE.
- [C19] Pierre Dissaux, Frank Singhoff, **Laurent Lemarchand**, Hai Nam Tran and Ill-Ham Atchadam. Combined Real-Time, Safety and Security Model Analysis. In *9th European Congress ERTS Embedded Real Time Systems*, pp 10, Toulouse, France, February 2020.
- [C18] Ill-ham Atchadam, **Laurent Lemarchand**, Hai Nam Tran, Frank Singhoff and Karim Bigou. When security affects schedulability of TSP systems : trade-offs observed by design space exploration. In *25th international conference on emerging technologies and factory automaton, ETFA 2020*, pp 369–376, *IEEE Industrial Electronics Society*, Vienna, Austria, September 2020.
- [C17] Goulven Guillou, **Laurent Lemarchand** and Jean-Philippe Babau. Sailboat Routing with Multiple Objectives for Sailing Races. In *23rd Chesapeake Sailing Yacht Symposium*, pp 11, Annapolis, United States, March 2019.
- [C16] Mohammed Islam Naas, Jalil Boukhobza, Philippe Raipin Parvedy and **Laurent Lemarchand**. An Extension to iFogSim to Enable the Design of Data Placement Strategies. In

-
- 2018 *IEEE 2nd International Conference on Fog and Edge Computing (ICFEC)*, pp 1–8, Washington DC, France, May 2018. IEEE.
- [C15] Mohammed Islam Naas, **Laurent Lemarchand**, Jalil Boukhobza and Philippe Raipin. A Graph Partitioning-based Heuristic for Runtime IoT Data Placement Strategies in a Fog infrastructure. In *ACM Symposium on Applied Computing (SAC)*, pp 767–774, Pau, France, April 2018.
- [C14] Mohammed Islam Naas, Philippe Raipin, Jalil Boukhobza and **Laurent Lemarchand**. iFogStor : an IoT Data Placement Strategy for Fog Infrastructure. In *IEEE 1st International Conference on Fog and Edge Computing*, pp 97–104, Madrid, Spain, May 2017.
- [C13] Djillali Boukhelef, Kamel Boukhalifa, Jalil Boukhobza, Hamza Ouarnoughi and **Laurent Lemarchand**. COPS : Cost Based Object Placement Strategies on Hybrid Storage System for DBaaS Cloud. In *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pp 659–664, Madrid, Spain, May 2017.
- [C12] Rahma Bouaziz, **Laurent Lemarchand**, Frank Singhoff, Bechir Zalila and Mohamed Jmaiel. Efficient Parallel Multi-Objective Optimization for Real-time Systems Software Design Exploration. In *International Symposium on Rapid System Prototyping (RSP'16)*, pp 58–64, Pittsburgh, United States, October 2016. IEEE.
- [C11] Rahma Bouaziz, **Laurent Lemarchand**, Frank Singhoff, Bechir Zalila and Mohamed Jmaiel. Architecture Exploration of Real-time Systems Based on Multi-Objective Optimization. In *20th International Conference on Engineering of Complex Computer Systems (ICECCS 2015)*, pp 10, Golden Coast, Australia, December 2015. IEEE.
- [C10] Pascal Rebreyend, **Laurent Lemarchand** and Reinhardt Euler. A Computational Comparison of Different Algorithms for Very Large p -median Problems. In *EvoCOP 2015 - The 15th European Conference on Evolutionary Computation in Combinatorial Optimisation*, volume 9026 of *Evolutionary Computation in Combinatorial Optimization*, pp 13–24, Copenhagen, Denmark, April 2015. Springer.
- [C09] **Laurent Lemarchand**, Isaac Armah-Mensah and Jean-Philippe Babau. Dynamic Server Configuration for Multiple Streaming in a Home Network. In *International Conference on Embedded and Ubiquitous Computing*, pages pp 39–45, Milan, Italy, August 2014. IEEE.
- [C08] **Laurent Lemarchand**, Reinhardt Euler, Congping Lin and Imogen Sparkes. Modeling the Geometry of the Endoplasmic Reticulum Network. In C. Martin-Vide A. Horia-Dediu and B. Truthe, editors, *1st International Conference on Algorithms for Computational Biology – AlCob'14*, volume LNBI 8542, pages 132–146, Tarragona, Spain, July 2014. Springer.
- [C07] Reinhardt Euler, **Laurent Lemarchand**, Congping Lin and Imogen Sparkes. Modeling the geometry of the Endoplasmic Reticulum network. In *20th Conference of the Interna-*

tional Federation of Operational Research Societies IFORS 2014, Barcelone, Spain, July 2014.

- [C06] Asma Mehiaoui, Ernest Wozniak, Sara Tucci Piergiovanni, Chokri Mraidha, Marco Di Natale, Haibo Zeng, Jean-Philippe Babau, **Laurent Lemarchand** and Sébastien Gerard. A two-step optimization technique for functions placement, partitioning, and priority assignment in distributed systems. In Bjorn Franke and Jingling Xue, editors, *SIGPLAN/SIGBED Conference on Languages, Compilers and Tools for Embedded Systems 2013*, pp 121–132, Seattle, United States, June 2013. ACM.
- [C05] **Laurent Lemarchand**, Maxime Louvel and Jean-Philippe Babau. VBR Video Abstraction for Home-Network Reservation. In *7th International Conference on Embedded and Multimedia Computing (EMC-12)*, volume 181 of *Lecture Notes in Electrical Engineering*, pp 113–122, Gwangju, South Korea, September 2012. Springer.
- [C04] Asma Mehiaoui, Sara Tucci Piergiovanni, Jean-Philippe Babau and **Laurent Lemarchand**. Optimizing the Deployment of Distributed Real-Time Embedded Applications. In *IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pp 400–403, Séoul, South Korea, August 2012.
- [C03] Reinhardt Euler and **Laurent Lemarchand**. Some Applications of Combinatorial Optimization in Parallel Computing. In *8th Franco-Japanese and 4th Franco-Chinese Conference*, volume 1120 of *Combinatorics and Computer Science CCS'95*, pp 348–366, Brest, France, July 1995. Springer.
- [C02] **Laurent Lemarchand**. Parallel synthesis of large combinational circuits for FPGAs. In Bob Hertzberger and Peter Sloot, editors, *High-Performance Computing and Networking*, pages 1042–1043, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.
- [C01] **Laurent Lemarchand**. Parallel performance directed technology mapping for FPGA. In *1999 Southwest Symposium on Mixed-Signal Design*, pages 189–194, 1999.

