



HAL
open science

Une approche interdisciplinaire pour l'ordonnement des transports

Bernat Gacias

► **To cite this version:**

Bernat Gacias. Une approche interdisciplinaire pour l'ordonnement des transports. Automatique / Robotique. Université Paul Sabatier - Toulouse III, 2010. Français. NNT: . tel-00554205

HAL Id: tel-00554205

<https://theses.hal.science/tel-00554205>

Submitted on 10 Jan 2011

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.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse III - Paul Sabatier
Discipline ou spécialité : *Systèmes industriels et Systèmes informatiques*

Présentée et soutenue par *Bernat GACIAS PASTOR*
Le 25 novembre 2010

Titre : *UNE APPROCHE INTERDISCIPLINAIRE POUR L'ORDONNANCEMENT DES TRANSPORTS*

JURY

Christian Prins, Professeur à l'Université de Technologie de Troyes (Rapporteur)
Jean-Michel Hoc, Directeur de Recherche au CNRS (Rapporteur)
Philippe Palanque, Professeur à l'Université Toulouse III - Paul Sabatier
Bernard Grabot, Professeur à l'Ecole Nationale d'Ingénieurs de Tarbes
Pierre Dejax, Professeur à l'Ecole des Mines de Nantes
Manel Mateo, Maître de Conférences à l'Universitat Politècnica de Catalunya
Pierre Lopez, Chargé de Recherche au CNRS
Julien Cegarra, Maître de Conférences au Centre Universitaire Champollion

Ecole doctorale : *Ecole Doctorale Systèmes*

Unité de recherche : *LAAS-CNRS*

Directeur(s) de Thèse : *Pierre LOPEZ et Julien CEGARRA*

Rapporteurs :

Christian Prins, Professeur à l'Université de Technologie de Troyes
Jean-Michel Hoc, Directeur de Recherche au CNRS

Remerciements

Ce travail de thèse a été développé au sein du groupe MOGISA (Modélisation, Optimisation et Gestion Intégrée de Systèmes d'Activités) au Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS.

Je veux d'abord remercier mes directeurs de thèse, Pierre Lopez et Julien Cegarra, qui m'ont donné l'opportunité de participer à ce projet et qui m'ont offert leur expertise et leur support tout au long de la durée de ce travail. Je les remercie particulièrement pour leur implication dans ce travail et leur patience. Je veux également nommer Christian Artigues et Wout van Wezel qui ont aussi participé à la réalisation de ce travail.

Je n'oublie pas le soutien reçu de la part de ma famille, spécialement de mes parents et de mon frère Tomeu. Finalement, je veux fortement remercier à deux personnes très spéciales dans ma vie, Ale et Luis, qui sont restés près de moi pendant ces trois années et qui ont été mon plus fort soutien.

Résumé

Dans cette thèse, nous proposons d'aborder l'ordonnancement des transports par une approche interdisciplinaire. L'idée est d'intégrer les facteurs humains dans le système d'aide à la décision réalisé, de façon à ce que l'homme puisse agir sur la modélisation et la résolution du problème. Le système proposé doit offrir de la flexibilité, afin d'être capable de s'adapter aux nouvelles situations et aux changements, même si ceux-ci n'ont pas été prévus initialement par le concepteur du système. Pour atteindre l'objectif fixé, nous nous sommes notamment appuyé sur une analyse du domaine de travail (« Work Domain Analysis ») basée sur une hiérarchie d'abstraction des entités (physiques ou plus abstraites) manipulées dans ce type de problèmes. Nous avons proposé une architecture pour le système d'aide à la décision basée sur cette analyse du domaine et la programmation par contraintes. Nous avons également conçu, et intégré dans le système, des algorithmes dédiés et des méthodes de résolution basés sur le principe d'inversion de modèle. Enfin, nous avons proposé une architecture d'interfaces avec l'objectif d'assister efficacement l'opérateur humain dans la réalisation des différentes sous-tâches nécessaires à la résolution globale du problème. L'étude du sujet interdisciplinaire a été précédée d'une analyse focalisée sur la résolution de problèmes théoriques d'ordonnancement à machines parallèles avec contraintes de précédence et temps de préparation des machines entre opérations, utilisant des méthodes de recherche arborescente basée sur les divergences.

Mots-clés : Système d'aide à la décision – Problème de tournées de véhicules – Interfaces homme-machine – Inversion de modèle

Abstract

An interdisciplinary approach for the vehicle routing problem

An interdisciplinary approach has been proposed for the vehicle routing problem. The idea is to consider human factors and dynamic aspects for the decision support system (DSS) design. In our approach, a link is done between methods of operations research and an ecological interface design coming from engineering cognitive. A work domain analysis for the vehicle routing problem has been done. The analysis is realized through an abstraction hierarchy, which facilitates the identification of the problem constraints. We have proposed a DSS architecture based on this analysis and on constraint programming. Specific algorithms and solving mechanisms based on model inversion have been proposed and integrated in the system. Finally, we have design a set of human-machine interfaces in order to facilitate the problem solving to the human planning. The interdisciplinary study has been preceded by an analysis of the parallel machine scheduling problem with precedence constraints and setup times. Tree searches and local searches based on limited discrepancy search have been proposed to solve the problem.

Key-words : Decision Support System – Vehicle Routing Problem – Human-Machine Interfaces
–Model Inversion

Sommaire

Introduction	1
I Systèmes d'aide à la décision pour l'ordonnement des transports	5
I.1 Problèmes de décision	5
I.2 L'ordonnement des transports	6
I.3 Conception de systèmes interactifs	17
I.4 Une approche interdisciplinaire	22
I.5 Conclusion	25
II Le domaine de travail	27
II.1 Introduction	27
II.2 Analyse du domaine de travail	27
II.3 Approche proposée pour l'analyse du domaine	34
II.4 Conclusion	47
III Architecture décisionnelle pour l'ordonnement des transports	49
III.1 Introduction	49
III.2 Synoptique de l'architecture	50
III.3 Analyse des tâches pour la résolution du problème	51
III.4 Mécanisme de résolution	59
III.5 Résultats expérimentaux	77
III.6 Conclusion	80

IV Les interfaces homme-machine	83
IV.1 Introduction	83
IV.2 Interfaces écologiques	83
IV.3 Les interfaces du système	86
IV.4 Évaluation de l'interface pour la sélection des solutions	99
IV.5 Conclusion	109
V Assistance à l'opérateur dans la phase de résolution	111
V.1 Introduction	111
V.2 Problèmes non-réalisables	111
V.3 Inversion de modèle	114
V.4 Résultats expérimentaux	127
V.5 Conclusion	130
VI Résolution d'un problème d'ordonnement	131
VI.1 Introduction	131
VI.2 Problèmes d'ordonnement	131
VI.3 Problème à machines parallèles	134
VI.4 Recherche arborescente	142
VI.5 Recherche arborescente basée sur les divergences	147
VI.6 Résultats expérimentaux	155
VI.7 Conclusion	163
Conclusion et perspectives	165
Bibliographie	169

Introduction

Dans un contexte de compétitivité accrue, les entreprises agencent la logistique autour de transports de plus en plus réactifs. Les approvisionnements, la production et la distribution s'organisent, à partir de production sans stocks, pour permettre des livraisons "juste à temps". Par ailleurs, les coûts de transport dans une chaîne logistique peuvent représenter jusqu'à 30 % des coûts. Ceux-ci ne cessent d'augmenter ces dernières années ; on a par exemple assisté à une augmentation de plus de 7 % en 2008 par rapport à 2007 pour le transport routier. La gestion de la réactivité et des coûts doit, de plus, prendre en compte les réglementations nationales (par ex. 35 heures) et européennes (réglementations sur le temps de travail), qui impliquent de nouvelles problématiques sur la composition des transports, la constitution des équipes ou leurs créneaux horaires. Pour obtenir des livraisons à un coût compétitif, il est donc essentiel d'organiser très efficacement la chaîne logistique et les transports.

Dans cette thèse, nous nous intéressons aux problèmes liés à l'ordonnancement des transports (réalisation temporelle et spatiale des activités). Pour réaliser cet ordonnancement, on ne peut écarter l'homme, dans son rôle d'exécution puisqu'il est l'acteur de l'exécution de ces transports, mais aussi dans son rôle de décision car il est le seul à même de pouvoir intégrer contraintes techniques, réglementaires, environnementales et sociales, dans le but de fournir des solutions robustes. Le problème d'ordonnancement "purement technique" est un problème déjà difficile à résoudre du point de vue de sa complexité algorithmique. Actuellement, il existe cependant une large taxonomie des problèmes, ainsi qu'une panoplie importante de méthodes pour résoudre très efficacement les nombreuses variantes existantes. Le contexte de l'"ordonnancement humain" propre à ce travail intègre cette complexité tout en impliquant de se pencher vers d'autres types de considérations qui influent notamment sur la modélisation des problèmes considérés, le type et la représentation des solutions produites, et les moyens d'interactions (pour la construction

progressive et la révision des solutions). Les recherches réalisées sont donc naturellement inscrites dans un contexte interdisciplinaire, associant spécialistes des facteurs humains et des aspects mathématiques.

Dans l'étude proposée, nous nous focalisons sur la conception d'un système d'aide à la décision pour résoudre ce type de problèmes. Les mécanismes et outils d'analyse, de conception et d'évaluation de systèmes socio-techniques complexes proposés dans le domaine de l'Ingénierie Cognitive sont combinés avec des techniques de modélisation et résolution issues de la Recherche Opérationnelle. L'objectif de l'approche est de proposer un système d'aide à la décision qui privilégie la coopération homme-machine, sans pour autant pénaliser l'utilisation d'algorithmes efficaces pour la résolution du problème.

Dans ce manuscrit, nous traitons deux sujets clairement différenciés. La première partie du manuscrit est articulée en cinq chapitres. Dans cette partie, nous décrivons les différents aspects liés à la conception de ce système d'aide à la décision pour l'ordonnancement des transports. La deuxième partie, qui est composée d'un unique chapitre (le chapitre VI), consiste en une analyse focalisée sur la résolution de problèmes théoriques d'ordonnancement à machines parallèles avec contraintes de précédence et temps de préparation des machines entre opérations. Bien que la différence entre les deux parties soit évidente, il existe néanmoins un point commun. Une partie des méthodes et techniques de résolution développées dans le chapitre VI pour résoudre le problème d'ordonnancement ont ainsi été intégrées dans le mécanisme de résolution du système d'aide à la décision.

Le premier chapitre est consacré à une introduction du problème qui est ensuite conservée dans le reste du document. Dans ce chapitre, nous introduisons, d'un côté, l'ordonnancement des transports et, d'un autre côté, les méthodes de conception de systèmes coopératifs. Nous nous basons sur un état de l'art sur les systèmes d'aide à la décision décrits dans la littérature pour identifier les limitations des systèmes actuels et pouvoir ainsi focaliser notre travail sur ces aspects.

Une analyse du domaine de travail pour l'ordonnancement des transports est proposée dans le deuxième chapitre. Il s'agit notamment d'identifier l'ensemble des degrés de liberté offerts par l'environnement aux opérateurs humains et seulement ensuite de travailler sur les tâches réalisables au sein de ces invariants.

Le troisième chapitre est composé de deux aspects principaux. Dans un premier temps, nous

présentons une analyse des tâches à réaliser pour la résolution d'un problème d'ordonnement des transports. Nous nous sommes fondés sur cette analyse pour proposer une répartition des fonctions entre l'opérateur humain et la machine. La deuxième partie du chapitre consiste en la description du mécanisme de résolution et des algorithmes qui le composent. La répartition des fonctions a permis d'identifier les algorithmes qui doivent faire partie du mécanisme de résolution. Nous avons proposé des algorithmes qui permettent de satisfaire ces fonctions.

Dans le quatrième chapitre, nous décrivons les interfaces du système d'aide à la décision. La méthodologie de conception d'interfaces homme-machine s'appuie fortement sur les études de la Psychologie Écologique qui implique de considérer le point de vue du système homme-environnement comme unité d'analyse. Actuellement, il n'a pas été suffisamment pris conscience de l'importance de ce type d'interfaces dans l'ordonnement des transports.

Le cinquième chapitre est consacré à la description des outils destinés à assister l'opérateur dans la tâche de relaxation de contraintes lorsque le problème est non-réalisable. L'approche proposée pour la conception de ces outils est fondée sur l'idée du principe d'inversion du modèle. Des techniques de classification issues de l'analyse de données sont proposées pour être intégrées dans ces mécanismes.

Enfin, dans le sixième chapitre (et deuxième partie) nous traitons le problème d'ordonnement à machines parallèles avec considération de contraintes de précédence et de temps de préparation entre tâches. Nous proposons de nouveaux algorithmes pour la résolution de ce problème basés sur les principes des algorithmes hybrides de recherche locale et de recherche arborescente décrits dans la littérature.

Chapitre I

Systemes d'aide à la décision pour l'ordonnancement des transports

I.1 Problèmes de décision

Dans la réalité des problèmes d'ordonnancement des transports, des décisions doivent constamment être prises pour trouver une solution qui satisfasse les requêtes des clients sans violer les contraintes du problème. La volonté d'automatiser à outrance les systèmes de décisions a souvent conduit à placer l'humain à la périphérie du système décisionnel. Cependant, la nécessité de modifier les solutions proposées par ces systèmes, de façon à ce que les exigences qui ne peuvent pas être formalisées soient aussi satisfaites, mais aussi la nécessité de réagir rapidement face à l'imprévu, obligent à une participation de l'humain dans le processus de construction des solutions effectivement appliquées.

Dans un tel contexte dynamique, il est souvent nécessaire de prendre des décisions dans des intervalles de temps très courts. Cette réactivité est en grande partie assurée par les planificateurs humains, dont les connaissances et le savoir-faire acquis au cours du temps ne peuvent être modélisés de manière formelle. Cela a conduit à concevoir une approche coopérative où les systèmes décisionnels évoluent vers les systèmes d'aide à la décision. Le but de ces derniers est d'assister le planificateur dans le processus de prise de décisions.

Dans ce cadre, pour une conception efficace de ces systèmes d'aide à la décision, il est nécessaire d'identifier tout d'abord quelles sont les capacités et les manques de l'humain pour cha-

cune des actions à réaliser pendant le processus de résolution du problème. Par exemple, l'humain est très compétent pour arriver à trouver un compromis entre différents critères. En revanche, il est très inefficace pour vérifier et maintenir un ensemble important de contraintes, tâche où, au contraire, le système informatisé est performant. Ceci permet de concevoir des algorithmes et des mécanismes d'interaction davantage adaptés aux besoins du planificateur humain.

I.2 L'ordonnement des transports

La problématique liée à l'ordonnement des transports a été largement étudiée au cours des dernières décennies. Dans cette catégorie de problèmes, nous trouvons deux classes de problèmes qui ont beaucoup attiré l'attention des chercheurs : le problème de transport et le problème de tournées de véhicules.

Le problème de transport (transportation problem) fut formulé pour la première fois par Hitchcock (1941). Ce problème consiste à minimiser le coût de transport total d'un plan d'expédition. Le problème est un cas particulier d'un problème de graphe (le problème de flot maximal à coût minimal dans un graphe orienté biparti). Il peut être résolu avec un algorithme de complexité polynomiale (voir par exemple Nemhauser and Wolsey (1999); Lacomme et al. (2003)).

Le problème de tournées de véhicules ou VRP (Vehicle Routing Problem) a été introduit par Dantzig and Ramser (1959). Le problème consiste à déterminer les itinéraires à suivre par une flotte de véhicules de transport (de biens, de passagers) de manière à satisfaire un ensemble de requêtes clients (livraisons, collecte, ramassage, ...). Ce problème est une extension du problème du voyageur de commerce ou TSP (Travelling Salesman Problem) (Lawler et al., 1985; Gutin and Punnen, 2002). Il fait partie de la classe de problèmes *NP-complets* (Lenstra and Kan, 1981). Il n'existe pas d'algorithme de complexité polynomiale pour résoudre le problème. Toutefois, des méthodes issues de la Recherche Opérationnelle existent pour résoudre efficacement certaines des nombreuses variantes du VRP (Toth and Vigo, 2001; Cordeau et al., 2005; Barnhart and Laporte, 2007; Golden et al., 2008). Ces méthodes sont principalement de méta-heuristiques (recherche tabou, algorithmes génétiques, ...) qui permettent de trouver des solutions de bonne qualité en un temps de calcul raisonnable.

Dans les paragraphes suivantes, nous présentons plus profondément le problème de transport et le problème de tournées de véhicules. Ensuite, nous énumérons les caractéristiques et

particularités des différentes variantes du problème de tournées de véhicules. Finalement, nous présentons brièvement les systèmes d'aide à la décision pour l'ordonnancement des transports existant dans la littérature.

I.2.1 Définition du problème de transport

Le problème de transport peut être décrit de la façon suivante. Une quantité de produit uniforme est disponible en certains points appelés origines (par exemple dépôts, ports, ...). D'un autre côté, une quantité de produit est demandée en d'autres points appelés destinations (par exemple clients, points de vente, ...). Il s'agit d'envoyer les produits des origines aux destinations. Nous considérons qu'il existe toujours un chemin pour aller de chaque origine à chaque destination. Le coût de transport d'une unité de produit des origines vers les destinations est connu. L'objectif est de déterminer la distribution de produit qui permette de satisfaire la demande de chaque destination et de minimiser le coût total de transport.

Nous supposons un ensemble de m origines et n destinations. Notons a_i la quantité d'unités de produit disponible à l'origine i , et b_j la quantité d'unités de produit demandé par la destination j . Le coût de transport d'une unité de produit depuis l'origine i vers la destination j est noté c_{ij} . Les variables de décision x_{ij} représentent le nombre d'unités de produit envoyées de l'origine i vers la destination j .

Le problème peut être formulé comme un programme linéaire :

$$\begin{aligned} \min \quad & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \sum_{j=1}^n x_{ij} &= a_i \quad \forall i = 1..m \\ \sum_{i=1}^m x_{ij} &= b_j \quad \forall j = 1..n \end{aligned} \tag{I.1}$$

Il existe une solution réalisable pour le problème I.1 si la condition $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ est satisfaite. Le problème est convertible en un problème de flot maximum à coût minimal de la façon suivante. Une source s et un puits t fictifs sont ajoutés au graphe de liaisons possibles. Un arc (s, i) de capacité a_i relie la source s à toute origine i . Un arc (j, t) de capacité b_j relie toute destination j au puits t . Le graphe de liaisons (entre les origines et les destinations) est biparti, les capacités des arcs sont infinies et le coût par unité de produit est égale à c_{ij} , où i spécifie l'origine et j

la destination. Le problème revient donc à trouver un flot de débit égal à la disponibilité totale $\sum_{i=1}^m a_i$ mais de coût minimal. L'algorithme de Busacker and Gowen (1961) permet de résoudre le problème en $O(NM^2U)$, où N est le nombre de sommets du graphe, M est le nombre d'arcs et U est le maximum des disponibilités des origines. Il existe des algorithmes avec une meilleure complexité théorique, mais associés à une implémentation bien plus délicate.

Le graphe des liaisons peut être modifié de manière à s'adapter aux variantes du problème. L'algorithme proposé permet aussi de résoudre le problème dans le cas où la quantité de produit disponible est plus grande que la quantité demandée ($\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$). Ceci est fait à partir de la considération d'une destination fictive k avec une demande égale à $\sum_{i=1}^m a_i - \sum_{j=1}^n b_j$ qui permet d'absorber l'excès de produit disponible avec un coût de transport nul ($c_{ik} = 0 \forall i = 1..m$). Dans le cas contraire ($\sum_{i=1}^m a_i < \sum_{j=1}^n b_j$), l'algorithme arrive à trouver la distribution avec un coût de transport minimal, mais la demande des destinations n'est pas complètement satisfaite. Dans le cas où il n'est pas possible d'atteindre une destination j à partir d'une origine i , nous pouvons considérer une valeur très importante pour le coût de transport par unité de produit entre ces deux sommets par rapport aux autres coûts.

Des autres variantes du problème peuvent être aussi résolues avec le calcul du flot maximum avec un coût minimal sur un graphe légèrement modifié par rapport au graphe présenté pour résoudre le problème de transport de base. Par exemple, dans le problème de transbordement (cross-docking), des sommets sont introduits entre les sommets origines et les sommets destinations. Ces sommets intermédiaires servent à modéliser des possibles escales ou différentes alternatives (itinéraires, moyens de transport, ...) pour le transport depuis l'origine vers la destination (voir un exemple dans le paragraphe IV.4.2). L'objectif est de déterminer la solution (itinéraires, moyens de transport, ...) qui permette la distribution des produits tout en minimisant le coût de transport.

L'autre contrainte qui peut être facilement intégrée dans le problème de transport est la contrainte de capacité. Dans le problème de transport original, les arcs entre les origines et les destinations ont une capacité infinie. Nous pouvons envisager aussi bien dans le problème de transport que dans le problème de transbordement une limite dans la capacité maximale entre deux sommets. Cette capacité maximale de l'arc entre deux sommets peut représenter une limite liée au moyen de transport utilisé entre deux sommets (voir le paragraphe IV.4.2).