Ateliers internationaux avec comité de lecture et actes

- [W08] Kilian Le Gall, **Laurent Lemarchand** and Catherine Dezan. Multi-objective optimization for an online re-planning of autonomous vehicles. In *SSIV'23, 9th Workshop on Safety and Security of Intelligent Vehicles*, Porto, 2023.
- [W07] Ill-Ham Atchadam, **Laurent Lemarchand**, Frank Singhoff and Hai Nam Tran. Observing the impact of multicore execution platform for TSP systems under schedulability, security and safety constraints. In *17th int. workshop on Security and Dependability of Critical Embedded Real-Time Systems (CERTS)*, Stuttgart, Germany, sept. 2022.

-
- [W06] Ill-Ham Atchadam, Frank Singhoff, Hai Nam Tran, nora bouzid and **Laurent Lemarchand**. Combined security and schedulability analysis for MILS real-time critical architectures. In *4th international workshop on Security and Dependability of Critical Embedded Real-Time Systems (CERTS)*, Stuttgart, Germany, June 2019.
- [W05] Pascal Rebreyend, **Laurent Lemarchand** and Damien Massé. Multi-Objective optimization : Comparison of methods for the p-median problem. In *20th annual workshop on metaheuristics*, Oslo, Norway, May 2019.
- [W04] Bertrand Gilles, Valérie-Anne Nicolas and **Laurent Lemarchand**. Timed Functional Modeling for Mixed-Signal Boards in Maintenance Testing. In *7th IEEE Latin-American Test Workshop, LATW 2006*, page p.19, Buenos Aires, Argentina, March 2006.
- [W03] Bertrand Gilles, Valérie-Anne Nicolas, **Laurent Lemarchand**, Lionel Marcé and Bruno Castel. Towards a New Modelling of Mixed-Signal Boards For Maintenance Testing. In *11th IEEE International Mixed-Signal Testing Workshop, IMSTW'05*, page p.90, Cannes, France, June 2005.
- [W02] **Laurent Lemarchand**, Bertrand Gilles, Valérie-Anne Nicolas, Lionel Marce, and Bruno Castel. Maintenance Testing of Mixed-Signal Boards. In *2nd IEEE Electronic System Test Workshop, ESTW05*, Austin, United States, November 2005.
- [W01] Valérie-Anne Nicolas, Bertrand Gilles, **Laurent Lemarchand** and Bruno Castel. A Maintenance-Oriented Board testing Approach. In *3rd IEEE International East-West Design and Test Workshop, EWDTW'05*, page p.143, Odessa, Ukraine, September 2005.

Thèse

- [T01] **Laurent Lemarchand**. *Parallelisation par partitionnement pour la synthèse logique combinatoire sur FPGA à base de LUT*. PhD thesis, Université de Rennes I, janv. 1999.

Revue nationale avec comité de lecture

- [RN01] **Laurent Lemarchand**. Conversion technologique en parallèle pour l'optimisation des délais sur FPGA à base de LUT. *Technique et Science Informatique*, 19(6) :813–834, 2000.

Conférences nationales

- [CN03] Amina Chikhaoui, **Laurent Lemarchand**, Kamel Boukhalfa and Jalil Boukhobza. StorNIR , a multi-objective replica placement strategy for cloud federations. In *Cloud 2021 : Journées GDR RSD*.
- [CN02] Pascal Rebreyend and **Laurent Lemarchand**. Testing Algorithm For Large P-median Problems In Heterogenous Road Networks. In *INFORMS*, Philadelphia, United States, November 2015.

-
- [CN01] J. Craveiro, J.L. Souza, J. Rufino, Vincent Gaudel, **Laurent Lemarchand**, Alain Plantec, Stéphane Rubini and Frank Singhoff. Scheduling analysis principles and tool for time and space partitioned systems. In *Simposio de informatica*, pages 582–585, Lisbonne, Portugal, September 2012.

Revue Open Access avec comité de lecture

- [OA01] **Laurent Lemarchand**, Damien Massé, Pascal Rebreyend and Johan Håkansson. Multiobjective Optimization for Multimode Transportation Problems. *Advances in Operations Research*, 2018 :1 – 13, June 2018.

Séminaires invités

- [I04] **Laurent Lemarchand**. Design Space Exploration for TSP systems on multicore platforms under schedulability, security and safety constraints. GDR RO & SOC2, journées du groupe de travail commun OSI, Paris, France, déc 2022
- [I03] **Laurent Lemarchand**. Optimisation multi-objectifs et applications de la théorie des graphes, séminaire invité, Ecole d’Intelligence Artificielle, Alger, Algérie, avr. 2022.
- [I02] **Laurent Lemarchand**. Routage de voiliers et autres applications MOO, Ecole Supérieure d’Informatique, Alger, Algérie, avr. 2018.
- [I01] **Laurent Lemarchand**. Applications of multi-objective optimization. CEMPS, University of Exeter, GB, oct. 2016.

Posters

- [P03] M. Vuksic, T. Kovacevic, B. Dzaja, P. Dukic, S. Sitic, H. N. Tran, V. Rodin, **L. Lemarchand**, V.-A. Nicolas, A. Plantec, F. Singhoff. Increasing the autonomy of an underwater ROV. SeaTech Week, Brest, France, sept 2022.
- [P02] Illham Atchadam, Frank Singhoff, Hai Nam Tran, **Laurent Lemarchand**. A design space exploration approach to jointly optimize security and schedulability in TSP systems. Colloque du GDR SOC2, Strasbourg, France, Jun 2022.
- [P01] Evan Flécheau, **Laurent Lemarchand**, Catherine Dezan. Multi-objective optimization at the EDge for Online and Real-time elf-Adaptation of Autonomous vehicles. Colloque du GDR SOC2, Strasbourg, France, Jun 2022 .

Rapports techniques

-
- [RT05] Bertrand Gilles, Valérie-Anne Nicolas, **Laurent Lemarchand** and Lionel Marcé. Timed Functional Modeling for Mixed-Signal Boards in Maintenance Testing : A Case Study. Technical report, February 2006.
- [RT04] Valérie-Anne Nicolas, Bertrand Gilles, **Laurent Lemarchand**, Lionel Marcé and Bruno Castel. Maintenance Testing of Mixed-Signal Boards : the FCB case study. Technical report, June 2005.
- [RT03] Valérie-Anne Nicolas, Bertrand Gilles, **Laurent Lemarchand**, Lionel Marcé and Bruno Castel. Plate-forme de test pour les cartes hybrides, November 2005. Rapport final du projet collaboration ISIS-MPP, UBO-LISYC, Région Bretagne.
- [RT02] **Laurent Lemarchand**. Implantation de PPart sous SmallTalk : synthèse de circuits hiérarchiques. Technical report, 2002.
- [RT01] **Laurent Lemarchand**. Une interface vers Lex et Yacc sous SmallTalk. Technical report, 2001.

Références bibliographiques

- [1] Philpott A. and Mason A. Optimising yacht routes under uncertainty. In *15th Chesapeake Yacht Symposium*, 2015.
- [2] Mohammad Aazam and Eui-Nam Huh. E-hamc : Leveraging fog computing for emergency alert service. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 518–523. IEEE, 2015.
- [3] AEEC. Avionics application software standard interface (ARINC) 653 : Avionics application software standard. Technical report, Airlines Electronic Engineering Committee, 1997.
- [4] Aldeida Aleti, Barbora Buhnova, Lars Grunske, Anne Koziol, and Indika Meedeniya. Software architecture optimization methods : A systematic literature review. *IEEE Transactions on Software Engineering*, 39(5) :658–683, 2012.
- [5] Björn Annighöfer and Frank Thielecke. A systems architecting framework for optimal distributed integrated modular avionics architectures. *CEAS Aeronautical Journal*, 6 :485–496, 2015.
- [6] José EC Arroyo, Paula M Dos Santos, Michele S Soares, and André G Santos. A multi-objective genetic algorithm with path relinking for the p-median problem. In *Advances in Artificial Intelligence—IBERAMIA 2010 : 12th Ibero-American Conference on AI, Bahía Blanca, Argentina, November 1-5, 2010. Proceedings 12*, pages 70–79. Springer, 2010.
- [7] Ill-ham Atchadam. *Exploration d’architectures logicielles pour les systèmes critiques partitionnés*. PhD thesis, UBO, Brest, 2023.

-
- [8] Charles Audet, Jean Bignon, Dominique Cartier, Sébastien Le Digabel, and Ludovic Salomon. Performance indicators in multiobjective optimization. *European journal of operational research*, 292(2) :397–422, 2021.
- [9] Roy B. Multicriteria methodology for decision aiding. *Nonconvex Optimization and Its Applications Series*, 12, 1996.
- [10] Francisco Barahona. On the k-cut problem. *Operations Research Letters*, 26(3) :99–105, 2000.
- [11] Vincent Baudoui. *Optimisation robuste multiobjectifs par modèles de substitution*. PhD thesis, Toulouse, ISAE, 2012.
- [12] Olivier Beaumont, Louis-Claude Canon, Lionel Eyraud-Dubois, Giorgio Lucarelli, Loris Marchal, Clément Mommessin, Bertrand Simon, and Denis Trystram. Scheduling on two types of resources : a survey. *ACM Computing Surveys (CSUR)*, 53(3) :1–36, 2020.
- [13] D Elliott Bell and Leonard J La Padula. Secure computer system : Unified exposition and multics interpretation. Technical report, MITRE CORP BEDFORD MA, 1976.
- [14] Richard Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1) :87–90, 1958.
- [15] Mohamed S. Ben Ameer and Anis Sakly. FPGA based hardware implementation of bat algorithm. *Applied Soft Computing*, 58 :378–387, 2017.
- [16] Cello Benchmark. <http://tesla.hpl.hp.com/opensource/cello99>. web page visited 02.02.2014.
- [17] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization : a methodological tour d’horizon. *European Journal of Operational Research*, 2020.
- [18] Jean-François Bérubé. Problèmes de tournées multicritères dans des graphes. 2008.
- [19] Kenneth J Biba. Integrity considerations for secure computer systems. Technical report, MITRE CORP BEDFORD MA, 1977.
- [20] Julian Blank and Kalyanmoy Deb. Pymoo : Multi-objective optimization in python. *IEEE Access*, 8 :89497–89509, 2020.
- [21] Stefan Bleuler, Marco Laumanns, Lothar Thiele, and Eckart Zitzler. Pisa—a platform and programming language independent interface for search algorithms. In *Evolutionary Multi-Criterion Optimization : Second International Conference, EMO 2003, Faro, Portugal, April 8–11, 2003. Proceedings 2*, pages 494–508. Springer, 2003.
- [22] Rahma Bouaziz Frikha. *Multi-Objective Architecture Exploration and Optimization of Critical Real-Time Systems*. PhD thesis, Sfax university, 2018. Thèse de doctorat dirigée par Jmail, Mohamed, RedCAD Lab, Sfax, Tunisie.