I.2.2 Définition du problème de tournées de véhicules (Vehicle Routing Problem - VRP)

Le problème de tournées de véhicules est un nom générique pour un ensemble de problèmes qui consistent à déterminer les itinéraires à suivre par une flotte de véhicules de transport (de biens, de passagers, ...) de manière à satisfaire un ensemble de requêtes clients (livraisons, collecte, ramassage) (cf. figure I.1).

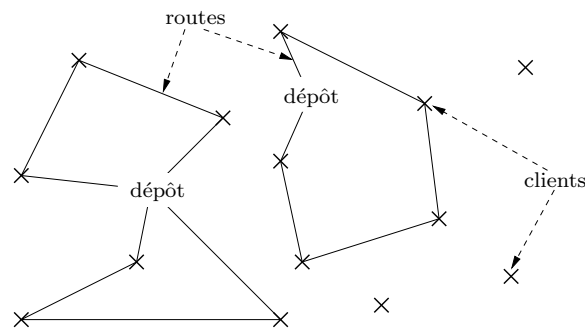


FIG. I.1 – Tournées de véhicules

Dans le problème de tournées de véhicules classique, nous considérons uniquement un dépôt et les tournées des véhicules doivent partir du dépôt et revenir à ce même dépôt. Chaque point de passage ou client doit être visité, mais chacun d'entre eux ne le sera qu'une seule fois et par une seule tournée. Les problèmes de tournées de véhicules se modélisent généralement à l'aide d'un graphe dans lequel les sommets représentent les points de passage ou clients et les arcs les axes routiers entre les clients. L'objectif le plus habituel est la minimisation de la distance parcourue ou du coût total de transport. Dans la littérature, nous trouvons plusieurs propositions de formulations linéaires pour ce problème (Miller et al., 1960; Sveska and Huckfeldt, 1973; Orloff, 1974; Bellmore and Hong, 1974).

Dantzig and Ramser (1959) introduisent pour la première fois la contrainte liée à la capacité limitée de transport des véhicules. Ce problème est aussi intitulé CVRP (Capacitated Vehicle Routing Problem). Dans ce problème, la demande de chaque client est modélisée par un poids affecté à chaque sommet. La somme totale des poids des sommets (demande des clients) appartenant à un même circuit ne peut pas dépasser la valeur d'une constante qui représente la valeur limite de la capacité des véhicules. Plusieurs modèles linéaires ont été proposés pour la

modélisation de cette variante du problème (Balinski and Quandt, 1964; Fisher and Jaikumar, 1981; Laporte et al., 1985). Ces formulations et des formulations additionnelles ont été compilées dans Laporte and Nobert (1987) et dans Toth and Vigo (2001).

Le CVRP est pour beaucoup d'auteurs le problème de base des problèmes de tournées de véhicules, car le problème sans la contrainte de capacité correspond au problème du voyageur de commerce avec la considération de multiples voyageurs. D'autres contraintes peuvent venir s'ajouter à ce problème de base. En effet, de nombreuses variantes du problème de tournées de véhicules sont présentes dans la littérature. Les problèmes réels d'ordonnancement de transport peuvent être souvent modélisés à partir de l'accumulation de plusieurs extensions du problème de tournées de véhicules. Pour cette raison, il est important de prendre en compte les contraintes considérées dans les extensions du problème de tournées de véhicules lors de la conception d'un système d'aide à la décision. Dans le paragraphe suivant, nous présentons de la manière la plus exhaustive possible les variantes du problème de tournées de véhicules.

I.2.3 Variantes du problème de tournées de véhicules

Le problème de tournées de véhicules a de nombreuses variantes ou extensions (voir Bodin et al. (1983) et Desrochers et al. (1990) pour une classification des différents types de problème). Dans ce paragraphe, nous présentons plus en détail les variantes les plus traitées dans la littérature.

Problème de tournées de véhicules avec fenêtres de temps

Le problème de tournées de véhicules avec fenêtres de temps ou VRPTW (Vehicle Routing Problem with Time Windows) est une des extensions les plus étudiées dans la littérature. Le VRPTW est l'extension du CVRP qui considère des contraintes temporelles sur l'horaire de passage des clients. Une fenêtre de temps est définie pour chaque client. Ces fenêtres spécifient des limites sur la date d'arrivée au plus tôt et sur la date d'arrivée au plus tard pour le service d'un client. Les fenêtres de temps molles peuvent être violées avec la considération d'un coût de pénalisation. En revanche, les fenêtres de temps strictes doivent être absolument respectées. Dans le cas où le véhicule arrive chez le client avant la date d'arrivée au plus tôt, le véhicule est alors obligé de patienter. Comme dans le CVRP, les objectifs d'optimisation les plus habituels sont la minimisation du coût de transport et des véhicules utilisés. Dans certains cas, la minimisation

du temps d'attente peut aussi faire partie des objectifs.

Dans la littérature, nous trouvons différentes formulations mathématiques pour ce type de problème (Solomon, 1987; Desrochers et al., 1988). Différents problèmes réels peuvent être aussi modélisés comme un problème de type VRPTW : la livraison de boissons ou de produits alimentaires, la livraison de journaux, la collecte de déchets industriels ou commerciaux (voir, *e.g.*, Golden et al. (2002); Kim et al. (2006)).

Problème de tournées de véhicules avec livraison et retour chargé

Le problème est connu avec le nom anglais de Vehicle Routing Problem with Backhauls ou VRPB. Cette variante du problème de tournées de véhicules permet indistinctement aux clients d'être livrés ou d'être collectés. Le problème considère les deux ensembles de clients séparément. Les clients de livraison vont recevoir une quantité de produit provenant du dépôt et les clients de collecte vont fournir une quantité de produit qui doit être ramenée au dépôt. La contrainte principale du problème est que pour chaque tournée les clients qui font partie des clients de livraisons doivent être servis avant les clients de collecte. Ceci implique qu'il faut que la livraison soit complètement finie pour commencer à collecter. Cette contrainte peut être modélisée par un ensemble de contraintes de précédence entre chaque client de livraison avec tous les clients de collecte.

Plusieurs formulations linéaires ont été proposées pour ce type du problème (Goetshalckx and Jacobs-Blecha, 1989; Toth and Vigo, 1997). Cette variante du problème est bien appropriée pour certains problèmes réels où le coût du réarrangement de la charge est important. En plus, Golden et al. (1985) soulignent l'importance d'une telle approche pour faire des économies dans les systèmes mixtes avec distribution et collecte, en opposition à une approche où la distribution et la collecte sont considérées comme deux problèmes traités de manière indépendante.

Problème de tournées de véhicules avec collecte et livraison

Ce problème est très proche du VRPB. Le nom en anglais est Vehicle Routing Problem with Pick-up and Deliveries (VRPPD). La caractéristique de ce type de problème est que chaque requête est définie par un sommet de collecte, un sommet de livraison et la demande à transporter entre ces deux sommets. Le VRP classique dans la version de livraison (respectivement de collecte) est donc considéré comme un VRPPD où les points de collecte (respectivement de

livraison) est le dépôt pour chaque requête.

L'application réelle la plus importante de ce problème est le transport à la demande (dial-a-ride problem or on-demand transportation problem, en anglais). Dans ce cas, il s'agit de transporter des personnes et le problème présente habituellement des contraintes additionnelles. Des fenêtres de temps peuvent être imposées pour le ramassage et pour le dépôt où, parfois, la contrainte considérée est de respecter une durée maximale entre le ramassage et le dépôt d'un usager.

Problème de tournées de véhicules avec livraison partagée

Le SDVRP (Split Delivery Vehicle Routing Problem) est une relaxation du problème de base de tournées de véhicules. Dans cette variante, les clients peuvent être servis par différents véhicules. Ceci implique la relaxation de la contrainte du CVRP qui limite à une le nombre de fois où un client peut être visité.

Problème de tournées de véhicules avec capacité sur les arcs

Ce problème peut être considéré comme une classe de problème différente plutôt que comme une extension du VRP. Dans le CARP (Capacitated Arc Routing Problem), la définition du client est différente par rapport à celle du CVRP. La différence principale entre les deux problèmes est que le CARP considère que les clients sont placés sur les arcs du graphe, contrairement au VRP où les clients sont représentés par les sommets du graphe. Ceci implique que chaque arête du graphe doit être traversée au moins une fois par un des véhicules.

Plusieurs problèmes réels peuvent être modélisés comme un problème de tournées de véhicules avec capacité sur les arcs, quelques exemples sont : la collecte de déchets, le nettoyage des rues, la distribution de courriers ou l'inspection de lignes électriques (voir Santos et al. (2008) pour un exemple d'application réelle).

Problème de tournées de véhicules avec contrainte sur la distance

Ce problème est connu avec le nom anglais de distance-constrained VRP. Le problème considère une borne limite pour la longueur totale des tournées. Cette contrainte peut s'appliquer à d'autres caractéristiques des tournées comme la durée, le coût ou le nombre de clients.

Des situations réelles peuvent être modélisées avec ce type de contraintes. Par exemple, Mendoza et al. (2009) utilisent ce type de contraintes pour modéliser une limite imposée sur la distance maximale à parcourir en une journée par un reviseur. Dans Ruiz et al. (2004), la contrainte est utilisée pour modéliser la limitation de la capacité de transport des véhicules qui disposent de compartiments de transport individualisés pour chaque client.

Problème de tournées de véhicules avec flotte hétérogène

Le problème de tournées de véhicules avec flotte hétérogène (Heterogeneous Fleet Vehicle Routing Problem ou HFVRP) est une variante du CVRP où chaque type de véhicule dispose d'une capacité de transport différente. Pour résoudre ce problème, en plus de la construction des tournées, il est aussi nécessaire de déterminer le nombre de véhicules de chaque type de véhicule à utiliser.

Problème de tournées de véhicules avec dépôts multiples

Le problème de tournées avec dépôts multiples ou MDVRP (Multi Depot Vehicle Routing Problem) est une extension qui considère plusieurs dépôts pour le départ et le retour des tournées. Chaque dépôt dispose d'une flotte de véhicules et le dépôt de début et de fin pour chaque tournée doit être le même.

Problème de tournées de véhicules périodique

Le problème de tournées de véhicules périodique ou PVRP (Periodic Vehicle Routing Problem) a été formulé par première fois par Beltrami and Bodin (1974). Dans le PVRP les tournées sont déterminées pour une période de temps (par exemple pour une semaine). Les clients peuvent être servis plusieurs fois dans cet horizon de temps. Une fréquence de passage dans l'horizon de temps est défini pour chaque client. Cette fréquence permet de déterminer les combinaisons de jours possibles pour le passage de chaque client. Pour résoudre un PVRP, il faut donc déterminer simultanément une combinaison de jours de passage pour chaque client et l'ensemble des tournées pour chaque jour de l'horizon de planification.

Ce type de problème se rencontre fréquemment dans les problèmes réels de collecte de déchets (produits recyclables, déchets industriels, ...) où un passage journalier n'est pas nécessaire pour chaque point de collecte.

Problème de tournées de véhicules combinées à la gestion de stocks

Cette variante nommée aussi IRP (Inventory Routing Problem) est très différente du VRP. La première différence est que, comme dans le PVRP, la planification des tournées est faite dans un horizon de temps. La deuxième différence qui caractérise ce type de problème est que les demandes ne sont pas imposées par le client, c'est-à-dire que la quantité de produit à livrer à chaque client fait partie des variables de décision. La seule condition à respecter est que les clients ne peuvent pas arriver à la rupture de stock de produit. Finalement, une fois prises les décisions concernant le moment où le client va être servi et la quantité concernée, il reste à déterminer l'ensemble des tournées des véhicules. L'objectif est de minimiser le coût de transport pour l'horizon de planification tout en assurant que les clients disposent à tout moment du produit.

Problème de tournées de véhicules stochastique

Le problème de tournées de véhicules stochastique ou SVRP (Stochastic Vehicle Routing Problem) correspond à un VRP où un ou plusieurs éléments sont modélisés par des variables aléatoires. Ces variables servent à modéliser l'incertitude vis-à-vis de certaines données du problème. Dans la littérature, les trois types de problèmes les plus étudiés sont le VRPSD (Vehicle Routing Problem with Stochastic Demand) qui correspond au problème de tournées de véhicules avec demandes stochastiques, le VRPSC (Vehicle Routing Problem with Stochastic Customers) qui associe une probabilité à la présence ou non des clients et le VRPSTT (Vehicle Routing Problem with Stochastic Travel Time) qui considère l'incertitude sur les temps de trajet.

Problème de tournées de véhicules dynamique

Le problème de tournées de véhicules dynamique ou DVRP (Dynamic Vehicle Routing Problem) fait partie des plus récentes extensions du VRP. La différence entre cette variante et les autres est que le DVRP modifie le modèle du problème au cours du temps. Actuellement, les avancées récentes en matière de télécommunications et de technologies de l'information donnent la possibilité de pouvoir accéder à de nombreuses informations très rapidement. Ces informations concernent habituellement l'évolution des données au cours du temps, mais parfois peuvent aussi entraîner l'apparition de nouvelles contraintes qui n'ont pas été prises en compte dans la première modélisation du problème. Ce fait explique l'émergence actuelle des problèmes de tournées dy-

namiques. Dans la littérature, nous pouvons identifier deux familles de problèmes de tournées dynamiques :

- Les problèmes où un ou plusieurs événements apparaissent au cours du temps une fois les tournées déjà en place ; ils vont ainsi ajouter de nouvelles variables et de nouvelles contraintes au modèle de base.
- Les problèmes où l'information sur un ou plusieurs événements évolue une fois les tournées déjà en place, c'est-à-dire que les valeurs de certains paramètres du modèle de base changent.

Ce dernier cas peut parfois être prévu à l'avance en modélisant l'information qui est susceptible d'évoluer par des variables aléatoires. Dans ce cas, le problème devient un problème de tournées de véhicules stochastique.

I.2.4 Systèmes d'aide à la décision pour les problèmes de tournées de véhicules

Les avancées en matière de technologies de l'information ont révolutionné la conception des systèmes d'aide à la décision pour les problèmes de tournées de véhicules. Les systèmes d'information géographique (GIS) sont de plus en plus utilisés dans la conception de systèmes d'aide à la décision dites spatiales (Spatial Decision Support Systems, SDSS). En effet, un grand effort est fait pour arriver à intégrer d'une manière efficace les GIS dans les systèmes les plus récents de la littérature. Ces systèmes d'information sont normalement reliés à des algorithmes d'optimisation spécifiques qui permettent de trouver des solutions de bonne qualité pour la variante de problème de tournées de véhicules considérée.

Dans ce contexte, Basnet et al. (1996) présentent un SAD pour la construction des tournées pour la distribution de lait en Nouvelle-Zélande. Le système exploite la capacité des planificateurs à reconnaître des patrons à partir des localisations des clients et des routes. Les auteurs proposent d'utiliser l'algorithme de "sweep" (Gillet and Miller, 1974) combiné avec l'algorithme d'insertion proposé par Syslo et al. (1983) pour la construction des tournées. Le système offre à l'utilisateur la possibilité de modifier les solutions. Pour cela, des entretiens avec des planificateurs ont permis d'identifier les principaux besoins pour la résolution du problème ; les informations pertinentes pour la prise de décision sont alors affichées pour aider le planificateur. Comme les auteurs l'admettent dans leur article, les algorithmes proposent des solutions qui parfois ne sont pas

réalisables. Ceci est une conséquence du fait que le système ne permet pas de prendre en compte toutes les contraintes qui interviennent dans le problème.

Ruiz et al. (2004) proposent un SAD interactif pour une entreprise de distribution d'aliments composés. Une approche exacte en deux phases (une énumération complète des tournées réalisables et un programme linéaire en nombres entiers qui en détermine la meilleure compte tenu d'une fonction objectif donnée) est proposée pour la résolution du problème de tournées de véhicules avec considération de contraintes particulières. Le système d'aide à la décision ne permet pas à l'opérateur de modifier les tournées. L'algorithme embarqué est le seul moyen de construire les tournées. L'algorithme propose à l'opérateur uniquement les solutions qui minimisent le coût de transport. Le problème d'un tel système est qu'il est complètement centré sur la machine et sur le mécanisme de résolution. L'opérateur ne peut que considérer les contraintes qui ont été identifiées lors de la conception du système d'aide à la décision. En plus, le système d'aide à la décision devient obsolète lorsque les caractéristiques du problème changent. Par exemple, le système ne considère pas la possibilité d'intégrer un nouveau centre de distribution (dépôt) au problème ou de modifier les caractéristiques de la flotte de véhicules.

Nous rencontrons ces inconvénients dans d'autres systèmes d'aide à la décision de la littérature. Par exemple, dans Tarantilis and Kiranoudis (2002), un SAD pour la conception de tournées de véhicules adapté à la ville d'Athènes est présenté. Un algorithme de recherche locale basé sur les opérateurs classiques de création de voisinages est proposé pour la résolution du HFVRP (Heterogeneous Fleet Vehicle Routing Problem) avec considération d'une durée maximale pour les tournées. Les auteurs proposent une architecture de données qui permet une intégration efficace d'un GIS. La solution proposée par le mécanisme de résolution est imposée à l'utilisateur, car l'opérateur ne dispose pas des outils d'interaction nécessaires pour modifier les solutions. D'autres exemples sont les systèmes d'aide à la décision proposés par Matsatsinis (2004) et par Ray (2007) qui sont aussi des systèmes avec une conception exclusivement centrée sur la machine. Le premier est un SAD pour une entreprise de distribution de ciment en Grèce. La variante de tournées de véhicules considérée est le problème de tournées de véhicules avec dépôts multiples et tournées multiples par véhicule et avec la considération de fenêtres de temps pour les clients (Multi-Depot Multi-Vehicle Routing with Time Windows). Le problème est résolu en deux étapes : un programme linéaire pour déterminer l'affectation des clients aux véhicules et une heuristique pour déterminer les itinéraires (l'ordre des clients) pour chaque véhicule. Le deuxième est un

SAD pour l'optimisation des tournées des véhicules longs/lourds où un algorithme hybride de recherche itérative est proposé. Dans ces trois systèmes (Tarantilis and Kiranoudis, 2002; Matsatsinis, 2004; Ray, 2007), l'opérateur humain joue un rôle complètement passif face au système. Une unique solution est proposée à l'opérateur et le SAD ne dispose pas non plus des outils nécessaires pour aider l'opérateur humain à modifier les solutions proposées.

Ces outils sont cependant disponibles dans les SAD proposés par Santos et al. (2008) pour la collecte de déchets au Portugal et par Mendoza et al. (2009) pour un problème de conception d'itinéraires pour un reviseur. Ces systèmes proposent des outils d'interaction homme-machine pour la modification des solutions et affichent aussi des informations sur les solutions qui peuvent être utiles à l'opérateur pour la réalisation de cette tâche. Santos et al. (2008) proposent comme algorithme une version modifiée de l'algorithme de "path-scanning" (Golden et al., 1983). Mendoza et al. (2009) proposent un algorithme génétique qui utilise une variante de l'heuristique "savings" (Clarke and Wright, 1964) pour obtenir la population initiale, une procédure de recherche locale basée sur les λ -voisinages proposés dans Osman (1993) pour améliorer les solutions et deux stratégies pour la réparation des solutions (la stratégie connue comme "split bin packing" et la stratégie proposée dans Prins (2004)). De plus, ce système propose plusieurs solutions alternatives, de manière à ce que l'opérateur puisse choisir la plus convenable en fonction du contexte. Toutefois, ces deux systèmes ne permettent pas la considération de contraintes qui n'ont pas été identifiées lors de la conception du système et se pose le problème de leur pérennité dans le cas de changement des caractéristiques du problème. La raison profonde en est que les caractéristiques spécifiques du problème sont intégrées au niveau de l'algorithme de résolution et elles ne peuvent pas être modifiées par l'opérateur lors de la modélisation du problème.

I.3 Conception de systèmes interactifs

I.3.1 Le système homme-machine

Un système peut être défini comme un ensemble de composants organisés interagissant pour parvenir à un objectif. Un système homme-machine est alors un système constitué de machines et d'agents humains. Contrairement à d'autres types de systèmes, les systèmes homme-machine ne sont pas complètement isolés. L'environnement peut avoir un effet important sur le fonctionnement du système.

Les systèmes homme-machine actuels sont de plus en plus complexes. Par exemple dans le domaine des transports, les avancées récentes en technologies de l'information ont supposé une augmentation considérable des transferts d'information et d'actions entre l'environnement et le système. Pour arriver à une conception efficace d'un système homme-machine, il est donc nécessaire de prendre en compte les caractéristiques de l'humain, des machines et de la nature des interactions entre l'humain et la machine et entre le système et l'environnement.

I.3.2 Les approches existantes pour la conception de systèmes

La littérature en conception de systèmes est extrêmement riche. De nombreux modèles de développement issus du domaine du génie logiciel ont été proposés au cours des dernières décennies. La tendance actuelle est d'enrichir ces modèles en considérant l'angle de l'interaction homme-machine (Kolski and Ezzedine, 2003). Ces approches s'avèrent efficaces pour la conception de systèmes de contrôle ou de supervision. Nous considérons qu'ils restent cependant insuffisants pour assurer une interaction homme-machine pertinente lorsqu'il s'agit de systèmes d'aide à la décision. En effet, les systèmes conçus avec ces approches ne permettent pas d'exploiter les capacités dont l'humain dispose pour résoudre les problèmes. L'humain y est généralement considéré comme un utilisateur qui interagit avec l'ordinateur uniquement pour l'aider à finir la résolution du problème, par exemple choisir entre deux solutions grâce à une interface. Une approche finalement plus efficace pour ce type de systèmes consiste à considérer l'humain comme un "opérateur". L'opérateur n'est pas limité à une relation homme-machine mais va aussi chercher des informations, négocier le relâchement de contraintes, etc. Dans ce cas, les tâches ne peuvent plus être analysées et modélisées uniquement sous un angle "utilisateur", il faut aussi analyser toutes les tâches de l'humain avec ou sans l'ordinateur pour ainsi arriver à concevoir un système capable de supporter l'opérateur dans chacune de ces tâches.