-
- [23] Djillali Boukhelef. *Vers une méthodologie d'optimisation du placement des objets DBAAS dans un environnement de Cloud computing*. Theses, USTHB, September 2019.
- [24] Jalil Boukhobza, Pierre Olivier, and Stéphane Rubini. A scalable and highly configurable cache-aware hybrid flash translation layer. *Computers*, 3(1) :36–57, 2014.
- [25] Jalil Boukhobza, Stéphane Rubini, Renhai Chen, and Zili Shao. Emerging nvm : A survey on architectural integration and research challenges. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 23(2) :1–32, 2017.
- [26] Alex Boydston, Peter Feiler, Steve Vestal, and Bruce Lewis. Architecture centric virtual integration process (acvip) : A key component of the dod digital engineering strategy. Technical report, Carnegie-Mellon University (, 2019.
- [27] Alan Burns. The ravenscar profile. *ACM SIGAda Ada Letters*, 19(4) :49–52, 1999.
- [28] Jean-François Bérubé, Michel Gendreau, and Jean-Yves Potvin. An exact ϵ -constraint method for bi-objective combinatorial optimization problems : Application to the Traveling Salesman Problem with Profits. *European Journal of Operational Research*, 194(1) :39–50, April 2009.
- [29] LG Chalmet, L Lemonidis, and DJ Elzinga. An algorithm for the bi-criterion integer programming problem. *European Journal of Operational Research*, 25(2) :292–300, 1986.
- [30] Estelle Chauveau. *Optimisation des routes maritimes : un système de résolution multicritère et dépendant du temps*. PhD thesis, Aix-Marseille, 2018.
- [31] Amina Chikhaoui. *Vers une approche intelligente de placement de données dans un cloud distribué basé sur un système de stockage hybride*. PhD thesis, UBO, Brest, 2022.
- [32] Amina Chikhaoui, Kamel Boukhalfa, and Jalil Boukhobza. A cost model for hybrid storage systems in a cloud federations. In *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 1025–1034, 2018.
- [33] Amina Chikhaoui, Kamel Boukhalfa, and Jalil Boukhobza. A cost model for hybrid storage systems in a cloud federations. In *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 1025–1034, 2018.
- [34] P.C. Chu and J.E. Beasley. A genetic algorithm for the generalised assignment problem. *Computers and Operations Research*, 24(1) :17–23, 1997.
- [35] Carlos A Coello Coello, Gary B Lamont, David A Van Veldhuizen, et al. *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer, 2007.
- [36] Thomas H Cormen, Charles Eric Leiserson, Ronald L Rivest, and Clifford Stein. *Algorithmique : cours avec 957 exercices et 158 problèmes*. Dunod, 2010.
- [37] Schaffer J. D. Multiple objective optimization with vector evaluated genetic algorithms. In *Proc. Int'l Conf. Genetic Algorithms and Their Applications, 1985*, pages 93–100, 1985.

-
- [38] Brian C Dean. Shortest paths in fifo time-dependent networks : Theory and algorithms. *Rapport technique, Massachusetts Institute of Technology*, 13, 2004.
- [39] Kalyanmoy Deb, Ram Bhushan Agrawal, et al. Simulated binary crossover for continuous search space. *Complex systems*, 9(2) :115–148, 1995.
- [40] Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i : Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4) :577–601, 2014.
- [41] Kalyanmoy Deb and Deb Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., 2001.
- [42] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm : Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2) :182–197, 2002.
- [43] Clarisse Dhaenens and Laetitia Jourdan. Metaheuristics for data mining : survey and opportunities for big data. *Annals of Operations Research*, 314(1) :117–140, 2022.
- [44] Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1) :269–271, 1959.
- [45] Johann Dréo, Alain Pétrowski, Patrick Siarry, and Eric Taillard. *Métaheuristiques pour l’optimisation difficile*. Eyrolles, 2003.
- [46] Juan J. Durillo and Antonio J. Nebro. jmetal : A java framework for multi-objective optimization. *Advances in Engineering Software*, 42 :760–771, 2011.
- [47] Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of research of the National Bureau of Standards B*, 69(125-130) :55–56, 1965.
- [48] Michael Emmerich, Nicola Beume, and Boris Naujoks. An emo algorithm using the hypervolume measure as selection criterion. In *Evolutionary Multi-Criterion Optimization : Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005. Proceedings 3*, pages 62–76. Springer, 2005.
- [49] John William Evans and Clarence Filsfils. *Deploying IP and MPLS QoS for multiservice networks : theory and practice*. Elsevier, 2010.
- [50] EVO. [http ://www.evostar.org/2022/](http://www.evostar.org/2022/). web page visited 07.29.2021.
- [51] Jesús Guillermo Falcón-Cardona, Raquel Hernández Gómez, Carlos A Coello Coello, and Ma Guadalupe Castillo Tapia. Parallel multi-objective evolutionary algorithms : A comprehensive survey. *Swarm and Evolutionary Computation*, 67 :100960, 2021.
- [52] Peter H Feiler, Bruce Lewis, Steve Vestal, and Ed Colbert. An overview of the sae architecture analysis & design language (aadl) standard : A basis for model-based architecture-driven embedded systems engineering. In *Architecture Description Languages : IFIP TC-2*

-
- Workshop on Architecture Description Languages (WADL), World Computer Congress, Aug. 22–27, 2004, Toulouse, France*, pages 3–15. Springer, 2005.
- [53] Dominique Feillet, Pierre Dejax, and Michel Gendreau. Traveling salesman problems with profits. *Transportation science*, 39(2) :188–205, 2005.
- [54] Jonathan E Fieldsend and Richard M Everson. The rolling tide evolutionary algorithm : A multiobjective optimizer for noisy optimization problems. *IEEE Transactions on Evolutionary Computation*, 19(1) :103–117, 2014.
- [55] Carlo Filippi and Elisa Stevanato. A two-phase method for bi-objective combinatorial optimization and its application to the tsp with profits. *Algorithmic Operations Research*, 7(2) :125–139, 2012.
- [56] Carlo Filippi and Elisa Stevanato. Approximation schemes for bi-objective combinatorial optimization and their application to the tsp with profits. *Computers & Operations Research*, 40(10) :2418–2428, 2013.
- [57] Robert W Floyd. Algorithm 97 : shortest path. *Communications of the ACM*, 5(6) :345, 1962.
- [58] Alexander I.J. Forrester and Andy J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1) :50–79, 2009.
- [59] MOEA Framework. <http://moeaframework.org>. web page visited 02.04.2023.
- [60] Xavier Gandibleux, Frédéric Beugnies, and Sabine Randriamasy. Martins’ algorithm revisited for multi-objective shortest path problems with a maxmin cost function. *4OR*, 4(1) :47–59, 2006.
- [61] Xavier Gandibleux, Frédéric Beugnies, and Sabine Randriamasy. Martins’ algorithm revisited for multi-objective shortest path problems with a maxmin cost function. *4OR*, 4(1) :47–59, 2006.
- [62] Xavier Gandibleux, Gauthier Soleilhac, and Anthony Przybylski. voptsolver, an ecosystem for multi-objective linear optimization. In *JuliaCon conference*, 2021.
- [63] Bertrand Gilles. *Méthodologie de test pour cartes mixtes analogiques-numériques*. PhD thesis, UBO, 2009. Thèse de doctorat dirigée par Nana Tchamnda, Laurent Informatique Brest 2009.
- [64] Christos Gkerekos and Iraklis Lazakis. A novel, data-driven heuristic framework for vessel weather routing. *Ocean Engineering*, 197 :106887, 2020.
- [65] D.E. Goldberg and V. Corruble. *Algorithmes génétiques : exploration, optimisation et apprentissage automatique*. Vie artificielle. Ed. Addison-Wesley France, 1994.
- [66] Joël Goossens, Emmanuel Grolleau, and Liliana Cucu-Grosjean. Periodicity of real-time schedules for dependent periodic tasks on identical multiprocessor platforms. *Real-time systems*, 52 :808–832, 2016.

-
- [67] Uma Goyal and Craig Blackstone. Untangling the web : mechanisms underlying er network formation. *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research*, 1833(11) :2492–2498, 2013.
- [68] Goulven Guillou. *Architecture multi-agents pour le pilotage automatique des voiliers de compétition et extensions algébriques des réseaux de Petri*. PhD thesis, Université de Bretagne Occidentale, 2010.
- [69] Lizheng Guo, Zongyao He, Shuguang Zhao, Na Zhang, Junhao Wang, and Changyun Jiang. Multi-objective optimization for data placement strategy in cloud computing. In *Information Computing and Applications : Third International Conference, ICICA 2012, Chengde, China, September 14-16, 2012. Proceedings, Part II 3*, pages 119–126. Springer, 2012.
- [70] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K Ghosh, and Rajkumar Buyya. ifogsim : A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments. *Software : Practice and Experience*, 47(9) :1275–1296, 2017.
- [71] Dan Gusfield. Very simple methods for all pairs network flow analysis. *SIAM Journal on Computing*, 19(1) :143–155, 1990.
- [72] Hideki Hagiwara. Weather routing of (sail-assisted) motor vessels. *PhD Thesis, Technical University of Delft*, 1989.
- [73] Pierre Hansen. Bicriterion path problems. In *Multiple Criteria Decision Making Theory and Application : Proceedings of the Third Conference Hagen/Königswinter, West Germany, August 20–24, 1979*, pages 109–127. Springer, 1980.
- [74] Monowar Hasan, Sibin Mohan, Rodolfo Pellizzoni, and Rakesh B Bobba. A design-space exploration for allocating security tasks in multicore real-time systems. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 225–230. IEEE, 2018.
- [75] Michael Held and Richard M Karp. The traveling-salesman problem and minimum spanning trees. *Operations Research*, 18(6) :1138–1162, 1970.
- [76] J. Hinnenthal and G. Clauss. Robust pareto-optimum routing of ships utilising deterministic and ensemble weather forecasts. *Ships and Offshore Structures*, 5(2) :105–114, 2010.
- [77] Ching-Lai Hwang, Kwangsun Yoon, Ching-Lai Hwang, and Kwangsun Yoon. Methods for multiple attribute decision making. *Multiple attribute decision making : methods and applications a state-of-the-art survey*, pages 58–191, 1981.
- [78] Teerawat Issariyakul, Ekram Hossain, Teerawat Issariyakul, and Ekram Hossain. *Introduction to network simulator 2 (NS2)*. Springer, 2009.

-
- [79] Hinnenthal J. and Clauss G. Robust pareto-optimum routing of ships utilising deterministic and ensemble weather forecasts. *Ships and Offshore Structures*, 5(2) :105–114, 2010.
- [80] Nicolas Jozefowicz. *Optimisation combinatoire multi-objectif : des méthodes aux problèmes, de la Terre à (presque) la Lune*. PhD thesis, Institut National Polytechnique de Toulouse (INP Toulouse), 2013.
- [81] Elena Kakoulli and Herodotos Herodotou. Octopusfs : A distributed file system with tiered storage management. In *Proceedings of the 2017 acm international conference on management of data*, pages 65–78, 2017.
- [82] Sungjoo Kang, Seongjin Lee, Youjip Won, and Byeongchan Seong. On-line prediction of nonstationary variable-bit-rate video traffic. *IEEE Transactions on Signal Processing*, 58(3) :1219–1237, 2009.
- [83] Maryam Karimi-Mamaghan, Mehrdad Mohammadi, Patrick Meyer, Amir Mohammad Karimi-Mamaghan, and El-Ghazali Talbi. Machine learning at the service of metaheuristics for solving combinatorial optimization problems : A state-of-the-art. *European Journal of Operational Research*, 296(2) :393–422, 2022.
- [84] Maryam Karimi-Mamaghan, Mehrdad Mohammadi, Bastien Pasdeloup, Romain Billot, and Patrick Meyer. An online learning-based metaheuristic for solving combinatorial optimization problems. In *ROADEF 2020 : 21ème Congrès Annuel de la Société Française de Recherche Opérationnelle et d’Aide à la Décision*, 2020.
- [85] Maryam Karimi-Mamaghan, Mehrdad Mohammadi, Bastien Pasdeloup, and Patrick Meyer. Learning to select operators in meta-heuristics : An integration of q-learning into the iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 304(3) :1296–1330, 2023.
- [86] George Karypis and Vipin Kumar. Multilevelk-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed computing*, 48(1) :96–129, 1998.
- [87] J. D. Knowles and D. W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation*, 8(2) :149–172, 2000.
- [88] Joshua Knowles and Evan J Hughes. Multiobjective optimization on a budget of 250 evaluations. In *Evolutionary Multi-Criterion Optimization : Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005. Proceedings 3*, pages 176–190. Springer, 2005.
- [89] Anne Kozirolek, Heiko Kozirolek, and Ralf Reussner. Peropteryx : automated application of tactics in multi-objective software architecture optimization. In *Proceedings of the joint ACM SIGSOFT conference–QoSA and ACM SIGSOFT symposium–ISARCS on Quality*