C'est dans ce contexte que l'ingénierie cognitive apparaît. L'ingénierie cognitive est une discipline émergente qui se place entre la psychologie cognitive, l'ergonomie et l'ingénierie des systèmes. L'objectif principal est de concevoir des méthodes, des outils et des techniques pour guider la conception de systèmes qui permettent de prendre en considération les facteurs humains. Pour cela, il est nécessaire de se focaliser sur les processus cognitifs de l'humain tels que la résolution de problèmes, le raisonnement, la prise de décisions, l'attention, la perception et la mémoire (Roth et al., 1994). La plupart des méthodes issues de l'ingénierie cognitive proposées

dans la littérature sont centrées sur l'analyse de la tâche (voir Diaper (2004) pour une revue de ces méthodes).

Les méthodes d'analyse de la tâche sont très appréciées dans les applications industrielles (Terrier and Cellier, 1999). Ces méthodes permettent d'identifier les actions à réaliser et les besoins d'information pour la réalisation de la tâche. Elles présentent deux limitations majeures. La première est que ces analyses sont souvent utilisées pour identifier les contraintes qui interviennent dans la réalisation de la tâche et pouvoir ainsi les intégrer dans les algorithmes de manière à automatiser le plus possible sa réalisation. Ce modèle sous-estime les capacités de l'humain qui est relégué à un rôle plutôt passif lors de la résolution du problème. De nombreuses études ont montré que les systèmes hybrides avec une bonne coopération homme-machine s'avèrent plus efficaces que la machine ou l'homme seuls.

La deuxième limitation est que les méthodes basées sur l'analyse de la tâche ne sont pas indépendantes de la tâche analysée. Le système conçu reste alors insuffisant face aux situations qui n'ont pas été traitées lors de la phase de conception, car une énumération complète de tous les scénarios possibles n'est pas envisageable (question de coût, temps, ...). On espère alors que l'opérateur sera capable d'affronter ces situations en proposant des solutions satisfaisantes, ceci sans aucun support efficace puisque le système n'est pas capable de garantir la complétude et la cohérence des solutions proposées dans un contexte non prévu.

Des méthodes alternatives sont alors nécessaires pour arriver à concevoir des systèmes qui permettent une intégration efficace de l'ergonomie. Une de ces méthodes est l'Analyse Cognitive du Travail (Cognitive Work Analysis - CWA). Le CWA articule l'analyse de la tâche à un ensemble d'étapes d'analyse du travail qui permettent aussi de prendre en considération les contraintes environnementales du système de travail.

Analyse cognitive du travail (Cognitive Work Analysis - CWA)

L'analyse cognitive du travail a été proposée comme une méthode pour la conception de systèmes complexes (Rasmussen, 1986a; Rasmussen et al., 1994; Vicente, 1999b). La méthode a ses origines dans l'approche de la psychologie écologique de la perception (Gibson, 1979). La philosophie de la psychologie écologique est de fonder l'analyse de la perception sur les rapports entre l'organisme vivant et son environnement. Ce principe de base a été repris dans le domaine de l'ingénierie cognitive par Rasmussen et ses collègues. Le CWA propose une analyse

du système en se focalisant sur les contraintes imposées par l'environnement sur le comportement de l'opérateur. Cette méthode s'écarte des méthodes classiques pour la conception de systèmes basées sur l'analyse des tâches réalisées par l'humain. Le CWA souligne l'importance de la description des limites du domaine de travail et des contraintes imposées pour ce domaine pour arriver à identifier (sans limiter) les possibilités de comportement de l'opérateur humain.

La méthode se situe dans les approches formatives selon la classification proposée par Rasmussen (1997) pour les méthodes d'analyse du travail. L'idée principale de l'approche est que l'opérateur doit terminer la conception du système. L'approche formative se focalise donc sur l'identification des exigences qui doivent être satisfaites, en opposition aux approches normatives (comme c'est le cas dans la plupart des méthodes d'analyse de la tâche) et descriptives qui cherchent respectivement à modéliser la tâche qui doit être réalisée et à décrire la manière dont la tâche est réalisée par les opérateurs (voir Cegarra and van Wezel (2010) pour une discussion plus approfondie sur les différents types d'approches appliquées aux problèmes d'ordonnancement). Un des avantages des méthodes formatives est que les différentes possibilités d'action sont présentées à l'opérateur et celui-ci doit choisir la mieux adaptée, contrairement aux autres approches qui cherchent à diriger et guider de manière beaucoup plus directe l'opérateur vers la tâche ou l'activité connue. Ceci a des conséquences très importantes pour la conception de systèmes. La première est que le système conçu à partir d'une approche formative reste indépendant de la stratégie utilisée par l'opérateur pour la réalisation de la tâche. La seconde est que le système est censé toujours assister l'opérateur aussi bien dans les situations familières que dans les situations moins habituelles qui n'ont pas été prévues lors de la conception du système.

La méthode se déroule en cinq étapes : l'analyse du domaine de travail, l'analyse de la tâche, l'analyse des stratégies, l'analyse socio-organisationnelle et l'analyse des compétences de l'opérateur. A travers les cinq étapes de la méthode, le concepteur arrive à la prise en compte des aspects cognitifs de l'opérateur à partir des aspects écologiques ou environnementaux décrits principalement dans la première étape. Dans la littérature, nous trouvons des exemples de systèmes d'aide à la décision pour les problèmes d'ordonnancement qui ont été conçus en ignorant les contraintes cognitives et qui ont montré leur inefficacité (difficulté d'utilisation, inadaptation aux modes opératoires des opérateurs expérimentés, ...) (voir Hoc et al. (2004)).

L'analyse du domaine de travail est la première et la principale étape du CWA. Cette étape consiste en la réalisation d'une description du domaine de travail du système. Ce dernier est vu

comme l'objet de l'action, en opposition à l'analyse de la tâche qui décrit les objectifs de l'action. La représentation du domaine de travail décrit la structure du système indépendamment des opérateurs, du degré d'automatisation, des événements, des tâches, des objectifs ou des interfaces. L'analyse consiste en l'identification des contraintes imposées par le domaine sur le système et qui vont limiter le comportement des agents d'exécution (homme ou machine). L'objectif de cette étape est d'identifier l'information nécessaire à fournir à l'opérateur dans des situations inattendues ou non familières. Cette description du domaine reste donc indépendante vis-à-vis des événements et des moyens, mais est par contre complètement dépendante de la raison d'être (but) de l'analyse. Une analyse du domaine de travail pour l'ordonnancement des transports est présentée dans le chapitre II comme la première étape pour la conception d'un système d'aide à la décision pour ce type de problèmes.

La deuxième étape de la méthode est l'analyse de la tâche. D'après Vicente (1999b), l'analyse de la tâche est une étape complémentaire à l'analyse du domaine de travail. En effet, même si l'analyse de la tâche ne permet pas d'identifier l'information nécessaire pour le support de l'opérateur humain face à des situations non familières ou inattendues, cette deuxième étape peut être utile pour identifier et décrire ce qui doit être fait face lors de situations habituelles, sans spécifier pour autant comment et pour qui cela doit être fait. L'objectif de cette analyse est donc de parvenir à identifier l'information nécessaire pour l'assistance à l'opérateur et également les contraintes qui doivent être prises en compte pour la sélection de la stratégie à suivre pour la réalisation de la tâche.

L'analyse des stratégies est la troisième étape de la méthode. Dans cette étape, le concepteur commence à s'intéresser aux aspects cognitifs de l'utilisateur dans la résolution du problème. Si dans l'étape d'analyse de la tâche l'intérêt est de déterminer ce qu'il doit être fait, dans cette étape le concepteur s'intéresse à la description du processus de comment les tâches doivent être réalisées. L'objectif de cette étape est donc de décrire les différentes stratégies possibles, de manière à identifier les besoins de chaque stratégie (information à fournir à l'opérateur) pour arriver à réaliser la tâche. Le système est ainsi capable de fournir l'assistance nécessaire à l'opérateur humain (même pour les stratégies de grande charge mentale), indépendamment de la stratégie choisie pour la réalisation de la tâche.

L'analyse socio-organisationnelle et de la coopération est la quatrième étape de la méthode. Une fois le domaine décrit, les tâches à réaliser définies et les différentes stratégies analysées,

cette étape aborde, d'une part, la répartition des rôles entre l'opérateur humain et la machine et, d'autre part, les formes de communication et de coopération des différents acteurs. L'objectif de cette analyse est de répartir les différents besoins identifiés dans les analyses des trois premières étapes entre les différents acteurs (opérateur ou groupe d'opérateurs humains et machine ou groupe de machines). Ainsi, le résultat de cette étape est une affectation des rôles et des responsabilités pour chaque acteur et une structure organisationnelle qui définit comment les acteurs doivent travailler (par groupes, par équipes, ...), comment doivent être instituées la communication et la coopération entre ces acteurs et quelles sont les relations de hiérarchie entre les unités organisationnelles pour cette coopération.

Enfin, la dernière étape de la méthode est l'analyse des compétences de l'opérateur. Cette étape cherche à identifier, à partir des besoins identifiés dans les étapes précédentes de la méthode, quelles sont les compétences dont l'opérateur doit faire preuve. Cette analyse est à tout moment cohérente avec les limitations et les capacités cognitives de l'humain.

I.4 Une approche interdisciplinaire

L'objectif principal de ce travail est de proposer une approche interdisciplinaire appliquée à la résolution des problèmes d'ordonnancement des transports. Nous nous intéressons plus particulièrement à la conception d'une architecture générique ainsi qu'aux outils d'interaction homme-machine nécessaires pour un système d'aide à la décision pour résoudre les problèmes de tournées de véhicules. Nous proposons une articulation entre des méthodes de recherche opérationnelle (la programmation par contraintes et des algorithmes dédiés pour les problèmes de tournées de véhicules) et une analyse du domaine de travail.

Nous avons constaté tout au long de ce chapitre certains manques dans les approches classiques de conception de systèmes, notamment pour les systèmes d'aide à la décision pour les problèmes de tournées de véhicules. Dans la plupart de ces systèmes, le problème est abordé dans sa globalité de manière automatique et permettent peu de coopération entre l'homme et la machine. Par exemple, nous avons relevé des SAD où une unique solution est proposée à l'opérateur. Jüngen and Kowalczyk (1995) notent cependant qu'une bonne performance d'une solution n'implique pas une acceptation irréfléchie de la part de l'opérateur, particulièrement à cause de l'existence de certaines décisions qui s'opposent directement au point de vue de l'hu-

main. D'un autre côté, une solution qui viole certaines contraintes du modèle initial peut être parfois jugée comme bonne.

Dans les SAD actuels, la considération des facteurs humains est très majoritairement absente des phases de modélisation et de résolution des problèmes. Il s'agit notamment du rôle du planificateur ou superviseur de tournées qui peut contribuer à l'élaboration de la tournée, si tant est qu'on lui laisse la possibilité d'agir sur les contraintes du problème traité. En effet, il peut disposer de connaissances ou d'heuristiques pour anticiper les aléas tels que les commandes urgentes, les pannes de véhicules, les embouteillages sur certaines routes, le remplacement de chauffeurs, etc. (voir Cegarra (2008) pour une discussion sur la contribution de l'humain pour des décisions d'ordonnancement). Il est donc important de permettre à l'humain d'agir sur les contraintes pour qu'il puisse contribuer à la robustesse de la solution produite par l'outil.

La deuxième des limitations que nous avons remarquée est liée au fait que les modèles sur lesquels s'effectuent la résolution sont imperméables aux changements rapides de situations. En effet, la dynamique dans ce domaine est très forte et, dans certains cas extrêmes, les contraintes peuvent avoir été modifiées avant même la fin de la phase de modélisation. D'après nous, le système d'aide à la décision doit permettre une modification complète du modèle du problème. Dans le cas contraire, les systèmes deviendront obsolètes dès qu'un nouveau type de contrainte s'ajoute au problème car les solutions proposées ne sont plus réalisables.

Finalement, une autre des limitations rencontrées est que les algorithmes et les interfaces sont souvent conçus pour résoudre un problème de tournées de véhicules et non pour résoudre les difficultés rencontrées par l'humain lorsque c'est lui qui résout tout ou partie du problème. En effet, nous avons vu des systèmes avec un unique algorithme de résolution et aussi avec des interfaces qui permettent seulement une interaction très légère entre l'utilisateur et le système (modifications locales de la solution finale avec un faible support, voire aucun, de la part du système et modifications des valeurs de certaines contraintes du modèle initial du problème). Dans les systèmes que nous avons décrits dans le paragraphe I.2.4, certaines tâches doivent être réalisées par l'humain, comme la sélection des véhicules à utiliser, le réaménagement final des tournées, de manière à satisfaire certaines contraintes qui n'ont pas pu être considérées dans le modèle initial, ou la relaxation de certaines contraintes lorsque le problème n'est pas réalisable. Les systèmes actuels n'offrent aucun support spécifique (ni algorithmes, ni interfaces) pour faciliter la réalisation de ces tâches à l'opérateur.

Dans ce travail, nous adoptons une démarche interdisciplinaire avec l'objectif d'aborder les limitations qui n'ont pas été correctement traitées auparavant. D'abord, nous réalisons une analyse du domaine de travail (première étape du CWA) qui permet d'identifier les objets et les contraintes du problème importantes pour la résolution. Cette analyse constitue une première étape dans le développement d'un système d'aide à la décision pour le problème de tournées de véhicules. Le modèle issu de cette analyse a vocation d'être robuste et permet de contourner les écueils énoncés précédemment. Il est ainsi bien adapté à la prise en compte de l'humain en tant qu'acteur du processus de décision et agent d'exécution (voir par exemple Higgins (1999, 2001)). De plus, les contraintes du problème étant considérées de manière exhaustive dans le modèle, ce dernier doit pouvoir absorber plus facilement les changements et perturbations qui surviendront en cours d'exécution des plans.

Ensuite, nous présentons l'architecture générique d'un système d'aide à la décision qui nous paraît bien adaptée pour la considération d'une coopération homme-machine. Dans cette architecture, l'humain se situe au centre du processus de prise de décisions. Nous avons intégré dans le système des algorithmes efficaces issus de la recherche opérationnelle susceptibles de fournir un support à l'opérateur pour chaque tâche à réaliser au cours de la résolution du problème. Nous proposons des algorithmes pour la construction des solutions ainsi que des algorithmes de vérification de la faisabilité des décisions afin d'aider l'opérateur humain à prendre les décisions correctes.

Les algorithmes dédiés ne sont pas le seul support pour l'opérateur. Dans le cadre de la conception d'interfaces écologiques (Ecological Interface Design, EID) proposé par Rasmussen and Vicente (1989), nous proposons des interfaces spécifiques pour la réalisation de chaque tâche. Ces interfaces ont la particularité d'afficher d'une manière perceptivement évidente l'information nécessaire de manière à aider l'opérateur à établir la séquence de décisions à prendre pour la résolution du problème. Chacune des interfaces proposées dispose d'outils d'interaction homme-machine particuliers qui facilitent à l'opérateur humain ce processus de prise de décisions.

Enfin, dans la logique de proposer des outils de support à l'opérateur pour chaque tâche à réaliser, nous utilisons le concept d'inversion de modèle avec le but d'aider l'opérateur humain à relâcher les contraintes lorsque le problème initial est non réalisable ou lorsque les décisions qui ont été prises dérivent en un problème sans solution. Ce sujet a été très peu abordé dans la littérature, nous trouvons uniquement les travaux de Jussien (2001) où la relaxation des

contraintes est abordé d'un point de vue strictement algorithmique. Nous pensons que la relaxation de contraintes ne peut être faite qu'avec une participation directe de l'opérateur humain. L'idée est de concevoir des algorithmes et des interfaces avec le but de guider l'opérateur humain vers les contraintes à relâcher et comment faire ceci de manière efficace.

I.5 Conclusion

Dans ce chapitre, nous avons introduit le contexte de notre travail. Nous avons tout d'abord présenté la problématique liée à l'ordonnancement des transports et à la prise de décisions. Nous avons ensuite décrit brièvement les approches existantes pour la conception de systèmes interactifs, notamment le CWA (Cognitive Work Analysis) qui, à notre avis, est la méthode qui favorise le plus une approche interdisciplinaire pour la conception de systèmes d'aide à la décision. Puis, nous avons énuméré les limitations existantes dans les systèmes d'aide à la décision actuels pour, enfin, introduire les solutions que nous proposons pour surmonter ces limitations. Ces apports sont présentées dans les chapitres suivants.

Chapitre II

Le domaine de travail

II.1 Introduction

Dans ce chapitre, nous proposons une analyse du domaine de travail pour le problème d'ordonnement des transports. Cette analyse est proposée comme la première et principale étape de la méthode CWA (Cognitive Work Analysis) (Rasmussen et al., 1994; Vicente, 1999b) présentée dans le paragraphe I.3.2. L'analyse du domaine de travail (Work Domain Analysis – WDA) est fondée sur des techniques de modélisation du domaine. Nous proposons une hiérarchie d'abstraction pour la modélisation du domaine. Le modèle issu de cette analyse a la vocation d'être robuste et permet de contourner les écueils énoncés dans le chapitre I. De plus, les contraintes nécessaires pour la résolution du problème étant considérées de manière exhaustive dans le modèle, ce dernier doit pouvoir absorber plus facilement les changements et perturbations qui surviendront en cours d'exécution des tournées.

II.2 Analyse du domaine de travail (Work Domain Analysis – WDA)

L'analyse du domaine de travail est une méthode qui permet d'identifier les contraintes du problème. Comme le préconisent (Rasmussen, 1985; Rasmussen et al., 1994; Vicente, 1999b), la hiérarchie d'abstraction constitue l'outil de modélisation recommandé pour procéder à l'analyse

Une partie de ces travaux ont été publiés dans Gacias et al. (2009)

du domaine de travail. Cette hiérarchie permet de décrire le domaine selon différents points de vue fonctionnels. La hiérarchie d'abstraction appartient à la classe des *hiérarchies stratifiées* décrites par Mesarovic et al. (1970). Les caractéristiques de cette classe de hiérarchies sont :

- Chaque strate ou niveau est une représentation du même système, la seule différence est que différents niveaux apportent des descriptions ou des modèles différents pour l'observation du système.
- Chaque niveau dispose de son propre ensemble de termes, concepts et principes.
- La sélection d'un niveau pour décrire un système dépend des observateurs et de leurs connaissances et intérêts pour le contrôle du système.
- Les exigences de fonctionnement au sein d'un même niveau apparaissent comme des contraintes pour la compréhension des opérations aux niveaux inférieurs. En même temps, l'évolution de l'état du système est spécifiée pour l'effet des niveaux inférieurs sur les niveaux supérieurs.

A l'origine, la hiérarchie d'abstraction a été proposée comme une méthode d'analyse pour le contrôle de processus continus. Rasmussen and Goodstein (1987) suggèrent que l'information basique nécessaire qui permet aux opérateurs d'identifier la séquence complète de contrôle du système peut être identifiée à partir d'une hiérarchie d'abstraction. Rasmussen (1985) propose cinq points de vue ou niveaux pour décrire le domaine d'un système de contrôle d'un processus. Les niveaux les plus hauts de la hiérarchie décrivent l'information fonctionnelle du domaine. Les niveaux les plus bas permettent de décrire l'information physique. En plus de cette décomposition "fins-moyens", une décomposition "tout-partie" est normalement réalisée en parallèle. Cette décomposition permet de considérer plusieurs niveaux de détail (e.g., systèmes, sous-systèmes, composants). Du point de vue le plus fonctionnel au plus physique, les cinq niveaux de décomposition habituellement utilisés sont :

- Le niveau des *objectifs fonctionnels*, qui représente les objectifs pour lesquels le système de travail a été conçu et les contraintes gouvernant les interactions entre le système de travail et son environnement.
- Le niveau des *fonctions abstraites*, qui représente des lois permettant d'effectuer des mesures et, par conséquent, d'établir les priorités qui dirigent le domaine. Ces lois et priorités confèrent une certaine régularité au comportement fonctionnel du système de travail.

- Le niveau des *fonctions générales*, qui représente les fonctions fondamentales et les activités de travail réalisées. Ces fonctions correspondent au domaine vu sous l’angle des mécanismes qui le définissent.
- Le niveau des *processus physiques*, qui spécifie les caractéristiques et les liens des composantes du domaine.
- Le niveau des *formes physiques*, qui constitue l’ensemble des ressources matérielles du domaine ainsi que leurs caractéristiques en termes d’appartenance et de localisation spatiale.

La compréhension du système de travail augmente donc avec le parcours des niveaux. Ainsi, remonter dans la hiérarchie d’abstraction permet une compréhension plus profonde du sens du système de travail vis-à-vis des objectifs à atteindre. Inversement, descendre les niveaux apporte une explication plus détaillée du fonctionnement du système de travail pour parvenir à ces objectifs.