-
- of software architectures—QoSA and architecting critical systems—ISARCS*, pages 33–42, 2011.
- [90] Young-Jou Lai, Ting-Yun Liu, and Ching-Lai Hwang. Topsis for modm. *European journal of operational research*, 76(3) :486–500, 1994.
- [91] Soniya Lalwani, Sorabh Singhal, Rajesh Kumar, and Nilama Gupta. A comprehensive survey : Applications of multi-objective particle swarm optimization (mopso) algorithm. *Transactions on combinatorics*, 2(1) :39–101, 2013.
- [92] K Lamamra, K Belarbi, A Belhani, and S Boukhtini. Nsga2 based of multi-criteria decision analysis for multi-objective optimization of fuzzy logic controller for non linear system. *Journal of Next Generation Information Technology*, 5(1) :57, 2014.
- [93] Emilien Lavigne. *AMSA, un framework dédié à la simulation des lois de contrôle pour des voiliers de compétition*. PhD thesis, Université de Bretagne Occidentale, 2019.
- [94] Vuk Lesi, Ilija Jovanov, and Miroslav Pajic. Security-aware scheduling of embedded control tasks. *ACM Transactions on Embedded Computing Systems (TECS)*, 16(5s) :1–21, 2017.
- [95] Chunlin Li, YaPing Wang, Hengliang Tang, and Youlong Luo. Dynamic multi-objective optimized replica placement and migration strategies for saas applications in edge cloud. *Future Generation Computer Systems*, 100 :921–937, 2019.
- [96] Rui Li, Ramin Etemaadi, Michael TM Emmerich, and Michel RV Chaudron. An evolutionary multiobjective optimization approach to component-based software architecture design. In *2011 IEEE congress of evolutionary computation (CEC)*, pages 432–439. IEEE, 2011.
- [97] Jorg Liebeherr. An efficient solution to traffic characterization of vbr video in quality-of-service networks. *Technical Report*, 1996, 1996.
- [98] Arnaud Liefvooghe, Laetitia Jourdan, and El-Ghazali Talbi. A software framework based on a conceptual unified model for evolutionary multiobjective optimization : Paradiseo-moeo. *European Journal of Operational Research*, 209(2) :104–112, 2011.
- [99] Congping Lin, Yiwei Zhang, Imogen Sparkes, and Peter Ashwin. Structure and dynamics of er : minimal networks and biophysical constraints. *Biophysical Journal*, 107(3) :763–772, 2014.
- [100] LION 17. <http://lion17.org/>. web page visited 15.09.2023.
- [101] Chung Laung Liu and James W Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM)*, 20(1) :46–61, 1973.
- [102] Xiyang Liu, Lei Fan, Liming Wang, and Sha Meng. PSO based multiobjective reliable optimization model for cloud storage. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable,*

Autonomic and Secure Computing; Pervasive Intelligence and Computing, pages 2263–2269. IEEE, 2015.

- [103] Antonio López Jaimes, Carlos A Coello Coello, and Jesús E Urías Barrientos. Online objective reduction to deal with many-objective problems. In *Evolutionary Multi-Criterion Optimization : 5th International Conference, EMO 2009, Nantes, France, April 7-10, 2009. Proceedings 5*, pages 423–437. Springer, 2009.
- [104] Flavien Lucas. *Résolution de problèmes réalistes de tournées à flotte hétérogène en milieu urbain : vers un solveur adaptatif mêlant recherche opérationnelle et apprentissage automatique*. PhD thesis, Université de Bretagne Sud, 2020.
- [105] Yong Ma, Mengqi Hu, and Xinping Yan. Multi-objective path planning for unmanned surface vehicle with currents effects. *ISA transactions*, 75 :137–156, 2018.
- [106] Yaser Mansouri, Adel Nadjaran Toosi, and Rajkumar Buyya. Data storage management in cloud environments : Taxonomy, survey, and future directions. *ACM Comput. Surv.*, 50(6), December 2017.
- [107] Guoqiang Mao and Huabing Liu. Real time variable bit rate video traffic prediction. *International Journal of Communication Systems*, 20(4) :491–505, 2007.
- [108] Ernesto de Queros Vieira Martins and JLE Santos. The labelling algorithm for the multiobjective shortest path problem. *Departamento de Matematica, Universidade de Coimbra, Portugal, Tech. Rep. TR-99/005*, 1999.
- [109] Florin Maticu, Paul Pop, Christian Axbrink, and Mafijul Islam. Automatic functionality assignment to autosar multicore distributed architectures. *SAE Technical Paper, Tech. Rep*, 2016.
- [110] Asma Mehiaoui. *Techniques d’analyse et d’optimisation pour la synthèse architecturale de systèmes temps réel embarqués distribués : problèmes de placement, de partitionnement et d’ordonnancement*. Theses, Université de Bretagne occidentale - Brest, June 2014.
- [111] Clair E Miller, Albert W Tucker, and Richard A Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4) :326–329, 1960.
- [112] Sparsh Mittal, A Pande, L Wang, and Praveen Kumar. Design exploration and implementation of simplex algorithm over reconfigurable computing platforms. In *IEEE International Conference on Digital Convergence*, pages 204–209, 2011.
- [113] Jacqueline Moore. Application of particle swarm to multiobjective optimization. *Technical report*, 1999.
- [114] Mohammed Islam Naas. *Placement des données de l’Internet des Objets dans une infrastructure de Fog*. PhD thesis, UBO, 2019. Thèse de doctorat dirigée par Boukhobza, Jalil Informatique Brest 2019.

-
- [115] Hanlin Niu, Yu Lu, Al Savvaris, and Antonios Tsourdos. An energy-efficient path planning algorithm for unmanned surface vehicles. *Ocean Engineering*, 161 :308–321, 2018.
- [116] Hand Oueldmoktar. *Energy-aware USVs path planning*. PhD thesis, Université Bourmédès, 2022.
- [117] M. Ozlen, B. A. Burton, and C. A. G. MacRae. Multi-objective integer programming : An improved recursive algorithm. *ArXiv e-prints*, April 2011.
- [118] Melih Ozlen, Benjamin A Burton, and Cameron AG MacRae. Multi-objective integer programming : An improved recursive algorithm. *Journal of Optimization Theory and Applications*, 160(2) :470–482, 2014.
- [119] Gintaras Palubeckis, Eimutis Karčiauskas, and Aleksas Riškus. Comparative performance of three metaheuristic approaches for the maximally diverse grouping problem. *Information Technology and Control*, 40(4) :277–285, 2011.
- [120] Jihoon Park, Sukkeun Kim, Geemoon Noh, Hyeongmin Kim, Daewoo Lee, and Inwon Lee. Mission planning and performance verification of an unmanned surface vehicle using a genetic algorithm. *International Journal of Naval Architecture and Ocean Engineering*, 13 :575–584, 2021.
- [121] Robert Clay Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6) :1389–1401, 1957.
- [122] Smail Rahmoun, Etienne Borde, and Laurent Pautet. Automatic selection and composition of model transformations alternatives using evolutionary algorithms. In *Proceedings of the 2015 European Conference on Software Architecture Workshops*, pages 1–7, 2015.
- [123] RV Rao and RJ Lakshmi. Ranking of pareto-optimal solutions and selecting the best solution in multi-and many-objective optimization problems using r-method. *Soft Computing Letters*, 3 :100015, 2021.
- [124] Diego Recalde, Daniel Severín, Ramiro Torres, and Polo Vaca. An exact approach for the balanced k-way partitioning problem with weight constraints and its application to sports team realignment. *Journal of Combinatorial Optimization*, 36(3) :916–936, 2018.
- [125] J. Reese. Solution methods for the p-median problem : An annotated bibliography. *Networks*, 48(3) :125–142, 2006.
- [126] ROADEF. *Le livre blanc de la Recherche Opérationnelle en France*. <https://www.roadef.org/roadef-livre-blanc>, 2019.
- [127] Gian-Carlo Rota. The number of partitions of a set. *The American Mathematical Monthly*, 71(5) :498–504, 1964.
- [128] Salah Eddine Saidi, Sylvain Cotard, Khaled Chaaban, and Kevin Marteil. An ilp approach for mapping autosar runnables on multi-core architectures. In *Proceedings of the 2015*

-
- Workshop on Rapid Simulation and Performance Evaluation : Methods and Tools*, pages 1–8, 2015.
- [129] M Sakarovitch. Optimisation combinatoire, tome 1 : Graphes et programmation linéaire, et tome 2 : Programmation discrète. 1984.
- [130] Fatemeh Sayyady, Golbarg K Tutunchi, and Yahya Fathi. P-median and p-dispersion problems : A bi-criteria analysis. *Computers & Operations Research*, 61 :46–55, 2015.
- [131] Patrick Siarry. *Métaheuristiques : Recuits simulé, recherche avec tabous, recherche à voisinages variables, méthodes GRASP, algorithmes évolutionnaires, fourmis artificielles, essais particuliers et autres méthodes d'optimisation*. Editions Eyrolles, 2014.
- [132] Frank Singhoff, Jérôme Legrand, Laurent Nana, and Lionel Marcé. Cheddar : a flexible real time scheduling framework. In *Proceedings of the 2004 annual ACM SIGAda international conference on Ada : The engineering of correct and reliable software for real-time & distributed systems using Ada and related technologies*, pages 1–8, 2004.
- [133] Sang Hyuk Son, Ravi Mukkamala, and Rasikan David. Integrating security and real-time requirements using covert channel capacity. *IEEE Transactions on Knowledge and data Engineering*, 12(6) :865–879, 2000.
- [134] Kenneth Sorensen, Marc Sevaux, and Fred Glover. A history of metaheuristics. *arXiv preprint arXiv :1704.00853*, 2017.
- [135] Warm start method. <https://www.ibm.com/docs/en/icos/20.1.0?topic=mip-starting-from-solution-starts>. *IBM CPLEX user manual*, web page visited 07.29.2021.
- [136] IBM ILOG CPLEX Optimization Studio. <https://www.ibm.com/docs/en/icos/12.8.0.0?topic=cplex-users-manual>. *IBM CPLEX user manual*, web page visited 07.10.2023.
- [137] A Suppakitnarm, Keith A Seffen, Geoff T Parks, and PJ Clarkson. A simulated annealing algorithm for multiobjective optimization. *Engineering optimization*, 33(1) :59–85, 2000.
- [138] Ken Tindell and John Clark. Holistic schedulability analysis for distributed hard real-time systems. *Microprocessing and microprogramming*, 40(2-3) :117–134, 1994.
- [139] Ekunda Lukata Ulungu and Jacques Teghem. The two phases method : An efficient procedure to solve bi-objective combinatorial optimization problems. *Foundations of computing and decision sciences*, 20(2) :149–165, 1995.
- [140] Ekunda Lukata Ulungu and Jacques Teghem. The two phases method : An efficient procedure to solve bi-objective combinatorial optimization problems. *Foundations of Computing and Decision Sciences*, 20(2) :149–165, 1995.
- [141] Michalis Vavouras, Kyprianos Papadimitriou, and Ioannis Papaefstathiou. High-speed fpga-based implementations of a genetic algorithm. In *2009 International Symposium on Systems, Architectures, Modeling, and Simulation*, pages 9–16. IEEE, 2009.