Vicente (1999c) propose une méthode de classification pour les méthodes d’analyse du travail. Cette classification est basée sur les propriétés de dépendance vis-à-vis des moyens, de dépendance vis-à-vis des événements et de pertinence du point de vue psychologique. Dans l’article, les différentes méthodes d’analyse de travail sont évaluées par rapport à ces trois propriétés. D’après Vicente (1999c), l’analyse du domaine de travail à travers une hiérarchie d’abstraction est la meilleure méthode en comparaison avec des méthodes d’analyse de la tâche, de l’activité ou du système.

- La dépendance vis-à-vis des moyens exprime le niveau de dépendance de la technique d’analyse par rapport aux caractéristiques de l’acteur (opérateur humain, groupe ou équipe d’opérateurs, ordinateur, ...) qui réalise le travail. Les techniques d’analyse indépendantes vis-à-vis des moyens sont préférables. Ces techniques n’introduisent pas de limites lors de la conception du système d’aide à la décision et favorisent aussi l’affectation de nouvelles fonctions pour les acteurs, par exemple dans le cas d’une re-conception ou modification du système d’aide à la décision. La hiérarchie d’abstraction est une méthode qui présente cette propriété d’indépendance vis-à-vis des moyens, car à aucun moment la description du domaine n’est dépendante des acteurs qui exécutent les tâches.
- La dépendance vis-à-vis des événements spécifie si la technique d’analyse est utilisée pour identifier les besoins d’information pour l’assistance à l’opérateur humain de manière indépendante à une classe d’événement, tâche, situation ou contexte. L’indépendance vis-à-vis

des événements permet de soutenir l'opérateur même dans des situations non familières qui n'ont pas été prévues lors de la conception du système d'aide à la décision. La hiérarchie d'abstraction est indépendante vis-à-vis des événements. En effet, la décomposition n'est pas basée sur l'énumération de tous les événements connus mais sur l'identification des contraintes du domaine dans le fonctionnement habituel du système de travail.

- La pertinence du point de vue psychologique spécifie la capacité de la technique d'analyse à produire comme résultat une représentation ou modèle psychologiquement plausible. Vicente (1999c) présente la hiérarchie d'abstraction comme une méthode pertinente du point de vue psychologique. L'auteur affirme que la méthode se focalise sur l'identification des objectifs fonctionnels majeurs du système de travail et que ceux-ci sont des objectifs pertinents pour l'opérateur humain. Dans l'article, cette affirmation est argumentée par le fait que le comportement humain est d'une manière générale fortement dirigé par les objectifs. Dans Vicente and Rasmussen (1992), l'auteur renforce cet argument en s'appuyant sur quelques résultats empiriques qui affirment que la manière de raisonner des opérateurs humains lors de la résolution d'un problème peut être représentée au travers d'une hiérarchie d'abstraction.

Les propriétés désirables pour une méthode d'analyse de systèmes de travail sont donc ; l'indépendance vis-à-vis des moyens, l'indépendance vis-à-vis des événements et la pertinence du point de vue psychologique du résultat de l'analyse de manière à que le système de support conçu à partir de l'analyse soit capable, en tout instant, de répondre aux besoins de l'opérateur.

L'analyse d'un domaine se veut la plus exhaustive possible. Une connaissance approfondie du domaine étudié est donc nécessaire pour la réalisation de l'analyse. Comme Vicente (1999a) le prétend, une décomposition de ce type doit promouvoir des interfaces, dites "écologiques", menant à une meilleure performance que les interfaces traditionnelles (voir paragraphe IV.2). La hiérarchie doit être considérée comme un outil de représentation du domaine de contraintes qui est utilisé comme support pour le concepteur. Il est alors nécessaire de déterminer de façon exhaustive les caractéristiques les plus importantes du domaine lors de la décomposition. Néanmoins, les tentatives de décomposition de problèmes d'ordonnancement et de transport proposées dans la littérature utilisent des approches parfois contradictoires et ouvrent certains débats sur la démarche à suivre pour la décomposition de ce type de domaines (intentionnels mais dynamiques en même temps) (Krosner et al., 1989; Kinsley et al., 1994; Wong et al., 1998; Higgins, 1999).

II.2.1 Analyse du domaine de travail dans les problèmes d'ordonnement

La CWA s'est avérée une méthode efficace pour la conception d'outils de contrôle de systèmes physiques fondés sur des lois physiques. Des exemples, parmi d'autres, sont l'analyse du domaine pour la supervision des centrales nucléaires (Itoh et al., 1995) ou électriques (Burns, 2000), ou encore l'analyse pour le processus de fabrication de ciment (van Paassen, 1995). Ces analyses ont été menées dans des systèmes de travail causaux où l'état du domaine est dirigé par les lois du domaine qui s'imposent à l'utilisateur. Toutefois, dans les systèmes de travail intentionnels (comme le problème d'ordonnement ou de transport), l'utilisateur est le centre du domaine et l'état de celui-ci est prioritairement déterminé par ses intentions. Dans la littérature, nous pouvons trouver quelques tentatives (rares) d'application de la méthode CWA dans des domaines intentionnels (Wong et al., 1998; Higgins, 2001; Morineau et al., 2005).

Le tableau II.1 montre les différentes décompositions des domaines qui ont été proposées pour les problèmes d'ordonnement et de transport. Nous pouvons observer des décompositions relativement différentes pour des domaines très similaires. Ceci est dû aux différents méthodes utilisés pour l'analyse. Certaines décompositions sont fortement focalisées sur le domaine avec l'utilisation de termes comme "masse" ou "retard" qui sont des concepts indépendants des événements et des moyens. D'un autre côté, d'autres décompositions s'appuient sur des activités humaines pour définir le domaine (*e.g.*, "satisfaire les plans de production") ou font référence à des moyens spécifiques (*e.g.*, "enregistrer appel").

Ces différents points de vue mettent en évidence les difficultés auxquelles les chercheurs sont confrontés lors de l'analyse de domaines pour les systèmes d'ordonnement et de transport. La description du domaine pour les systèmes intentionnels est parfois difficile à déterminer. Dans les systèmes physiques, la situation évolue de façon indépendante de l'humain. D'un autre côté, dans les systèmes intentionnels, la plupart des évolutions et changements dépendent des décisions prises par l'humain. Ceci est la différence par exemple entre la supervision d'une centrale électrique et la conception d'un horaire de trains proposée par un opérateur. Cette différence est particulièrement importante dans le plus bas niveau de la hiérarchie (les objets physiques). Dans le tableau II.1, nous trouvons les machines dans les niveaux des objets physiques des différentes décompositions proposées pour les problèmes d'ordonnement. L'opérateur peut

	Ordonnancement				Transports	
	<i>Système de production de circuits imprimés</i> (Krosner et al., 1989)	<i>Système complexe de fabrication</i> (Kinsley et al., 1994)	<i>Production manufacturière</i> (Moray, 1999)	<i>Ordonnancement job-shop</i> (Higgins, 2001)	<i>Distribution d'ambulances</i> (Wong et al., 1998)	<i>Distribution militaire</i> (Hajdukiewicz et al., 1999)
Objectifs fonctionnels	Minimiser les coûts, atteindre les objectifs de production	Produire les pièces correctement et efficacement	Produire 168 articles en 12 heures pour satisfaire la commande	Maximiser le retour financier à long terme	Distribuer les ambulances selon les urgences médicales	Soin des personnes blessées avec considération des risques et des ressources
Fonctions abstraites	Produire les pièces, réduire la déviation par rapport aux dates cibles, satisfaire les plans de production	Masse, retard, valeur, priorités	Satisfaire les plans de production	Maximiser la viabilité financière à court terme, maximiser la répétition d'usage	Expédier en 3 minutes, minimiser les ruptures	Evaluation de la blessure, gestion des ressources, temps, probabilité de succès
Fonctions générales	Transporter une pièce,	Charger, décharger, préparation, traiter, assemblage, inspection, traitement du matériel, et ordonnancement	Ordonnancer les séquences d'actions dans les travaux en cours d'exécution	Traiter le papier	Prendre l'appel, planifier, distribuer, coordonner	Type d'urgence, transport, qualité des soins requis
Processus physiques	Insertions des composantes, transporter les pièces via AGV (Automated Guided Vehicle)	Traitement (couper, faire le grillage) PC (séquencer, distribuer les AGV)	Transport avec AGV, traiter, inspection, assembler, charge de palettes, décharge de palettes	Imprimer, perforer, plier, couper	Enregistrer appel, estimation des besoins, localisation de l'ambulance la plus proche, localisation du renfort, ressource adaptée aux besoins, affecter l'ambulance à l'urgence	Symptômes, vitesse des ambulances, type de soin requis
Formes physiques	Position et état de chaque station de travail, pièces et ressources de transport	Centres de machines, PC	AGV, palettes, machines numériques, cellules d'inspection, robots, pièces à assembler	Encre, planches, papier, cutter, folder	Travaux, disponibilité ambulances, carte de la région, stations	Ambulances, unités de soins

TAB. II.1 – Différentes tentatives de décomposition de domaines intentionnels (ordonnancement et transport)

décider d'utiliser les machines dans un mode inhabituel, de manière à faciliter la production. Si le but du WDA est d'énumérer les actions possibles sur le domaine, le concepteur doit alors inclure ce type d'utilisation des machines non habituelles, cela amène à une décomposition qui n'est plus indépendante par rapport aux événements et aux moyens.

Dans la littérature, nous identifions deux approches opposées pour la décomposition du domaine pour les problèmes d'ordonnancement et de transport. Wong et al. (1998) proposent une tentative de décomposition pour un système d'ordonnancement de transports (planification des ambulances). Des activités humaines sont incluses dans la décomposition et les auteurs arrivent à la conclusion que la méthode n'est pas idéale pour la modélisation de systèmes intentionnels. Hajdukiewicz et al. (1999) répliquent avec une proposition de la méthode WDA appliquée à la logistique militaire et mettent en évidence les avantages de la méthode pour la modélisation de ce type de systèmes.

Ces deux approches apparaissent aussi dans les problèmes d'ordonnancement. Krosner et al. (1989) évoquent les limites de la hiérarchie d'abstraction pour la décomposition du domaine de ce type de problèmes. Les auteurs remarquent les difficultés à trouver une continuité entre les niveaux de la hiérarchie dans les systèmes discrets, contrairement aux contrôle de processus continus où les processus sont les reflets d'une réalité physique guidée par des lois stables. Les auteurs finissent par proposer une sous-division des niveaux de la hiérarchie selon la nature de l'agent d'exécution du processus (automatique ou humain), ce qui rend la décomposition complètement dépendante des acteurs d'exécution. Kinsley et al. (1994) critiquent cette approche et développent une autre tentative de décomposition pour ce type de problèmes. Cependant, les auteurs rencontrent des difficultés à décomposer le domaine. Par exemple, ils ont recours au terme "priorités" (dans les niveaux de fonctions abstraites) dont l'évolution dépend des décisions humaines et non du fonctionnement du système. En effet, lors de la décomposition du domaine, les auteurs relaxent encore une fois la contrainte d'indépendance vis-à-vis des moyens. En conséquence, l'interface conçue à partir de la décomposition perd partiellement la propriété d'être capable d'agir face à l'imprévu, comme c'est le cas dans les décompositions basées sur des agents d'exécution spécifiques.

Dans Higgins (1999, 2001), le problème est abordé d'un autre point de vue. L'auteur considère l'ordonnancement comme le résultat d'un agent (humain ou ordinateur) et non comme le résultat du domaine de contraintes. Par conséquent, la décomposition reste indépendante vis-à-vis des

événements et des moyens. En contrepartie, certaines contraintes nécessaires pour assister l'humain ne sont plus présentes dans l'interface finale, ce qui remet en question la pertinence psychologique de l'interface. Le but de la méthode CWA est de concevoir des interfaces qui permettent une perception facilitée des contraintes du problème. Un domaine de contraintes non complètement accessible dans la décomposition peut impliquer une représentation incomplète ou même incorrecte et cela peut limiter les possibilités d'action de l'opérateur humain.

Pour la modélisation des domaines intentionnels, les deux approches de la littérature proposent donc soit de relâcher la propriété d'indépendance vis-à-vis des moyens, soit de favoriser une représentation incomplète du domaine avec le risque de réduire la pertinence psychologique de l'interface conçue à partir de la décomposition. Nous considérons qu'une représentation incomplète du domaine s'avère insuffisante pour assister l'opérateur humain dans son travail quotidien. De plus, la dépendance vis-à-vis des moyens peut entraîner des inconvénients dans les problèmes de transport. Les opérateurs humains mettent en place des stratégies ou des connaissances spécifiques qui permettent par exemple de relâcher des contraintes. Si ces aspects ne sont pas pris en compte dans la décomposition, celle-ci ne serait plus pertinente d'un point de vue psychologique. Par contre, si ces aspects sont pris en compte, comme dans les méthodes d'analyse de la tâche, la décomposition devient dépendante des événements et des moyens existants.

Nous proposons dans le paragraphe suivant une méthode qui permet d'identifier les actions possibles à partir de l'énumération des contraintes qui peuvent être éventuellement relâchées par l'opérateur humain.

II.3 Approche proposée pour l'analyse du domaine

Dans ce paragraphe, nous réalisons une analyse du domaine pour le problème de l'ordonnement des transports. Pour cela, nous proposons de commencer par l'énumération de toutes les variantes possibles du problème de tournées de véhicules que nous trouvons dans la littérature. Cette démarche permet de lister et d'identifier les caractéristiques (objets physiques, contraintes particulières, objectifs, ...) de chaque variante de façon à les inclure dans la décomposition du domaine. Dans l'interface résultante, les différentes possibilités de procéder seront prises en compte même si elles ne sont pas essentielles de manière constante ou qu'elles n'ont pas été révélées par l'analyse de tâches. Par exemple, dans certaines variantes du problème de tournées de véhicules,

il existe la possibilité de pouvoir louer un véhicule avec un certain coût associé. Si l'opérateur humain se trouve face à un problème où les véhicules disponibles ne sont pas suffisants pour satisfaire la demande des clients avec un délai de temps acceptable, une solution possible est l'affectation d'un véhicule supplémentaire pour servir certains clients. Si cette option n'est pas anticipée par le concepteur, le système d'assistance serait incomplet et inadéquat pour aider réellement l'opérateur pour la résolution du problème.

II.3.1 Hiérarchie d'abstraction

Nous proposons une hiérarchie d'abstraction pour le domaine de travail d'un système d'ordonnancement du transport.

La hiérarchie d'abstraction structure le problème en différents niveaux de raisonnement du point de vue de l'opérateur. Cette hiérarchie a pour but d'identifier les différentes possibilités d'action dans le domaine sans rentrer dans des stratégies particulières. Par exemple, si l'intérêt est de réduire le coût total d'un ensemble de routes, l'opérateur humain peut agir sur la capacité grâce à une diminution du nombre de véhicules à utiliser (avec éventuellement des véhicules surchargés), ou encore en gérant le temps, *e.g.*, l'opérateur peut réduire le temps de route s'il décide de ne pas servir certains clients (qui est une autre possibilité dans certaines variantes du problème de tournées de véhicules).

La hiérarchie d'abstraction permet non seulement d'identifier les possibilités d'action de l'opérateur humain, mais permet également de tracer efficacement la propagation des effets dans le domaine de travail comme conséquence de ces actions. Le but du WDA est alors de fournir une décomposition du problème (identification des limites structurelles du système de travail, identification des contraintes, . . .) qui sera utilisée comme fondement à la conception d'un système d'aide à la décision générique. Dans certains cas, plusieurs représentations des différents niveaux d'abstraction sont possibles. Nous avons décidé de conserver la représentation qui reste la plus générique ou qui favorise la clarté de la décomposition.

Les niveaux de la décomposition

Les niveaux de décomposition ont été introduits dans le paragraphe II.2. Les cinq niveaux s'avèrent nécessaires pour la description du domaine de travail. Les trois niveaux fonctionnels sont bien adaptés à la typologie du problème, par contre le débat est plus ouvert lorsqu'on

considère la pertinence du niveau des processus physiques dans ce type de problèmes. Wong et al. (1998) définissent dans ce niveau des actions à réaliser qui peuvent être considérées comme la stratégie à suivre (identifier l'ambulance la plus proche, vérifier la compatibilité avec l'urgence, ...) pour l'affectation de l'ambulance à l'urgence (voir tableau II.1). Nous avons décidé de définir dans ce niveau les aspects statiques apportés par les objets physiques, plus particulièrement les capacités, les contraintes et les requêtes spécifiées par ces objets. Finalement, dans le niveau des objets physiques, nous considérons les objets, leurs caractéristiques physiques et leur localisation spatiale.

Les objectifs fonctionnels. Le premier niveau d'abstraction décrit d'une part les objectifs du système de travail vis-à-vis de son environnement et d'autre part les contraintes externes imposées par l'environnement. Les objectifs du système de travail peuvent être séparés en objectifs primaires et objectifs secondaires.

Les objectifs primaires définissent les raisons de l'existence du système de travail dans son environnement. L'objectif primaire du système de travail est de transporter des marchandises ou passagers en prenant en compte les demandes des clients. Habituellement, les objectifs secondaires expriment les valeurs des personnes dans le système de travail. Une entreprise de transports cherche à faire des bénéfices ; une manière de contribuer à l'atteinte de ce but est la *minimisation des coûts*. D'autres objectifs comme la sécurité ou le respect des normes législatives peuvent être aussi considérés à ce niveau de la décomposition. Nous pouvons considérer que l'importance de ces objectifs secondaires est limitée dans la décomposition lorsque l'on considère la hiérarchie comme un outil d'aide pour le concepteur. En effet, si la décomposition est pertinente, les contraintes et limites liées à ces objectifs vont de toute façon apparaître dans la décomposition.

Le niveau des objectifs fonctionnels décrit aussi les contraintes externes imposées par l'environnement sur le domaine de travail. Normalement, ces contraintes expriment les valeurs de l'environnement et de la société. Dans le domaine analysé, la contrainte externe est l'exigence d'un niveau minimal de la qualité de service client. En tant qu'entreprise de services, la contrainte externe est fortement liée à l'objectif primaire du système de travail. La qualité de service est satisfaite uniquement lorsque le transport des marchandises ou passagers est réalisé en satisfaisant les requêtes des clients. En effet, si la contrainte externe est satisfaite, l'objectif principal du système de travail sera aussi atteint.

Finalement, nous observons que les différents objectifs de ce niveau peuvent parfois être opposés. Par exemple, un retard dans la livraison d'un client peut entraîner une réduction des coûts par le fait que la distance parcourue est plus courte.

Fonctions abstraites. Ce niveau d'abstraction représente des critères pour mesurer le progrès du système de travail vers les objectifs fonctionnels. Nous avons décidé de définir les concepts dont les critères dérivent, plutôt que de décrire les critères proprement dits. Les avantages de décrire les concepts et non les critères sont que la généralité de l'analyse est favorisée et que le système de support sera prêt à traiter avec des objectifs établis au préalable mais aussi avec des objectifs conflictuels inattendus.

Pour juger si les objectifs fonctionnels sont atteints, nous devons évaluer la *gestion de la capacité* et la *gestion du temps*. Dans notre système de travail les objectifs principaux vont être satisfaits par une gestion efficace de la capacité et du temps disponible. Nous considérons que la gestion de la distance parcourue est également comprise dans la gestion du temps. A titre d'exemple, nous pouvons considérer deux des critères les plus importants pour les problèmes de tournées de véhicules : la minimisation du nombre de véhicules et la minimisation de la distance parcourue par les véhicules. Ces deux critères dérivent respectivement de la gestion de la capacité et de la gestion du temps.

Les concepts appartenant à ce niveau d'abstraction permettent de raisonner à partir des principes de base lorsque les opérateurs sont confrontés à des situations inattendues ou non familières. Par exemple, si un nouveau client demande à être livré alors même que la tournée est déjà planifiée, l'opérateur humain doit alors raisonner en termes de nouvelles requêtes temporelles et de capacité avant de prendre la décision d'introduire le client dans les tournées.

Les deux objectifs fonctionnels sont liés aux concepts de ce niveau. En effet, la minimisation des coûts dépend de l'efficacité de la gestion de la capacité et du temps. Le client est satisfait lorsqu'il est livré dans la période prévue, ce qui implique de gérer simultanément la capacité et le temps disponible pour l'atteinte de cet objectif.

Nous remarquons que les deux concepts peuvent être liés et parfois opposés ; une gestion efficace de la capacité qui minimise le nombre de véhicules peut entraîner des faiblesses dans la gestion du temps disponible (retards dans les livraisons, augmentation de la distance parcourue, ...), et vice versa. Arriver à un compromis est parfois nécessaire afin d'atteindre les objectifs du

premier niveau.

Fonctions liées aux objectifs. Ce niveau d'abstraction décrit les fonctions qu'il faut remplir pour atteindre les objectifs fonctionnels. L'opérateur humain doit coordonner et gérer les fonctions nécessaires pour résoudre le problème et les contraintes qui doivent être satisfaites afin de proposer une solution garantissant une bonne qualité de service tout en minimisant les coûts.