-
- [142] Aphrodite Veneti, Charalampos Konstantopoulos, and Grammati Pantziou. Continuous and discrete time label setting algorithms for the time dependent bi-criteria shortest path problem. *Operations Research and Computing : Algorithms and Software for Analytics*, pages 62–73, 2015.
- [143] Aphrodite Veneti, Charalampos Konstantopoulos, and Grammati Pantziou. An evolutionary approach to multi-objective ship weather routing. In *2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–6. IEEE, 2015.
- [144] Shanu Verma, Millie Pant, and Vaclav Snasel. A comprehensive review on NSGA-II for multi-objective combinatorial optimization problems. *IEEE Access*, 9 :57757–57791, 2021.
- [145] Serdar Vural, Pirabakaran Navaratnam, Ning Wang, Chonggang Wang, Lijun Dong, and Rahim Tafazolli. In-network caching of internet-of-things data. In *2014 IEEE international conference on communications (ICC)*, pages 3185–3190. IEEE, 2014.
- [146] Peter H Westfall and S Stanley Young. *Resampling-based multiple testing : Examples and methods for p-value adjustment*, volume 279. John Wiley & Sons, 1993.
- [147] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1) :67–82, 1997.
- [148] Tao Xie and Xiao Qin. Scheduling security-critical real-time applications on clusters. *IEEE transactions on computers*, 55(7) :864–879, 2006.
- [149] Qixuan Xue, Yongxin Zhu, Yajie Wang, Kedun Mao, Han Wu, Mengjun Li, Yishu Mao, and Junjie Hou. A scheduling scheme of task allocation in real time multiple-partition embedded avionics. In *2017 IEEE International Conference on Smart Cloud (SmartCloud)*, pages 41–46. IEEE, 2017.
- [150] Hamza Yapici and Nurettin Cetinkaya. A new meta-heuristic optimizer : Pathfinder algorithm. *Applied soft computing*, 78 :545–568, 2019.
- [151] Boyang Yu and Jianping Pan. Location-aware associated data placement for geodistributed data-intensive applications. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 603–611. IEEE, 2015.
- [152] Qingfu Zhang and Hui Li. MOEA/D : A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6) :712–731, 2007.
- [153] Eckart Zitzler, Simon Künzli, et al. Indicator-based selection in multiobjective search. In *PPSN*, volume 4, pages 832–842. Springer, 2004.
- [154] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. Spea2 : Improving the strength pareto evolutionary algorithm. *TIK-report*, 103, 2001.

-
- [155] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *Parallel Problem Solving from Nature—PPSN V : 5th International Conference Amsterdam, The Netherlands September 27–30, 1998 Proceedings 5*, pages 292–301. Springer, 1998.

Titre : Approches exactes et heuristiques pour l'exploration d'architectures logicielles et matérielles

Mot clés : Optimisation multi-objectif, systèmes embarqués, graphes, modélisation mathématique, méthodes évolutionnaires

Résumé : Les systèmes embarqués (SE) incorporent des fonctionnalités de plus en plus complexes. L'avènement dans les années 2000 des méthodes multi-objectifs a fourni des opportunités pour optimiser l'exploration des alternatives logicielles et matérielles concernant la définition et le déploiement des logiciels embarqués. Nos travaux exploitent de façon plus ou moins sophistiquée les méthodes et outils issus de la Recherche Opérationnelle (théorie des graphes, programmation linéaire MILP, méta heuristiques MOEA) pour modéliser et résoudre des problèmes d'optimisation, le plus souvent multi-critères à la fois en termes d'objectifs et de contraintes, associés à certains de ces SE dans des domaines divers : l'explora-

tion d'architectures pour optimiser l'ordonnement de systèmes temps réel ; le placement d'objet dans le Fog et le Cloud ; le routage pour les voiliers de course et l'optimisation de missions de drones.

Nous montrons dans ce document comment des méthodes génériques peuvent être adaptées pour résoudre efficacement les problèmes précités : modélisation ad-hoc, approches de type diviser pour régner sur les données basée sur la théorie des graphes, découpage en phases avec raffinement de l'espace de recherche, parallélisation MOEA, prise en compte de la dynamique de l'environnement en amont pour la définition d'une solution ou adaptation durant l'exécution d'une mission, etc.

Title: Exact and heuristic approaches for Design Space Exploration of software and hardware architectures

Keywords: Multi-objective optimisation, embedded systems, graph theory, mathematical modelling, evolutionary algorithms

Abstract: Software complexity of Embedded Systems (ES) increases continuously. The raise of Multi-Objective optimization techniques in the 2000 provides opportunities for Design Software Exploration (DSE) of such systems. Our research aims at adapting and exploiting Operation Research methods and tools for solving optimisation problems associated to the design and deployment of embedded software. Multiple optimisation criteria are most often considered, and we solve problems in

various area : DSE for scheduling of real-time systems, data placement in Fog or Cloud, weather routing for sailing boats or routing of mission for drones, etc.

We show how generic approaches can be specialized for efficiently solving our cited problems : ad-hoc or MILP model definition, divide and conquer approaches, space exploration refinement methods, parallel solving, dynamic mission adaptation for a drone, etc.