Les principales contraintes du domaine de travail ont été identifiées à partir de l'analyse des différentes variantes réelles et académiques du problème de tournées de véhicules rencontrées dans la littérature (voir le paragraphe I.2). Même si chaque variante présente des particularités, nous pouvons considérer que les fonctions spécifiées à ce niveau couvrent une large partie des problèmes d'ordonnement des transports.

Les fonctions qu'il faut remplir pour arriver à une gestion efficace de la capacité sont : *la satisfaction des contraintes liées au volume à transporter, la satisfaction des contraintes liées au poids à transporter, la sélection des itinéraires à parcourir et la satisfaction des contraintes existantes entre les objets* (contraintes d'incompatibilité, contraintes d'affectation, contraintes de précedence, ...). Ce sont les fonctions qui ont une influence directe sur la gestion de la capacité. Si l'intérêt de l'opérateur humain est d'améliorer un critère qui dérive de la gestion de la capacité, l'opérateur a la possibilité d'agir soit de façon directe sur ces fonctions (par exemple les contraintes peuvent être relâchées et les fonctions ne seront plus remplies) soit de façon indirecte (avec la modification des objets ou des caractéristiques des objets qui ont un rapport avec les fonctions). Les possibilités d'agir qui ont une influence sur la gestion de la capacité sont : (1) agir sur la disponibilité des véhicules de façon à modifier la capacité disponible en volume et en poids ; (2) agir sur la demande des clients de manière à modifier les besoins de capacité en volume et en poids ; et (3) agir sur les contraintes d'affectation qui peuvent avoir un impact sur la gestion de la capacité. Par exemple, si deux clients doivent impérativement être servis par le même véhicule, cette contrainte peut mener à une gestion de la capacité inefficace qui pourrait éventuellement entraîner une augmentation des coûts ; dans ce cas l'opérateur peut décider de ne pas servir un des clients de manière à atteindre l'objectif de minimiser les coûts.

Les fonctions en relation avec la gestion du temps sont : *la sélection des itinéraires à parcourir par les véhicules, satisfaire les temps nécessaires pour servir chaque client* (temps de charge, temps de décharge, temps de réparation, ...), *satisfaire les fenêtres de temps de service des clients*

et de temps de travail des conducteurs et satisfaire les contraintes existantes entre les objets. La sélection des itinéraires est la seule fonction qui ne consiste pas à satisfaire une contrainte du problème. L'opérateur avec le support du système d'aide à la décision doit faire la sélection des itinéraires à suivre pour chaque véhicule. Une sélection d'itinéraires inadéquate peut mener à une gestion du temps inefficace qui pourrait éventuellement entraîner des conséquences importantes sur les objectifs fonctionnels du système de travail. Si l'objectif est d'améliorer les critères qui sont liés à la gestion du temps, les possibilités sont : (1) agir sur les routes de manière à modifier le temps de trajet des véhicules, pour ce faire l'opérateur humain dispose de plusieurs options, comme de forcer les conducteurs à conduire plus vite ou de modifier l'ordre de service des clients ; (2) agir sur le temps nécessaire pour servir les clients en ajoutant par exemple un travailleur supplémentaire dans les véhicules ou en prenant la décision de ne pas servir certains clients ; (3) agir sur les fenêtres de temps de travail des conducteurs, par exemple avec la proposition d'heures supplémentaires ; (4) agir sur les fenêtres de temps des clients, de manière à éviter les retards dans les livraisons ; et finalement (5) agir sur les contraintes d'affectation qui ont une influence sur la gestion du temps.

L'opérateur dispose donc de plusieurs possibilités d'action pour chaque fonction de manière à trouver une solution qui respecte les objectifs du système de travail.

Processus liés aux objets. Ce niveau de la hiérarchie décrit le système de travail à partir des capacités et limitations fonctionnelles des objets physiques. Le but est de définir quelles sont les capacités et les limitations apportées par les objets physiques au système de travail. Ces propriétés sont nécessaires pour arriver à remplir les fonctions liées aux objectifs.

Dans notre système de travail, les objets physiques définissent le *réseau routier*. La position géographique des clients et des dépôts et les caractéristiques des véhicules et des produits (vitesse maximale, produits dangereux, les remorques, ...) définissent le réseau routier du problème. Ce réseau routier a une forte influence dans la détermination des itinéraires à suivre par les véhicules.

Le second concept de ce niveau est la *disponibilité* des objets physiques considérés comme les ressources du problème (les véhicules et les conducteurs). La disponibilité de ces ressources déterminent la capacité de transport disponible (le volume et le poids) mais entraîne aussi des contraintes temporelles comme les fenêtres de temps de travail des conducteurs qui doivent être respectées. Nous avons recours à ce terme pour souligner la possibilité d'utiliser les véhicules et

conducteurs disponibles, mais aussi pour indiquer la capacité disponible en ce qui concerne le volume et le poids.

Les clients et les produits spécifient le troisième concept du niveau : *les demandes* du problème. Les demandes n'entraînent pas seulement des besoins en capacité. Les contraintes temporelles telles que les fenêtres de temps des clients et les temps de service de chaque client doivent aussi être prises en compte. La satisfaction de ces contraintes doit être assurée pour arriver à atteindre un service client satisfaisant. La capacité disponible fournie par les ressources (véhicules et conducteurs) et la capacité requise par le demande sont les concepts que l'opérateur doit prendre en considération pour satisfaire les contraintes de capacité (contraintes de volume et de poids).

Finalement, nous considérons que les attributs de ce niveau (la disponibilité des ressources, le réseau routier spécifié par les objets physiques et les requêtes clients) spécifient un ensemble de contraintes entre les objets physiques qui doivent être satisfaites. Ces contraintes ont été rassemblées sous le nom de contraintes entre les objets physiques. Les contraintes d'affectation, ou inclusion, et de non affectation, ou exclusion, sont prises en compte dans cet ensemble de contraintes. Par exemple, si un véhicule a des difficultés d'accès chez un client, alors une nouvelle contrainte d'exclusion entre le client et le véhicule est considérée.

Objets physiques. Les objets physiques du système de travail sont : les véhicules, les conducteurs, les marchandises, les dépôts et les clients. Dans ce niveau, nous considérons les objets mais aussi les caractéristiques en termes d'appartenance et de localisation spatiale. Ainsi, par exemple dans le cas des véhicules, nous considérons dans ce niveau : la capacité de transport du véhicule (volume, poids et longueur), la vitesse (limite et moyenne), le type (licence, produits, ...), le coût d'utilisation et la localisation géographique du véhicule (affectation à un dépôt).

Les liens entre les niveaux

La nature des liens entre les niveaux de la hiérarchie d'abstraction est de type fins-moyens. Cette relation peut être caractérisée à partir du triplet comment-quoi-pourquoi. Les concepts de chaque niveau spécifient ce qui est considéré. Le rapport du concept avec le niveau inférieur est de spécifier comment le concept peut être implémenté ou réalisé. Ces concepts répondent à la question "comment doit-il être réalisé?". Le rapport du concept avec le niveau supérieur est

d'indiquer les fins ou pourquoi le concept est présent dans l'analyse du domaine de travail. Ces concepts du niveau supérieur répondent à la question "pourquoi doit-il être réalisé?"

Le triplet est clairement respecté dans les niveaux fonctionnels de la décomposition (les objectifs fonctionnels, les valeurs et les mesures de priorité et les fonctions générales). Si nous prenons comme exemple la gestion de la capacité du niveau des fonctions abstraites, ce concept spécifie ce qui doit être réalisé. La relation entretenue avec les objectifs fonctionnels concerne la finalité : *Pourquoi une gestion de la capacité efficace doit-elle être respectée ? Pour atteindre la minimisation des coûts et arriver à offrir un service client satisfaisant.* La relation entretenue avec les fonctions générales relève des moyens employés : *Comment une gestion de la capacité efficace est-elle atteinte ? Avec la satisfaction des contraintes de capacité, avec la proposition d'itinéraires efficaces pour les véhicules et avec la considération des contraintes entre les objets physiques.*

En ce qui concerne les niveaux physiques de la décomposition, le rapport fins-moyens entre les concepts est moins évident. Ceci est dû au fait que les concepts du niveau des processus physiques ne sont pas vraiment des processus. Les termes de la décomposition dans ce niveau (la disponibilité des ressources et de la capacité, le réseau routier spécifié par les positions géographiques et les caractéristiques des objets et les requêtes et demandes des clients) représentent les contributions des objets physiques plutôt à travers des concepts statiques (capacité, routes, demandes) qu'à travers de processus. Cependant, le rapport fins-moyens est aussi respecté. Par exemple, pour le concept de la disponibilité du niveau de processus physiques, la nature de la relation entre la disponibilité et le niveau des fonctions générales est de finalité : *Pourquoi les ressources disponibles (véhicules et conducteurs) et la capacité disponible (volume et poids) sont-ils considérés dans le système de travail ? Les ressources sont considérées pour la sélection des itinéraires et pour satisfaire les contraintes spécifiées par les limitations introduites par les véhicules et les conducteurs. La disponibilité de capacité est considérée pour arriver à satisfaire les contraintes de capacité.* Inversement, la nature de la relation entre la satisfaction de la contrainte liée au volume et le niveau des processus physiques est d'identifier les moyens : *Comment la satisfaction du volume est-elle atteinte ? En respectant que le volume fourni par les ressources disponibles est suffisant pour satisfaire le volume requis par les demandes des clients.*

Une hiérarchie d'abstraction pour l'ordonnement des transports

La figure II.1 montre la hiérarchie d'abstraction proposée pour le problème d'ordonnement des transports.

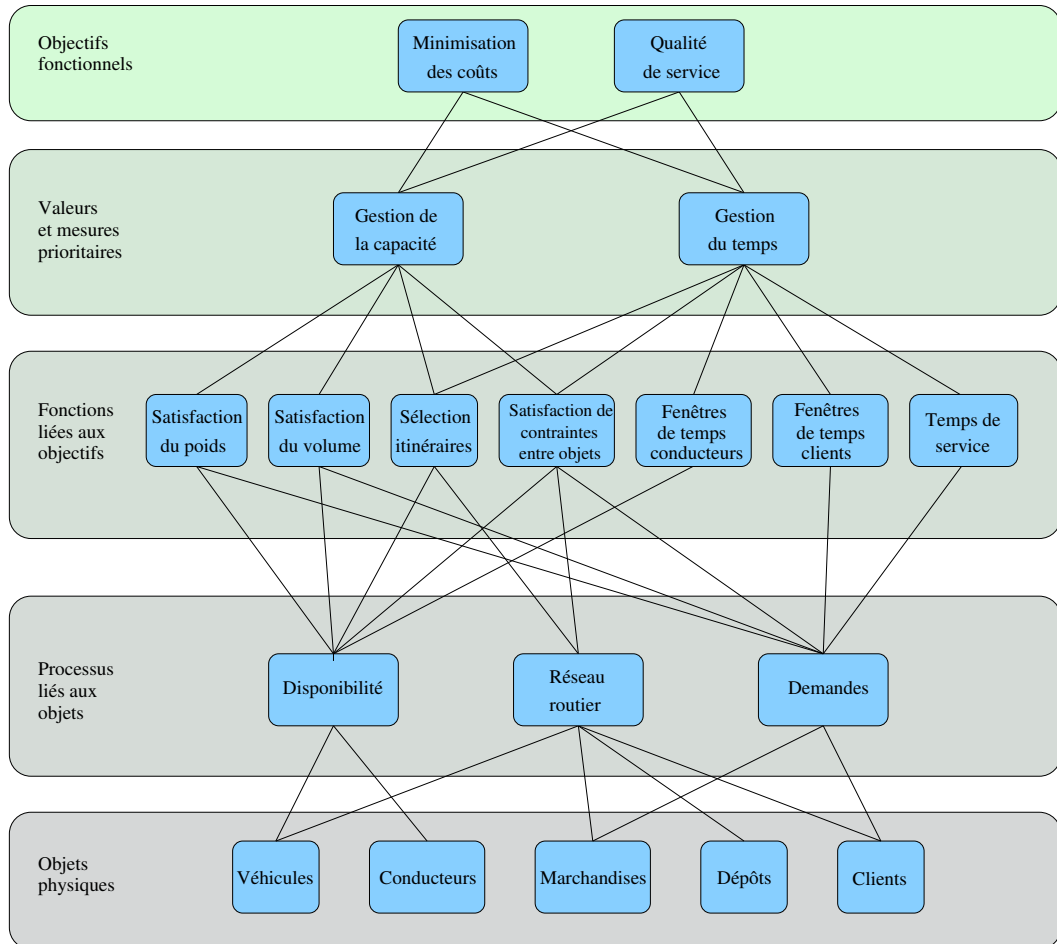


FIG. II.1 – Analyse du domaine de travail à travers une hiérarchie d'abstraction pour le problème d'ordonnement des transports

II.3.2 Analyse du domaine de validité de la décomposition

Une fois la hiérarchie d'abstraction proposée, nous allons passer en revue les différentes variantes du problème de tournées de véhicules de manière à déterminer le degré de généralité de la décomposition. L'objectif est d'identifier les variantes du problème de tournées de véhicules

qui sont couvertes par la décomposition et celles qui ne le sont pas. Pour cela, nous utilisons le schéma de classification pour les problèmes de tournées de véhicules proposé par Desrochers et al. (1990) et les extensions déjà introduites dans le paragraphe I.2.3. Desrochers et al. (1990) proposent quatre grands champs pour la classification des problèmes. Le premier champ spécifie les caractéristiques et les contraintes concernant les clients et les dépôts. Le deuxième champ spécifie les caractéristiques des véhicules. Le troisième champ spécifie d'autres caractéristiques du problème qui ne concernent ni les clients, ni les dépôts, ni les véhicules. Finalement, le dernier champ concerne la fonction objectif du problème.

Dans les paragraphes qui suivent, nous allons énumérer les options pour chaque champ de la classification, de manière à identifier quelles sont les extensions du problème qui sont représentées par la hiérarchie d'abstraction proposée.

Caractéristiques des clients et des dépôts

Dans le schéma de classification, ce champ est divisé en quatre sous-champs. Le premier de ces sous-champs spécifie le nombre de dépôts du problème. Il existe des problèmes avec un seul dépôt et des problèmes avec de multiples dépôts. Ces derniers correspondent à la variante MDVRP présentée dans le paragraphe I.2.3. La hiérarchie d'abstraction proposée reste adéquate pour les problèmes avec un unique ou plusieurs dépôts.

Le deuxième champ spécifie les caractéristiques de la demande. Ce champ définit trois caractéristiques différentes. D'abord, la première caractéristique spécifie la modélisation des clients dans le graphe. Les clients peuvent être modélisés par des sommets, par des arcs (problème connu sous le nom de CARP) ou peuvent être aussi modélisés par un couple origine-destination de deux sommets (ce qui peut être vu comme un problème du type VRPPD). La décomposition proposée ne limite pas la modélisation des clients dans les graphes. Dans la hiérarchie d'abstraction, le réseau routier définit la position géographique des clients et des dépôts et les axes routiers existants entre ces localisations. En aucun cas, la modélisation choisie pour ces objets physiques n'est définie dans la hiérarchie d'abstraction.

La seconde caractéristique spécifie la nature de la demande. Les demandes peuvent toutes être de la même classe (livraisons ou collectes) ou avoir des demandes mixtes entre livraisons et collectes. Nous pouvons rapprocher ce dernier cas des variantes VRPB (Vehicle Routing Problem with Backhauls) ou VRPPD (Vehicle Routing Problem with Pick-up and Deliveries) présentées

dans le paragraphe I.2.3. La troisième caractéristique spécifie si les demandes sont déterministes ou stochastiques. La modélisation des demandes avec des variables aléatoires correspond à la variante connue sous le nom de VRPSD (Vehicle Routing Problem with Stochastic Demand). Nous observons que la décomposition proposée est pertinente pour toute classe de demandes (livraisons, collectes ou mixtes). Indépendamment du type de demande, les contraintes liées à la limite de capacité (volume et poids) doivent être satisfaites. D'un autre côté, la possibilité de considérer des demandes stochastiques n'est pas reflétée dans la décomposition. Toutefois, le but est de concevoir un système d'aide à la décision qui soit capable d'assister l'opérateur même dans les situations qui n'ont pas été prévues par avance. Pour cette raison, nous pensons qu'il n'est pas nécessaire d'introduire une modélisation avec des variables aléatoires pour représenter l'incertitude. En effet, si la conception du système d'aide à la décision est pertinente, le système va fournir à l'opérateur humain les outils pour arriver à gérer cette incertitude.

Le troisième sous-champ spécifie les contraintes temporelles des clients et des dépôts. Ces contraintes sont modélisées par des fenêtres de temps possibles pour l'horaire de passage chez les clients. Cette extension du problème est connue sous le nom de VRPTW ou VRPMTW lorsqu'une ou plusieurs fenêtres de temps sont disponibles pour chaque client. Ces différentes variantes sont aussi prises en compte dans la hiérarchie d'abstraction. Les clients imposent ces contraintes temporelles et la satisfaction de ces fenêtres de temps fait partie des fonctions nécessaires à remplir dans la hiérarchie d'abstraction de manière à atteindre une bonne qualité dans le service client.

La dernière caractéristique spécifie les contraintes de sélection de clients. Différentes variantes du problème peuvent être considérées : une unique planification doit être réalisée ou plusieurs planifications doivent être faites sur une période de temps (comme dans le Periodic Vehicle Routing Problem et dans l'Inventory Routing Problem). Cette dernière variante du problème avec planifications multiples de tournées sur un horizon de temps n'est pas incluse dans la hiérarchie proposée. Les concepts de périodicité ou de fréquence de service des clients n'apparaissent pas dans la décomposition. L'analyse du domaine réalisée ne représente donc pas les limites structurelles du problème lorsque l'opérateur est confronté à un problème de tournées de véhicules périodique. Cependant, ce type de problèmes est habituellement décomposé en deux sous-problèmes pour être résolu. Dans un premier temps, un problème d'affectation des clients pour chaque période de temps est résolu, puis les tournées de véhicules journalières sont définies.

Dans ce cas, la décomposition proposée reste complètement adéquate pour le second problème mais reste incomplète pour résoudre le problème d'affectation.

Caractéristiques des véhicules

Cinq caractéristiques liées aux véhicules sont considérées pour la classification des variantes du problème de tournées de véhicules. La première caractéristique spécifie le nombre de véhicules du problème, c'est-à-dire, s'il existe un nombre de véhicules disponibles maximal et si tous les véhicules disponibles doivent être impérativement utilisés ou non. La deuxième caractéristique spécifie le type de flotte. La flotte peut être homogène ou hétérogène (Heterogeneous Fleet Vehicle Routing Problem ou HVRP). La troisième caractéristique liée aux véhicules spécifie si ceux-ci disposent ou non de compartiments individualisés pour chaque client (ce type de contraintes est considéré dans le distance-constrained vehicle routing problem). L'analyse du domaine de travail proposée est indépendante du nombre de véhicules et de leurs caractéristiques. Ces caractéristiques sont prises en compte dans la décomposition, car elles définissent aussi le réseau routier et spécifient les contraintes de capacité qui doivent être satisfaites.

Finalement, les deux dernières caractéristiques spécifient des contraintes temporelles. Le problème peut présenter des intervalles de disponibilité des véhicules et des bornes inférieures et supérieures pour la distance ou pour la durée des tournées. Ces contraintes doivent être considérées lors de la sélection des itinéraires. La sélection des itinéraires consiste à proposer une séquence de clients pour chaque tournée. Pour arriver à une proposition de routes efficace, les contraintes comme les périodes de disponibilité des véhicules, les fenêtres de temps des conducteurs et les bornes pour les caractéristiques des tournées doivent être respectés.

Caractéristiques du problème

Ces caractéristiques du problème sont les propriétés du réseau routier, les stratégies de service et les contraintes dues aux liens entre les dépôts, les clients et les véhicules.

Les propriétés du réseau (réseau orienté, non-orienté ou mixte) sont considérées dans la définition du réseau routier qui est défini à partir des localisations des clients et des dépôts, mais aussi à partir des caractéristiques des véhicules et des produits transportés.

Les différentes stratégies de service doivent être choisies par l'opérateur humain. Dans certains cas, l'opérateur humain peut préférer permettre la division des demandes des clients. Dans

d'autres cas, lorsque les demandes sont indifféremment des livraisons ou des collectes, l'utilisateur peut privilégier la stratégie de livrer d'abord pour collecter après. Cette stratégie correspond à la variante VRPB (Vehicle Routing Problem with Backhauls). L'opérateur humain peut aussi permettre plus d'une tournée par véhicule ou autoriser des tournées avec un dépôt de fin de trajet différent du dépôt de départ. La décomposition reste indépendante de la stratégie suivie par l'opérateur pour la résolution du problème. Pour cette raison, les stratégies ne sont pas identifiées dans la hiérarchie d'abstraction. De toutes façons, les tournées doivent respecter les contraintes du problème de manière à atteindre les objectifs. Les stratégies de l'opérateur pour la résolution du problème doivent donc être considérées lors de la phase de conception du système d'aide à la décision pour arriver à proposer des outils qui permettent d'aider l'opérateur lorsque celui-ci applique sa stratégie particulière pour la résolution du problème.

Finalement, les liens entre les véhicules, les clients et les dépôts sont spécifiés. La première de ces relations est la contrainte de précédence entre deux clients (le véhicule doit servir un client avant de servir l'autre). La plupart des liens sont des restrictions d'inclusion et d'exclusion. Les auteurs identifient les possibles restrictions entre objets : les restrictions dépôt-client, les restrictions client-client, les restrictions dépôt-véhicule et les restrictions client-véhicule. Dans la hiérarchie d'abstraction, toutes ces restrictions sont considérées avec le concept de contraintes des objets physiques. Ces contraintes doivent être satisfaites pour arriver à atteindre les objectifs. En plus de ces restrictions, les contraintes entre les conducteurs et les autres objets physiques (clients, dépôts et véhicules) sont aussi considérées.

Dans la classification, les auteurs considèrent la synchronisation entre véhicules. Cette requête est nécessaire par exemple lorsque les véhicules doivent échanger les charges. Ce type de restriction n'est pas spécifié dans la hiérarchie d'abstraction. En effet, nous pensons que d'autres objets physiques et de nouvelles contraintes doivent être ajoutés à la hiérarchie d'abstraction pour arriver à décrire le domaine de travail de ce type de problèmes.

Objectifs

Les fonctions objectif sont spécifiées à partir de cinq quantités. Ces quantités représentent la durée des tournées (temps de trajet et temps de service), le coût des véhicules et un coût de pénalité liée à la violation de contraintes des véhicules (contraintes de capacité, contraintes de disponibilité et contraintes sur les tournées), le coût des clients liés à la déviation par rapport à

un service préférentiel et un coût de pénalisation pour la violation de contraintes liées aux clients (fenêtres de temps).

Les quantités proposées pour définir les fonctions objectif sont fortement liées à la gestion du temps et la gestion de la capacité. Les fonctions générales de la hiérarchie d'abstraction ont un fort impact sur tous ces critères de mesure de la qualité d'une solution ; d'une part la considération des coûts et de l'autre la considération d'une pénalisation lorsque les contraintes du problème sont violées. Le fait de définir des concepts généraux dans le niveaux des priorités permet de faire en sorte que la décomposition reste adéquate pour la plupart des critères considérés habituellement dans les problèmes de transport.

Récapitulatif

Dans les paragraphes précédents, les caractéristiques des différentes variantes du problème de tournées de véhicules ont été analysées de manière à déterminer si la hiérarchie d'abstraction proposée dans le paragraphe II.3.1 reste pertinente pour chaque variante. De cette analyse, nous pouvons conclure que la décomposition est adéquate pour la quasi-totalité des variantes. Il existe deux types de problèmes dont les caractéristiques des domaines ne sont pas complètement identifiées dans la décomposition proposée. Ces deux types de problèmes sont, d'une part, les variantes qui impliquent une planification sur un horizon de temps, c'est-à-dire le problème de tournées de véhicules périodique (PVRP) et le problème de tournées de véhicules combinées à la gestion de stocks (IRP), et d'autre part, les problèmes avec des localisations et contraintes particulières (points d'échange de charge, synchronisation des véhicules, ...). Pour ce dernier type de problème, de nouveaux objets physiques et de nouvelles contraintes doivent éventuellement être ajoutés à la hiérarchie d'abstraction proposée.

II.4 Conclusion

Dans ce chapitre, nous avons introduit l'analyse du domaine de travail (WDA) comme outil de support pour la conception d'un système d'aide à la décision pour l'ordonnancement des transports. Cette analyse permet d'identifier les possibilités d'action pour le contrôle du domaine de travail dans des situations familières, mais aussi dans des situations nouvelles. Nous avons aussi présenté la hiérarchie d'abstraction comme outil principal pour la matérialisation de cette

analyse. Ensuite, nous avons évoqué les problèmes existants pour la décomposition de systèmes intentionnels (systèmes où l'humain se trouve au cœur du domaine) en se basant sur plusieurs tentatives de décomposition de la littérature.

Finalement, nous avons proposé une analyse générique du domaine pour le problème de l'ordonnancement des transports au travers d'une hiérarchie d'abstraction. Cette analyse a été réalisée à partir de l'identification des différentes variantes des problèmes de tournées de véhicules rencontrées dans la littérature. L'analyse qui en résulte reste indépendante vis-à-vis des événements et des moyens. De plus, nous considérons que les interfaces homme-machine proposées à partir de cette hiérarchie d'abstraction resteront pertinentes du point de vue psychologique, car l'analyse est assez exhaustive et permet non seulement d'identifier les possibilités d'action pour chaque situation spécifique mais aussi de visualiser comment sont propagées les conséquences possibles des actions sur les autres concepts du système de travail. Ces propriétés vont être particulièrement exploitées lors de la conception du système d'aide à la décision. La partie du système d'aide à la décision dédiée à la modélisation du problème est structurée à partir des objets physiques, des caractéristiques de ces objets et des contraintes identifiées dans la hiérarchie d'abstraction. Cette structure du domaine et les liens entre les différents niveaux ont été aussi exploités dans la conception de la partie du système d'aide à la décision dédiée à l'assistance de l'opérateur lorsque le problème n'est pas réalisable (voir chapitre V).

Chapitre III

Architecture décisionnelle pour l'ordonnancement des transports

III.1 Introduction

Nous présentons dans ce chapitre l'architecture générique d'un système d'aide à la décision pour l'ordonnancement des transports. L'analyse du domaine de travail réalisée dans le chapitre II est le centre conceptuel du système d'aide à la décision. Le mécanisme de résolution et les interfaces d'interaction homme-machine ont été conçus en se fondant sur cette analyse qui nous a permis d'identifier les objets, les contraintes et les objectifs du système. Après la présentation de l'architecture du système, nous abordons l'analyse des principales tâches à réaliser pour résoudre le problème. L'objectif de l'analyse est de déterminer la meilleure répartition des fonctions entre l'homme et la machine. Nous nous sommes servis de cette analyse pour développer des interfaces pour la coopération homme-machine ainsi que des algorithmes dédiés, afin d'aider l'opérateur humain dans la réalisation de chacune des tâches. Dans ce chapitre, nous décrivons le mécanisme de résolution et les algorithmes associés. Les interfaces sont elles présentées dans le chapitre IV. Un des algorithmes a été conçu pour résoudre le problème dans sa globalité. Nous avons testé cet algorithme sur des instances de la littérature. Les résultats obtenus sont analysés dans la dernière partie du chapitre.

III.2 Synoptique de l'architecture

Nous proposons une architecture générique pour un système d'aide à la résolution de problèmes de tournées de véhicules. Nous adoptons pour cela une démarche interdisciplinaire faisant converger deux composantes : (1) une interface humaine ou écologique basée sur une hiérarchie d'abstraction issue d'une analyse du domaine de travail ; (2) des mécanismes de résolution basés sur des techniques de la recherche opérationnelle, notamment la programmation par contraintes (cf. figure III.1).

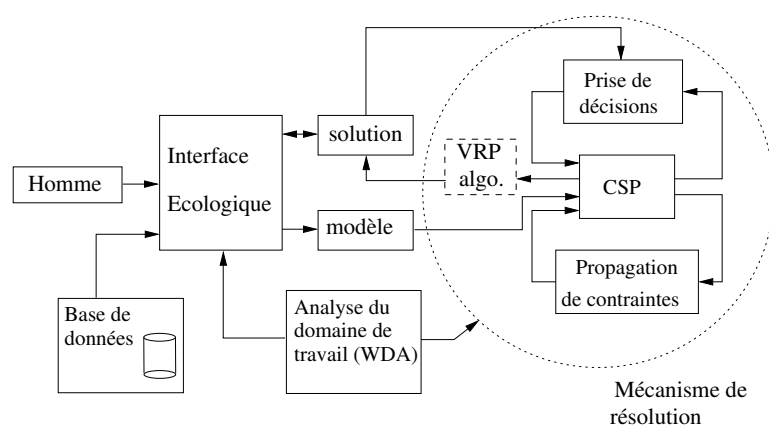


FIG. III.1 – Architecture du système d'aide à la décision

L'idée est de pallier certaines des limitations des systèmes d'aide à la décision actuels annoncées dans le chapitre I. D'une part, un système avec une telle architecture doit être capable de s'adapter à la dynamique du domaine et aux changements rapides des situations. L'analyse du domaine de travail a permis une identification exhaustive des contraintes et des objets du système de travail, le système d'aide à la décision conçu à partir de cette analyse doit ainsi être capable de proposer une aide à l'opérateur dans toutes les situations, y compris les nouvelles ou inattendues (considération d'un nouveau type de contrainte, ajout d'un nouveau dépôt, ...).

D'autre part, les facteurs humains sont considérés lors des phases de modélisation et de résolution du problème. L'architecture proposée permet à l'opérateur de participer activement à la modélisation et à la résolution du problème à travers l'interface écologique (voir chapitre IV). Dans cette architecture, l'opérateur humain se situe au centre du processus de prise de décisions. Toutefois, le mécanisme de résolution intègre des algorithmes spécifiques pour la résolution du

problème de tournées de véhicules qui peuvent aussi participer efficacement au processus de prise de décisions lors de la phase de résolution du problème.

III.3 Analyse des tâches pour la résolution du problème

III.3.1 Identification des tâches pour la résolution du problème

Dans ce paragraphe, l'idée est d'identifier les tâches qui doivent être réalisées lors de la résolution du problème. Ces tâches vont servir comme point de départ pour la conception de la structure (les sous-parties du système, les interfaces nécessaires, ...) du système d'aide à la décision. Nous nous sommes appuyé sur des entretiens avec deux opérateurs issus d'entreprises de transport sécurisé pour l'identification des tâches. Ces deux entreprises doivent chaque jour planifier les tournées à réaliser le lendemain. L'objectif principal des opérateurs est de satisfaire les demandes des clients et d'assurer la sécurité des transports. Bien que la minimisation des coûts de transport soit un objectif secondaire, la planification des tournées reste cependant très importante de manière à pouvoir assurer les contrats avec les clients.

Modélisation du problème

La tâche identifiée comme la première étape à accomplir pour résoudre le problème est la modélisation du problème. Dans le cas des entreprises de transport, cette modélisation consiste principalement à identifier les clients qui ont sollicité un service (localisation géographique des clients), à spécifier les ressources disponibles (véhicules et conducteurs) et, enfin, à définir les contraintes du problème (temps de service chez le client, contraintes temporelles des clients, contraintes de durée pour les tournées, ...). Une des remarques des planificateurs à ce sujet est que les outils disponibles ne permettent pas de prendre en compte certaines contraintes jugées importantes pour la résolution du problème.

Il est alors nécessaire d'inclure dans le système d'aide à la décision une interface homme-machine qui facilite cette modélisation du problème. Dans cette sous-partie du système, l'opérateur humain doit ainsi être capable de définir les objets physiques du problème, les caractéristiques de ces objets qui ont une importance dans la résolution du problème et les contraintes du problème. Les objets physiques et leurs caractéristiques ont été identifiés dans le dernier niveau de la hiérarchie d'abstraction présentée dans le paragraphe II.3.1. Les contraintes imposées

pour chaque objet physique et les contraintes générales du problème (contraintes sur les caractéristiques des tournées et contraintes entre les objets physiques) ont été identifiées de manière exhaustive dans le niveau des fonctions générales (troisième niveau de la hiérarchie d'abstraction du paragraphe II.3.1).

Résolution du problème

Une fois le problème modélisé, la tâche suivante consiste à proposer une solution au problème. Les procédures des deux planificateurs pour trouver une solution satisfaisante sont très similaires. Nous allons décrire ces procédures de manière à identifier des sous-tâches principales pour la résolution du problème. Cette identification de sous-tâches permet de séparer le mécanisme de résolution en différentes parties. Le but n'est pas d'imposer une stratégie particulière à l'opérateur, mais d'arriver à proposer des outils de support grâce à des algorithmes de résolution adaptés à chaque étape de la résolution (sans forcer l'opérateur humain à réaliser obligatoirement les sous-tâches de manière indépendante pour résoudre le problème).

Dans les deux entreprises, les clients sont groupés par zones géographiques. Chaque groupe de clients est servi par un des véhicules. La première des entreprises divise les clients en un nombre fixe de groupes (chaque jour le nombre de tournées est le même). La deuxième entreprise divise le nombre de clients entre le nombre de véhicules disponibles. Nous remarquons rapidement que les deux entreprises ont constamment besoin d'un réaménagement des tournées (diviser une tournée en deux différentes, joindre deux tournées en une, transférer certains clients d'une tournée à une autre, ...) de manière à ce que la charge de travail des tournées soit équilibrée et que la durée maximale des tournées ne dépasse pas la durée de la journée de travail. Ce réaménagement est fait manuellement dans les deux entreprises. L'efficacité du réaménagement dépend donc en grande partie de l'expérience acquise par le planificateur au cours du temps. Nous pouvons donc définir deux sous-tâches qui peuvent être importantes pour la résolution du problème ; la première est la sélection des véhicules à utiliser pour les tournées et la deuxième est l'affectation des clients aux véhicules. La présence de cette dernière tâche est motivée par le fait que dans certains problèmes réels les planificateurs n'arrivent pas à déterminer complètement les itinéraires à suivre par les véhicules. Les planificateurs se contentent souvent de définir les clients qui doivent être servis par chaque véhicule et le conducteur décide en cours de route l'itinéraire à suivre. Nous pensons qu'il est donc pertinent que le système d'aide à la décision dispose d'algorithmes et d'outils

d'interaction homme-machine spécifiques pour aider l'opérateur dans la réalisation de ces deux tâches.

Une fois les clients définitivement affectés aux différentes tournées, les conducteurs d'une des entreprises disposent d'une certaine liberté pour le choix des itinéraires à suivre (ordre des clients à servir). Les conducteurs décident de l'ordre des clients à servir en se basant sur leur expérience. L'autre entreprise utilise un algorithme qui vise à minimiser la distance parcourue. L'algorithme parvient à proposer une solution de bonne qualité pour les itinéraires à suivre pour chaque véhicule. Un des inconvénients annoncés par cette dernière entreprise est que les conducteurs ont tendance à se méfier des itinéraires conçus. Les conducteurs mettent souvent en cause les résultats de l'algorithme en se basant sur leur expérience et sur des connaissances qui ne peuvent pas être considérées par l'algorithme d'optimisation (problèmes de congestion des routes par exemple). Ceci nous fait réfléchir à l'importance d'un outil de support au planificateur et/ou aux conducteurs pour la conception des itinéraires. Cet outil doit inclure des algorithmes d'optimisation pertinents mais aussi des outils d'interaction qui permettent à l'opérateur humain d'exploiter son expertise et ses connaissances, en particulier dans le cas où ces connaissances ne peuvent pas être prises en compte par l'algorithme d'optimisation.

Nous avons décrit les procédures suivies par les deux entreprises. Nous pouvons constater que ces deux entreprises ne disposent pas des outils nécessaires pour une conception efficace de bonnes solutions. Cependant, ce manque est couvert par des planificateurs experts capables de faire face aux problèmes qui se présentent à eux (phénomènes saisonniers, problèmes de personnel, ...) et de proposer ainsi des solutions satisfaisantes permettant de remplir les contrats avec les clients.

Sélection d'une solution

Une des principales difficultés rencontrées par les planificateurs est le manque d'outils de support pour arriver à concevoir des solutions qui respectent certaines contraintes (contraintes secondaires ou molles, préférences) que les planificateurs considèrent souvent comme très importantes. Si le système d'aide à la décision propose à l'opérateur des solutions de piètre qualité en ce qui concerne les critères de performance identifiés dans la hiérarchie d'abstraction, l'opérateur perd confiance envers le système et finit par favoriser une approche pragmatique (manuelle) lui permettant de prendre en compte les critères secondaires (Moray et al., 1995). Le système doit donc permettre à l'opérateur de choisir des solutions performantes pour l'ensemble des critères

considérés comme importants pour l'opérateur.

Ces critères ou contraintes peuvent être considérés en permettant au planificateur de participer à la prise de décisions lors de la construction d'une solution. Nous proposons une autre façon de prendre en compte ces critères. L'idée est de proposer un outil de sélection de solutions qui permette à l'opérateur humain de comparer les solutions par rapport à ces critères secondaires. L'avantage principal de concevoir un outil de support à l'opérateur humain pour la sélection de la solution finale est que les critères et/ou les contraintes qui n'ont été pas prévus lors de la construction des solutions peuvent toujours être considérés dans le processus de sélection de cette solution.

Les interfaces classiques des problèmes de transport affichent les solutions possibles une par une. Ce type d'interfaces ne favorisent pas l'analyse et la comparaison de solutions. En effet, ces interfaces augmentent logiquement la charge mentale due à la nécessité de mémoriser de grandes quantités d'information (surtout pour les problèmes multicritères comme c'est généralement le cas dans les problèmes réels d'ordonnement des transports). Cette augmentation de la charge mentale est supposée accroître le risque de prise de mauvaises décisions. Mietus (1994) remarque que les planificateurs ont tendance à focaliser sur une solution lorsque des choix multiples leur sont proposés. Dans la même logique, Cegarra and Hoc (2008) démontrent que les planificateurs préfèrent parfois des solutions de moins bonne qualité à celle qu'ils pouvaient possiblement atteindre parce que la charge mentale requise est trop élevée.

D'après nous, il est donc nécessaire de mettre à disposition de l'opérateur humain de nouveaux outils pour faciliter cette tâche de comparaison des solutions, afin d'arriver à sélectionner la solution la mieux adaptée au problème réel.

Poursuite des tournées en temps réel

Une autre tâche importante pour parvenir aux objectifs du système de travail est la poursuite en temps réel des tournées. Cette poursuite consiste à suivre à tout instant l'état des tournées et à appliquer les mesures nécessaires en cas de contretemps ou d'incidents. Cette tâche est d'une importance extrême dans les deux entreprises interviewées. Nous avons remarqué que la gestion en temps réel est la tâche qui demande le plus d'expérience et de savoir-faire de la part des planificateurs.

Les avancées technologiques dans les systèmes actuels de géolocalisation permettent à tout

moment de disposer de la position géographique des véhicules. Toutefois, les planificateurs ne disposent pas des outils de support pour anticiper les conséquences des incidents et réagir en conséquence. D'après nous, le système d'aide à la décision pour l'ordonnancement des transports doit disposer des outils nécessaires pour la gestion en temps réel des tournées. L'accroissement de la performance des systèmes embarqués actuels et des techniques de communication permettent une gestion en temps réel efficace basée sur des algorithmes de calcul de faible complexité (heuristiques) ou avec un temps de calcul limité (métaheuristiques, recherche locale, ...).

Schéma des tâches à réaliser pour la résolution du problème

La figure III.2 décrit les principales tâches à réaliser pour la résolution du problème, de manière à satisfaire les objectifs du système de travail.

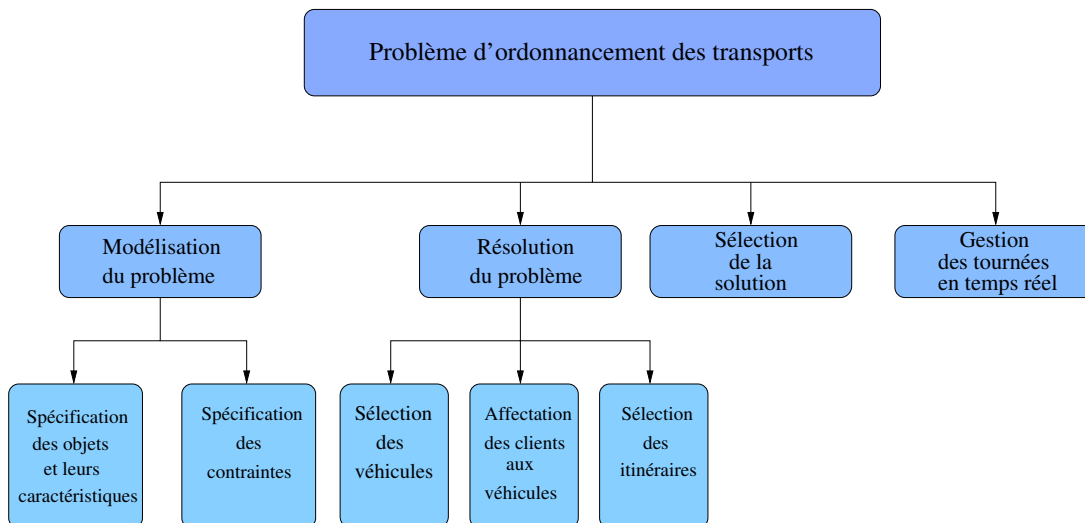


FIG. III.2 – Schéma des tâches à réaliser pour la résolution du problème

III.3.2 Répartition des fonctions entre l'homme et la machine

Nous proposons dans ce paragraphe l'affectation entre l'opérateur humain et le système des différentes fonctions à réaliser pour la résolution du problème. Dans la littérature sur la conception de systèmes, il existe un manque de propositions de méthodes et d'outils pour arriver à déterminer la meilleure affectation des fonctions entre l'opérateur humain et le système. En effet, une des critiques faites au CWA par Chauvin and Hoc (sous presse, 2010) est que le CWA

aborde les questions stratégiques (quatrième étape du CWA) sans avant avoir traité au préalable la répartition des fonctions entre les agents. Ce sujet a été déjà traité pour des situations dynamiques (*e.g.*, Hoc (sous presse, 2010)). Dans van Wezel et al. (2010), les auteurs proposent une procédure fondée sur l'analyse des tâches à réaliser et sur les capacités de l'opérateur et des algorithmes disponibles pour résoudre ces tâches de manière à proposer la meilleure affectation des fonctions entre l'opérateur humain et le système. La procédure est spécifique pour les systèmes d'ordonnement et de planification. Dans l'article, la méthode est appliquée de manière illustrative pour un exemple de système d'aide à la décision pour un problème d'ordonnement de trains dans les voies ferroviaires. Plus récemment, Dimopoulos and Ioannides (sous presse, 2010) se servent de la méthode pour évaluer la pertinence d'un système déjà existant d'aide à la décision pour un problème d'ordonnement d'atelier.

Nous nous basons sur les principes de la méthode pour déterminer le meilleur mode de contrôle pour chacune des tâches. Cinq modes de contrôle sont définis dans l'article (voir figure III.3). Du mode le plus manuel au mode le plus automatique, ces modes sont : manuel, consultatif, interactif ou dynamique, de supervision et automatique.

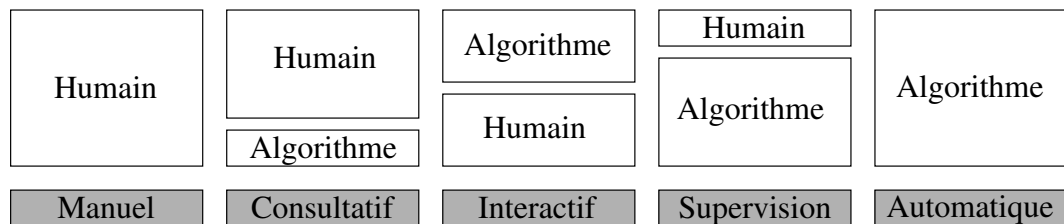


FIG. III.3 – Différents modes de contrôle proposés par van Wezel et al. (2010)

Modélisation du problème

La première des tâches identifiées est la modélisation du problème. Cette tâche est très importante pour la résolution du problème, une mauvaise modélisation du problème peut impliquer que les solutions finales ne soient pas réalisables. Il est alors nécessaire d'avoir conscience des caractéristiques et des particularités du problème pour la réalisation de cette tâche. La modélisation n'est pas une tâche de grande complexité, mais une forte connaissance de l'environnement est cependant nécessaire. Bien qu'une partie de cette modélisation puisse être, dans certaines situations, partiellement automatisée, par exemple avec un chargement du problème à partir d'un

fichier externe qui contient l'information sur les caractéristiques physiques du problème (localisation des clients, caractéristiques des véhicules, ...), la participation de l'opérateur humain à cette étape de modélisation est absolument nécessaire pour une résolution efficace du problème.

Résolution du problème

La deuxième des tâches est la résolution du problème. Cette tâche a été divisée en trois sous-tâches (la sélection des véhicules, l'affectation des clients aux véhicules et la sélection des itinéraires) de manière à pouvoir intégrer dans le système d'aide à la décision des interfaces et des algorithmes dédiés pour chacune d'elles. Ces trois sous-tâches ont un effet très important sur la qualité de la solution finale. En même temps, ce sont des tâches d'une complexité importante et qui peuvent aussi exiger une grande charge mentale pour leur résolution, notamment pour les problèmes de grande taille ou pour les problèmes avec un nombre important de contraintes à prendre en compte. Dans ce contexte, il semble très pertinent de donner un rôle important aux algorithmes de résolution pour la réalisation de ces tâches. Cependant, il n'est pas non plus très souhaitable d'écartier complètement l'humain de ce processus de résolution. Le fait d'introduire l'opérateur humain dans ce processus entraînerait, en premier lieu, la proposition de solutions plus robustes, car il existe toujours des contraintes ou des limites dont l'humain a connaissance et qui ne peuvent pas être considérées par les algorithmes de résolution. En second lieu, cela faciliterait le processus de modification ou de réaménagement des solutions, ce qui est souvent nécessaire pour satisfaire les critères d'équilibrage de tournées ou dans le cas de contretemps lors de la phase de poursuite en temps réel des tournées.

Nous avons décidé de maintenir trois modes de contrôle différents (consultatif, interactif et de supervision) pour la réalisation de ces sous-tâches. Dans le mode consultatif, l'opérateur humain prend les décisions pour la construction d'une solution et le mécanisme de résolution s'occupe de vérifier si les décisions sont réalisables. Dans le mode de supervision, le système propose une solution, cette solution doit être validée par l'opérateur humain pour pouvoir continuer avec la résolution du problème. L'opérateur humain dispose aussi des outils d'interaction nécessaire pour modifier la solution proposée par le système. Finalement, dans le mode interactif, l'opérateur peut proposer une solution partielle qui est, dans un deuxième temps, complétée par le système. Ce mode est très pertinent pour un système d'aide à la décision car l'opérateur peut imposer certaines préférences qui, autrement, ne seraient pas prises en compte par l'algorithme de résolution.

Sélection de la solution finale

Cette tâche est très importante pour arriver à satisfaire les objectifs du système de travail. Les algorithmes issus de la recherche opérationnelle peuvent facilement calculer un grand nombre de solutions réalisables pour ce type de problèmes, sauf pour le cas des problèmes sur-contraints, et peuvent aussi déterminer de manière efficace la meilleure ou les meilleures solutions en regard d'un certain critère d'optimisation. Les problèmes réels d'ordonnancement de transports sont cependant multicritères et, dans la plupart des cas, ces critères ne sont pas entièrement établis au préalable. Dans ce type de problèmes, les opérateurs humains cherchent une solution réalisant le meilleur compromis entre les critères. Jüngen and Kowalczyk (1995) notent que dans certains problèmes, l'humain peut arriver à rejeter une bonne solution réalisable et, en revanche, accepter une solution qui viole légèrement une des contraintes du problème.

L'opérateur humain doit donc jouer un rôle crucial dans la sélection de cette solution finale. L'humain est nécessaire pour définir la hiérarchisation des différents critères. En plus, les planificateurs peuvent considérer aussi d'autres contraintes qu'il est difficile de spécifier lors de la modélisation du problème : personnelles (*e.g.*, un client qui doit être servi par un véhicule spécifique), environnementales (*e.g.*, bouchons ou routes barrées en raison de travaux), sociales (*e.g.*, fêtes ou occasion spéciales qui peuvent entraîner une sur-demande de service), etc. Un mode de contrôle manuel est donc proposé pour la réalisation de cette tâche.

Poursuite des tournées en temps réel

La modification des tournées en temps réel est une tâche avec des caractéristiques très similaires aux tâches liées à la résolution du problème. Nous considérons qu'il est nécessaire que le système d'aide à la décision intègre des algorithmes capables de proposer des solutions réalisables après chaque difficulté ou contretemps apparu en cours de route. L'opérateur humain doit disposer des outils pour valider et modifier les propositions faites par l'algorithme. Nous proposons donc les trois mêmes modes de contrôle proposés pour les sous-tâches liées à la résolution du problème : consultatif, interactif et de supervision.

III.4 Mécanisme de résolution

Le mécanisme de résolution est composé de deux modules : la programmation par contraintes et les algorithmes spécifiques du problème de tournées de véhicules. L'articulation de ces deux composants offre un cadre idéal pour l'intégration dans un mécanisme de résolution d'un système d'aide à la décision dont le but principal est de favoriser la coopération homme-machine. D'un côté, la programmation par contraintes facilite la détection d'inconsistances après la prise de chaque décision et, d'un autre côté, les algorithmes spécifiques permettent de proposer des solutions efficaces pour la résolution des problèmes.

III.4.1 Programmation par contraintes

Afin d'intégrer, de manière interactive et incrémentale, les contraintes du problème, nous nous plaçons dans le paradigme de la programmation par contraintes (PPC) (Dechter, 2003; Rossi et al., 2006). Un tel paradigme présente en effet de multiples avantages pour la problématique qui nous occupe.

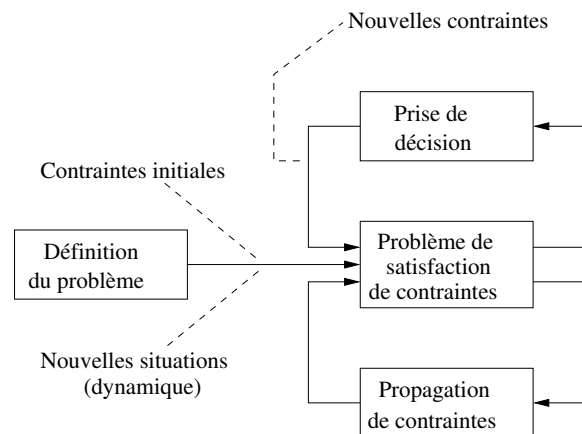


FIG. III.4 – Programmation par contraintes

En premier lieu, et c'est la principale originalité de la PPC, il s'agit d'une approche qui favorise la séparation entre la définition formelle du problème, les mécanismes d'analyse et de filtrage du système de contraintes induit par cette définition (propagation) et les méthodes de résolution (cf. figure III.4). Cette caractéristique nous paraît particulièrement importante pour la conception de systèmes flexibles permettant la considération d'une coopération au sein du

système homme-machine.

Ensuite, en PPC, le traitement des contraintes n'est pas globalisé, mais au contraire particularisé en fonction du type de la contrainte traitée, suivant l'assertion « 1 contrainte = 1 algorithme ». Ce principe permet notamment de décomposer le système de contraintes global en autant de sous-problèmes que de types de contraintes et de spécialiser ainsi le processus de propagation de contraintes (filtrage de valeurs inconsistantes du domaine d'une variable). Ce traitement individualisé des contraintes permet aussi de proposer des mécanismes de guidage sur les contraintes à relâcher (inversion du modèle) adaptés aux caractéristiques de chaque type de contrainte (voir le chapitre V).

Enfin, le paradigme de la PPC par son aptitude à l'ajout incrémental de nouvelles contraintes, est bien adapté à la prise en compte de la dynamique du problème traité. Des formalismes spécialisés comme ceux des CSP dynamiques (Mittal and Falkenhainer, 1990) et des CSP mixtes et conditionnels (Gelle and Faltings, 2003) sont tout particulièrement prometteurs.

III.4.2 Algorithmes spécifiques pour la résolution du problème

Un des avantages de l'approche proposée est la possibilité de réaliser une hybridation entre les mécanismes de PPC et d'autres techniques de résolution davantage dédiées. Le système proposé permet aisément d'intégrer cet aspect. Après application de la PPC, nous pouvons traiter le problème par les techniques de résolution qui se sont avérées les plus efficaces pour résoudre des problèmes semblables au problème réel. Les décisions prises par ces techniques et algorithmes peuvent ensuite être réintégrées, de la même manière que les décisions prises par l'opérateur humain, dans le schéma de résolution de la PPC.

Dans ce contexte, une piste intéressante à suivre est l'intégration de techniques de sélection des algorithmes spécifiques à appliquer selon le type de problème. Dans Desrochers et al. (1999), les auteurs proposent une méthode basée sur la recherche de plus courts chemins pour associer des techniques diverses de résolution du problème réel à résoudre. Cette identification est faite à partir des ressemblances entre le problème réel, une fois modélisé, et les problèmes classiques de la littérature pour lesquels des méthodes et algorithmes de résolution efficaces sont connus.

Les algorithmes intégrés actuellement dans le système d'aide à la décision sont décrits dans les paragraphes suivants. D'abord, le problème est défini formellement. Ensuite, nous présentons les algorithmes pour les trois étapes indépendantes de résolution du problème (la sélection de

véhicules, l'affectation des clients aux véhicules et la sélection des itinéraires). L'objectif de cette sous-division est que le système doit être capable d'assister l'opérateur humain non seulement dans la résolution globale du problème, mais aussi dans chacune des tâches qui sont ou peuvent être réalisées (dépendant de la stratégie de l'opérateur) pendant la résolution du problème. Des algorithmes pour la résolution de chacune des sous-tâches doivent être donc intégrés dans le mécanisme de résolution.

III.4.3 Définition formelle du problème

Le problème considéré est le problème de tournées de véhicules avec prise en compte de différents types de contraintes identifiées lors de l'analyse du domaine de travail. Dans ce paragraphe, nous introduisons la notation relative au problème.

Les objets du problème sont :

- les clients (C_i , $i = 1..nc$)
- les véhicules (V_j , $j = 1..nv$)
- les conducteurs (D_l , $l = 1..ndr$)
- les dépôts (De_t , $t = 1..nde$)
- et les marchandises (P_k , $k = 1..np$)

Un ensemble de nc clients doit être servi par un ensemble de nv véhicules. Chaque client i demande une quantité d_i^k de produit k . Les demandes des clients peuvent être indifféremment des livraisons ($d_i^k > 0$) ou des collectes ($d_i^k < 0$). Un temps de service Ts_i est associé à chaque client. Chaque véhicule j dispose d'une capacité limite relative au poids (C_j^p), au volume (C_j^v) et à la longueur maximale transportable (C_j^l). De la même manière, les caractéristiques de chaque produit k sont le poids (P_k^p), le volume (P_k^v) et la longueur (P_k^l).

La variable x_i^j est utilisée pour indiquer la présence du client i sur la route du véhicule j .

$$x_i^j = \begin{cases} 1 & \text{si le client } i \text{ est servi par le véhicule } j \\ 0 & \text{sinon} \end{cases}$$

L'ensemble des contraintes de capacité liées aux livraisons peuvent maintenant être exprimées

de la manière suivante :

$$\begin{aligned}
 \sum_i^{nc} \sum_k^{np} x_i^j \times \max(d_i^k, 0) \times P_k^p &\leq C_j^p \quad \forall j = 1..nv \\
 \sum_i^{nc} \sum_k^{np} x_i^j \times \max(d_i^k, 0) \times P_k^v &\leq C_j^v \quad \forall j = 1..nv \\
 \max_{i, d_i^k > 0} (x_i^j \times P_k^l) &\leq C_j^l \quad \forall j = 1..nv
 \end{aligned} \tag{III.1}$$

et l'ensemble de contraintes liées aux collectes sont :

$$\begin{aligned}
 \sum_i^{nc} \sum_k^{np} x_i^j \times \max(-d_i^k, 0) \times P_k^p &\leq C_j^p \quad \forall j = 1..nv \\
 \sum_i^{nc} \sum_k^{np} x_i^j \times \max(-d_i^k, 0) \times P_k^v &\leq C_j^w \quad \forall j = 1..nv \\
 \max_{i, d_i^k < 0} (x_i^j \times P_k^l) &\leq C_j^l \quad \forall j = 1..nv
 \end{aligned} \tag{III.2}$$

Dans le cas où de nouvelles contraintes relatives aux caractéristiques des marchandises doivent être prises en compte, le système d'aide à la décision permet de les ajouter.

Des contraintes sur les différentes tournées sont aussi considérées. Les tournées peuvent être limitées en distance (V_j^D), en durée (V_j^T) ou en nombre de clients (V_j^C). Notons R_j la tournée réalisée par le véhicule j . Les grandeurs $D(R_j)$ et $T(R_j)$ représentent respectivement la distance et la durée de la tournée R_j . L'ensemble de contraintes des tournées peut alors être défini comme :

$$\begin{aligned}
 D(R_j) &\leq V_j^D \quad \forall j = 1..nv \\
 T(R_j) &\leq V_j^T \quad \forall j = 1..nv \\
 \sum_i^{nc} x_i^j &\leq V_j^C \quad \forall j = 1..nv
 \end{aligned} \tag{III.3}$$

Une fenêtre de temps décrit l'intervalle dans lequel un client doit être servi : $FTC_i = [r_i, d_i]$. Les dépôts et les conducteurs ont aussi leurs fenêtres de temps respectives : $FTDe_t = [r_t, d_t]$ et $FTD_l = [r_l, d_l]$. Ces fenêtres décrivent respectivement les intervalles de temps dans lequel le départ et l'arrivée des véhicules aux dépôts doivent être réalisés et ceux du temps de travail des conducteurs.

Nous considérons aussi des contraintes entre les objets physiques du problème. L'un de ces

types de contraintes concerne les contraintes de précédence (simple et immédiate) entre les clients. S'il existe une contrainte de précédence simple entre deux clients i et i' ($i \prec i'$), le client i doit alors être servi avant le client i' . En revanche, la contrainte de précédence immédiate ($i \preceq i'$) implique que le client i' doit être servi immédiatement après le client i et par le même véhicule. Enfin, dans ce type de contraintes, nous considérons les contraintes d'affectation ou d'incompatibilité (exclusion) entre les objets physiques du problème. Dans les problèmes réels, nous trouvons souvent des situations pour illustrer ce type de contraintes, par exemple l'affectation entre un type de marchandise et un type de véhicule avec des caractéristiques particulières qui assure la sûreté du transport, ou dans le cas où deux clients doivent être impérativement servis par le même véhicule pour la raison que la collecte d'un des clients garantit la livraison de l'autre.

III.4.4 Algorithmes pour la sélection de véhicules

L'objectif principal des algorithmes proposés pour la phase de sélection des véhicules est d'aider l'opérateur humain à sélectionner les véhicules qui vont être utilisés pour servir les clients. La solution de cette étape peut parfois être non-réalisable car les véhicules sélectionnés peuvent ne pas être suffisants pour satisfaire les demandes des clients sans violer aucune des contraintes du problème. Toutefois, cette étape peut être très utile aux planificateurs pour une gestion efficace de la flotte, notamment dans les problèmes où la location de véhicules est nécessaire.

Dans cette phase de résolution, les véhicules disponibles sont groupés selon leurs caractéristiques. Les variables de décision de cette phase sont donc le nombre de véhicules de chaque type qui vont être utilisés pour servir les clients. Notons tv le nombre de types de véhicules différents et nv_τ le nombre de véhicules du type τ qui vont être utilisés pour servir les clients.

Mode consultatif : vérification de la faisabilité de la solution proposée par l'opérateur humain.

Les algorithmes de ce mode s'occupent de vérifier si la solution proposée par l'opérateur humain est réalisable ou non. L'algorithme vérifie en premier si la solution proposée satisfait les contraintes de capacité des véhicules relatives aux limites de poids, volume et longueur maximale transportable (équations III.4, III.5 et III.6). L'algorithme vérifie aussi les contraintes d'affectation entre objets physiques relatives aux véhicules de manière à déterminer si la présence d'un

véhicule d'un type spécifique est nécessaire ou non.

$$\sum_{i=1}^{nc} \sum_{k=1}^{np} d_i^k \times P_k^p \leq \sum_{\tau=1}^{tv} nv_{\tau} \times C_{\tau}^p \quad (\text{III.4})$$

$$\sum_{i=1}^{nc} \sum_{k=1}^{np} d_i^k \times P_k^v \leq \sum_{\tau=1}^{tv} nv_{\tau} \times C_{\tau}^v \quad (\text{III.5})$$

$$\max_{i, d_i^k \neq 0} (P_k^l) \leq \max_{nv_{\tau} > 0} (C_{\tau}^l) \quad (\text{III.6})$$

Ensuite, l'algorithme compare le nombre de véhicules de la solution proposée par l'opérateur ($\sum_{\tau}^{tv} nv_{\tau}$) avec une borne inférieure pour le nombre minimum de véhicules nécessaires pour servir les clients (BI_{nv}). Cette borne est calculée à partir des contraintes induites par les fenêtres de temps des clients, par les fenêtres de temps des dépôts et par les contraintes sur les tournées (distance, durée et nombre maximum de clients par tournée). Pour chacune de ces contraintes, une borne inférieure est calculée. La borne inférieure pour le nombre de véhicules BI_{nv} est donc égale au maximum des cinq bornes inférieures calculées.

Le nombre de véhicules de la solution proposée par l'opérateur doit être alors au moins égal à la borne inférieure :

$$\sum_{\tau}^{tv} nv_{\tau} \geq BI_{nv} \quad (\text{III.7})$$

Pour le calcul de la borne inférieure BI_{nv}^{FTC} liée aux contraintes temporelles induites par les fenêtres de temps des clients ($FTC_i = [r_i, d_i]$), l'algorithme détermine d'abord pour chaque paire de clients s'ils sont en conflit. Un client est en conflit avec un autre si les deux clients ne peuvent être servis par le même véhicule. Notons par $t_{ii'}$ la durée du trajet entre le client i et le client i' . Pour une paire de clients $\langle i, i' \rangle$, si $r_i + Ts_i + t_{ii'} > d_{i'}$ et $r_{i'} + Ts_{i'} + t_{i'i} > d_i - Ts_i$, alors les deux clients $\langle i, i' \rangle$ sont en conflit. Les clients qui sont mutuellement en conflit sont groupés en sous-ensembles de clients. Ainsi, BI_{nv}^{FTC} est égal au nombre d'éléments du plus grand sous-ensemble, car aucun client de ce sous-ensemble ne peut partager le véhicule avec les autres clients du sous-ensemble.

Pour le calcul des bornes liées aux contraintes relatives à la distance et à la durée maximale des tournées (BI_{nv}^D et BI_{nv}^T , respectivement), l'algorithme calcule d'abord une borne inférieure relative à la distance ($BI_{distance} = \sum_i^{nc} \min_{i' \neq i} (d_{ii'})$) et à la durée minimale de la tournée ($BI_{duree} = \sum_i^{nc} \min_{i' \neq i} (t_{ii'}) + Ts_i$) nécessaire pour servir tous les clients. La durée des trajets entre clients peut

être calculée à partir de l'information sur la vitesse des véhicules et peut être régulièrement actualisée suite à un retour d'information après chaque tournée. La borne relative à la distance peut ainsi être calculée par l'équation suivante :

$$BI_{nv}^D = \frac{BI_{distance}}{\max_j(V_j^D)} \quad (III.8)$$

et l'expression pour le calcul de la borne relative à la contrainte sur la durée des tournées vaut :

$$BI_{nv}^T = \frac{BI_{duree}}{\max_j(V_j^T)} \quad (III.9)$$

Enfin, la borne liée à la contrainte sur le maximum nombre de clients à servir par une tournée est égale à :

$$BI_{nv}^C = \frac{nc}{\max_j(V_j^C)} \quad (III.10)$$

La dernière borne inférieure est la borne liée aux contraintes définies par les fenêtres de temps des dépôts (BI_{nv}^{FTDe}). Cette borne calcule le nombre de véhicules nécessaires pour satisfaire la demande des clients lorsqu'on considère l'heure d'ouverture et de fermeture des dépôts :

$$BI_{nv}^{FTDe} = \frac{BI_{duree}}{\max_t(d_t) - \min_t(r_t)} \quad (III.11)$$

L'algorithme de plus grande complexité pour le calcul de ces bornes est celui chargé de déterminer le sous-ensemble de clients en conflit avec un plus grand nombre d'éléments dans le cas de durées de trajet asymétriques ($t_{ii'} \neq t_{i'i}$). La complexité de cet algorithme est en $O(n^3)$.

Le calcul des cinq bornes permet de déterminer la valeur de la borne inférieure pour le nombre de véhicules à utiliser :

$$BI_{nv} = \max\{BI_{nv}^{FTC}, BI_{nv}^{FTDe}, BI_{nv}^D, BI_{nv}^T, BI_{nv}^C\} \quad (III.12)$$

Si les contraintes liées aux limites de capacité de transport des véhicules sont satisfaites, les requêtes de véhicules respectées et le nombre de véhicules de la solution proposée par l'opérateur humain supérieur ou égal à BI_{nv} , la solution est alors acceptée comme valide. Dans le cas où les véhicules sélectionnés dans la solution proposée par l'opérateur ne sont pas en nombre suffisant

pour assurer le service des clients, le système dispose d'un mécanisme d'assistance à l'opérateur pour la relaxation des contraintes du problème de manière à pouvoir servir les clients du problème avec les véhicules sélectionnés. Ces mécanismes de guidage pour la relaxation des contraintes sont présentées dans le chapitre V.

Mode de supervision : minimisation du nombre de véhicules.

Un algorithme qui propose à l'opérateur une ou plusieurs solutions sur les véhicules à utiliser pour satisfaire la demande des clients est intégré dans le mécanisme de résolution. L'algorithme minimise le nombre de véhicules à utiliser pour servir les clients ($\min \sum_{\tau}^{tv} nv_{\tau}$). Les solutions proposées par l'algorithme satisfont donc les contraintes liées aux limites de capacité de transport des véhicules (équations III.4, III.5 et III.6) et le nombre de véhicules de chaque solution est supérieur ou égal à la borne inférieure BI_{nv} (équation III.12).

Tout d'abord, nous considérons uniquement une des deux limitations (poids ou volume). Un algorithme exact de complexité linéaire (algorithme 1) est proposé pour le calcul d'une solution qui minimise le nombre de véhicules. Le principe général de l'algorithme est de sélectionner itérativement les véhicules de plus grande capacité. Une fois achevée la sélection des véhicules qui assurent la satisfaction de la contrainte de capacité, nous vérifions que la borne inférieure BI_{nv} est satisfaite. Si ce n'est pas le cas, des véhicules sont ajoutés à la solution de manière à arriver à satisfaire l'équation III.7.

Finalement, un algorithme qui permet de proposer à l'opérateur une liste de solutions intéressantes est également intégré dans le mécanisme de résolution (voir algorithme 2). Le principe de l'algorithme est de remplacer les véhicules de plus grande capacité par des véhicules plus petits, ce qui est toujours intéressant pour une question de coût et de commodité. La fonction *Remplacement_Possible(solution, exces, V_r , V_a)* retourne vrai, dans le cas où il est possible de remplacer un véhicule de la solution par un véhicule avec une capacité moins importante sans violer la contrainte de capacité. Cette fonction détermine donc le véhicule V_r de plus grande capacité qui peut être remplacé dans la solution par un autre véhicule V_a de plus petite capacité, c'est-à-dire que la condition $C_{V_a}^p + exces \geq C_{V_r}^p$ est satisfaite. Notons que la liste de solutions proposées par l'algorithme n'est pas exhaustive car l'algorithme modifie chaque solution une seule fois.

Ces algorithmes sont très efficaces (bonnes solutions, complexité polynomiale) pour la résolu-

Algorithme 1: Minimisation du nombre de véhicules

Pas 1 : Calculer le poids total à transporter à partir des demandes clients,

$$poids \leftarrow \sum_i^{nc} \sum_k^{np} d_i^k \times P_k^p$$

si le poids total transportable est plus grand que le poids à transporter, $\sum_j^{nv} C_j^p \geq poids$, **alors**

tant que ($poids > 0$) **faire**

 Pas 2 : Sélectionner le véhicule V_s disponible avec une capacité de poids à transporter maximale

si $C_{V_s}^p < poids$ **alors**

 Pas 3 : Ajouter le véhicule V_s à la solution

sinon

 Pas 4 : Sélectionner le véhicule V_s avec une capacité de poids à transporter minimale mais supérieure à $poids$

 Pas 5 : Ajouter le véhicule V_s à la solution

 Pas 6 : Actualiser le poids, $poids \leftarrow poids - C_{V_s}^p$

 Pas 7 : Calculer l'excès de capacité des véhicules, $exces \leftarrow |poids|$

sinon

 └ Echec : Problème non-réalisable

si le nombre de véhicules de la solution est inférieur à la borne inférieure BI_{nv} **alors**

 Pas 8 : Sélectionner le véhicule V_s

 Pas 9 : Ajouter le véhicule V_s à la solution

 Pas 10 : Actualiser l'excès de capacité de transport des véhicules,

$$exces \leftarrow exces + C_{V_s}^p$$

tion du problème lorsque seule une contrainte liée à la capacité (poids ou volume) est considérée. Toutefois, dans certains problèmes réels, notamment ceux avec marchandises volumineuses et lourdes à la fois, ces algorithmes ne sont pas suffisants pour produire des solutions avec un nombre minimum de véhicules. Nous n'avons pas pu déterminer un algorithme de complexité polynomiale pour résoudre ce type de problème. Ce que nous proposons est une résolution en deux étapes. Dans un premier temps, l'algorithme 1 est lancé séparément pour chacune des contraintes (poids et volume). Si une des deux solutions proposées satisfait les deux contraintes, une solution optimale est atteinte et nous pouvons arrêter la recherche. Dans le cas où aucune des deux solutions ne satisfait les deux contraintes (poids et volume), le nombre maximum de véhicules des deux solutions nv_{max} est utilisé comme borne pour la deuxième étape de la résolution. Cette étape consiste à résoudre un problème de faisabilité. Il s'agit de trouver une solution avec un nombre de véhicules fixé (nv_{max}) qui permette de satisfaire les contraintes de capacité liées au poids et au volume. Si aucune solution n'est trouvée, le nombre de véhicules de

Algorithme 2: Génération de solutions avec un nombre minimum de véhicules

Pas 1 : Insérer la solution de l'algorithme 1 dans une liste de solutions
tant que *Remplacement_Possible(solution, excès, V_r, V_a)* **faire**

Pas 2 :	Créer une nouvelle solution (<i>nsolution</i>) avec la suppression du véhicule V_r et avec l'ajout du véhicule V_a
Pas 3 :	Ajouter la nouvelle solution dans la liste de solutions
Pas 4 :	Actualiser l'excès de capacité de la solution, $excès \leftarrow excès + C_{V_a}^p - C_{V_r}^p$
Pas 5 :	Actualiser la solution à modifier, $solution \leftarrow nsolution$

la solution est incrémenté d'une unité. L'algorithme de cette deuxième étape de la résolution est malheureusement de complexité non polynomiale.

Mode interactif : finir la résolution.

L'algorithme qui correspond à ce mode de contrôle permet de compléter une solution partielle proposée par l'opérateur humain. Les algorithmes proposés sont les mêmes algorithmes que pour la minimisation du nombre de véhicules. Cependant, il existe déjà un certain nombre de véhicules qui doivent impérativement apparaître dans la solution proposée à l'opérateur. Ces véhicules ont été sélectionnés au préalable par l'opérateur humain.

III.4.5 Algorithmes pour l'affectation des clients aux véhicules

Dans cette deuxième étape de la résolution, le but est de déterminer l'ensemble de clients qui vont être servis par chaque véhicule. La solution obtenue satisfait toutes les contraintes du problème. Cette solution est donc réalisable.

Mode consultatif : outil d'aide pour une approche de résolution manuelle.

Dans ce mode, l'opérateur propose une affectation pour chaque client. Les algorithmes intégrés dans le système vérifient la faisabilité des décisions prises par l'opérateur humain. Le but des algorithmes est de détecter, en premier lieu, si les contraintes du problème sont violées et, en second lieu, de vérifier s'il existe une solution réalisable.

Pour cela, nous proposons un algorithme en différentes étapes (voir algorithme 3). Cet algorithme est lancé après chaque décision (affectation d'un client i à un véhicule j) de l'opérateur.

D'abord, un algorithme vérifie que toutes les contraintes entre les objets physiques qui impliquent le client i ne sont pas violées. Puis, un deuxième algorithme vérifie que le client i n'est

Algorithme 3: Vérification de la faisabilité de la décision d'affecter le client i au véhicule j

Pas 1 : Vérification de la satisfaisabilité des contraintes particulières relatives à l'affectation du client i , *par exemple l'affectation du client i à un véhicule particulier*
 Pas 2 : Vérification que le client i n'est pas en conflit avec les autres clients du véhicule j
 Pas 3 : Test d'admissibilité basé sur le raisonnement énergétique
si aucune contrainte n'est violée **alors**
 | Pas 4 : Recherche arborescente basée sur les divergences pour arriver à trouver une solution réalisable
sinon
 | Echec : Le client i ne peut pas être affecté au véhicule j

pas en conflit avec les autres clients déjà affectés au véhicule. La complexité de cet algorithme est donc linéaire par rapport au nombre de clients déjà affectés.

Ensuite, dans le cas où aucune inconsistance n'a été détectée, nous proposons un algorithme basé sur le raisonnement énergétique (Lopez and Esquirol, 1996) de manière à vérifier que les fenêtres de temps des clients affectés au véhicule j peuvent être respectées. Cet algorithme est très similaire à l'algorithme proposé dans le paragraphe VI.4.2. L'avantage de cet algorithme par rapport aux algorithmes classiques de raisonnement énergétique est qu'il permet de considérer les temps de trajet entre les clients dans le test de faisabilité. Le temps de service chez les clients représente la durée des tâches, le véhicule est la machine ($m = 1$ par rapport à la notation de l'algorithme du chapitre VI) et la durée de trajet entre les clients peut être vue comme le temps de préparation des machines. Il s'agit de vérifier que le client i qui vient d'être affecté au véhicule j peut être inséré dans la tournée.

Dans le problème sans considération des temps de préparation, l'analyse de l'admissibilité est complète concernant l'étude des bilans énergétiques sur les intervalles de temps $\Delta = [t_1, t_2]$, dont les bornes sont dans l'un des trois produits cartésiens $O_1 \times O_2$, $O_1 \times O(t)$, $t \in O_1$ et $O(t) \times O_2$, $t \in O_2$ (Lopez, 2003). Les ensembles de dates peuvent être énumérés en $O(n^2)$ et sont définis comme :

$$\begin{aligned}
 O_1 &= \{r_i\} \cup \{d_i - Ts_i\} \cup \{r_i + Ts_i\} \\
 O_2 &= \{d_i\} \cup \{r_i + Ts_i\} \cup \{d_i - Ts_i\} \\
 O(t) &= \{r_i + d_i - t\}
 \end{aligned}
 \tag{III.13}$$

Pour notre problème, il suffit de considérer les intervalles Δ où la consommation du client i est positive. Les bilans énergétiques doivent alors être vérifiés pour les intervalles Δ où $t_1 < r_i + Ts_i$

et $t_2 > d_i - Ts_i$. La complexité de l'algorithme sans considération des temps de trajet est en $O(n^3)$. La consommation énergétique due aux temps de trajet telle qu'elle est présentée dans le chapitre VI augmente la complexité jusqu'à $O(n^4)$. Ces temps de trajet peuvent cependant être considérés de manière "légère" sans augmenter la complexité de l'algorithme. Cette version "légère" (ou "molle") consiste à considérer pour chaque client avec une consommation positive dans l'intervalle, la plus petite des durées de trajet entre le client et tous les autres clients du véhicule. Dans la version "dure" proposée dans le chapitre VI, nous considérons pour chaque client avec une consommation positive, la plus petite des durées de trajet entre le client et les autres clients qui ont aussi une consommation positive (et non pour tous les clients du véhicule comme dans la version "molle").

Enfin, une recherche arborescente basée sur le concept de divergences (voir le paragraphe VI.5 pour une description détaillée de la méthode) est proposée de manière à trouver une solution réalisable et ainsi pouvoir accepter la décision prise par l'opérateur. Cette recherche est nécessaire car le raisonnement énergétique n'assure pas l'existence d'une solution à cause des temps de trajet entre clients. L'opérateur peut indiquer le temps maximum de calcul autorisé pour cette recherche car une recherche complète peut être très gourmande en temps de calcul, notamment pour les véhicules avec un grand nombre de clients. Une fois la recherche achevée, si aucune solution réalisable n'a été trouvée, l'opérateur humain doit alors modifier l'affectation du client i , ou celle d'un des autres clients affectés au véhicule.

Mode de supervision : affectation des clients aux véhicules.

Le système intègre aussi un algorithme qui permet d'affecter automatiquement les clients aux véhicules. Pour cela, nous adoptons le principe de l'algorithme de "sweep" proposé dans Gillet and Miller (1974). Tout d'abord, les clients soumis à des contraintes d'affectation aux véhicules sont affectés. Ensuite, le reste des clients est affecté aux véhicules. L'ordre d'affectation des clients est l'ordre établi par l'algorithme de "sweep". Chaque client est affecté au véhicule actif (véhicules avec clients déjà affectés) avec une distance moyenne minimale entre le client et les clients qui sont déjà affectés au véhicule. L'algorithme 3 permet de déterminer si le client peut être affecté ou non à un véhicule. Dans le cas où le client ne peut être affecté à aucun des véhicules actifs, le client est alors affecté à un des véhicules disponibles qui n'a encore aucun client.

Mode interactif : finir la résolution.

Le principe de ce mode est de compléter une solution qui a été initialement proposée par l'opérateur humain. L'algorithme utilisé est le même que celui décrit pour le mode de supervision. Remarquons que la collaboration homme-machine peut être très efficace pour cette étape de résolution. En effet, un des principaux inconvénients de l'algorithme est qu'il n'est pas capable d'identifier des *patterns* à partir de la configuration spatiale (localisations des clients) du problème, ce qui parfois peut induire à une performance pauvre de la solution obtenue. Par exemple, les algorithmes basés sur des heuristiques ou sur les règles de priorité ne sont pas capables d'identifier de manière efficace les limites entre tournées, c'est-à-dire, de choisir le bon moment pour arrêter d'affecter les clients à un véhicule et d'ajouter une nouvelle tournée au problème. Une des qualités de l'humain est en effet cette facilité à reconnaître ces modèles spatiaux qui peuvent être utilisés pour initialiser la solution et aider l'algorithme à proposer des solutions de meilleure qualité.

III.4.6 Algorithmes pour la sélection des itinéraires

Dans cette dernière phase de résolution du problème, la séquence des clients à servir pour chaque véhicule doit être définie. Le résultat de cette étape est une solution complète du problème. Ceci implique que si les deux autres étapes de la résolution n'ont pas été réalisées, c'est dans cette étape que le problème complet doit être résolu.

Mode consultatif : outil d'aide pour une approche de résolution manuelle.

Dans ce mode de contrôle, l'opérateur propose la séquence de clients pour chaque véhicule. Après chaque décision de l'opérateur, un algorithme vérifie si la décision prise par l'opérateur est réalisable. Actuellement, l'algorithme intégré dans le système d'aide à la décision consiste en une recherche arborescente basée sur les divergences de manière à trouver une solution réalisable (cf. chapitre VI). L'algorithme vérifie que la sous-séquence proposée par l'opérateur est réalisable et qu'il existe une solution réalisable qui contienne cette sous-séquence de clients, mais aussi les autres clients qui sont aussi affectés dans le même véhicule et qui n'ont pas encore été ordonnés dans la séquence.

Des techniques de propagation de contraintes peuvent aussi être intégrées dans le mécanisme

de résolution. Ces techniques permettent de détecter des inconsistances plus rapidement et ainsi accroître l'efficacité du système. En effet, dans le cas où il n'existe pas une solution réalisable, la recherche arborescente peut être très gourmande en temps de calcul.

Mode de supervision : sélection des itinéraires.

L'idée de ce mode de contrôle est de proposer une solution complète au problème. Pour cela, le système dispose de plusieurs mécanismes : un algorithme fondé sur les principes des techniques classiques d'insertion de clients de la littérature et deux algorithmes d'optimisation d'une solution. Le but du premier algorithme est de proposer rapidement une solution de qualité à l'opérateur. Les algorithmes d'optimisation ont pour objectif d'améliorer la qualité de la solution. Ces algorithmes sont décrits dans les paragraphes qui suivent. L'hybridation entre les métaheuristiques et la programmation par contraintes a été déjà proposée efficacement pour résoudre les problèmes de tournées de véhicules (Pesant and Gendreau, 1999; Caseau et al., 1999; Backer et al., 2000).

Algorithme pour déterminer une solution réalisable. Une fois les clients affectés, l'algorithme se base sur la même logique que l'algorithme d'insertion proposé par Clarke and Wright (1964). Chaque client est inséré dans la séquence dans la position qui minimise la durée de la tournée. Dans le cas où un client ne peut pas être inséré dans la tournée, une recherche arborescente basée sur les divergences est lancée de manière à trouver une solution réalisable. Il faut rappeler qu'il existe toujours une séquence réalisable pour chaque tournée, car celle-ci est une des conditions d'acceptation d'une proposition d'affectation d'un client à un véhicule.

Un des principaux inconvénients de l'algorithme d'insertion proposé est qu'il peut devenir très inefficace lorsque le problème est sur-constraint (Kilby et al., 2000). Dans l'algorithme proposé, l'efficacité de l'algorithme d'insertion est très importante. La recherche d'une solution réalisable peut être très coûteuse en temps de calcul. Pour cette raison, nous proposons un algorithme d'insertion alternatif qui priorise la recherche d'une solution réalisable plutôt que la minimisation de la fonction objectif. Il est basé sur le principe proposé dans Liu and Shen (1999). Pour déterminer le prochain client à ordonner dans la séquence, l'algorithme considère le client avec un nombre de positions minimum dans la séquence où le client peut être placé sans violer aucune des contraintes du problème. Les clients les plus conflictuels sont ainsi ordonnés en premier dans

la séquence.

Une des questions qu'il faut résoudre est de trouver un critère pour déterminer lequel des deux algorithmes d'insertion doit être appliqué selon les caractéristiques du problème. Une solution envisageable est d'utiliser normalement l'algorithme d'insertion qui se focalise sur la minimisation de la durée de la tournée, sauf dans le cas où il existe un des clients qui n'a pas encore été ordonné avec moins de deux positions possibles dans la séquence. Dans ce cas, le séquençement de ce client conflictuel est prioritaire.

Algorithmes d'optimisation d'une solution. L'opérateur humain dispose de deux algorithmes pour l'optimisation des solutions. Le premier des algorithmes favorise une approche d'optimisation locale de la solution et le deuxième est un algorithme de recherche locale pour l'optimisation complète de la solution. L'opérateur humain doit limiter le temps d'exécution de ces algorithmes.

Le premier des algorithmes consiste en une recherche basée sur les divergences (voir le paragraphe VI.5 pour une description détaillée de la méthode) qui permet l'optimisation des tournées de manière indépendante. L'algorithme vise donc à optimiser l'ordre des clients dans la séquence de la tournée sans autoriser un changement dans l'affectation des clients. L'avantage d'une telle approche est que l'opérateur dispose d'un algorithme qui lui permet d'optimiser la solution tout en gardant les tournées qui satisfont les critères établis. La structure de l'algorithme est très similaire aux algorithmes présentés dans le chapitre VI. Nous partons d'une solution pour une des tournées ; cette solution peut être proposée soit par l'opérateur humain soit par l'algorithme présenté dans le paragraphe précédent. Ensuite, la recherche arborescente est lancée de manière à explorer les solutions admissibles du problème. Une borne supérieure est calculée à chaque nœud de la recherche de manière à augmenter l'efficacité de l'algorithme. La recherche arborescente basée sur les divergences classique (LDS ou Limited Discrepancy Search) a été comparée avec un algorithme hybride de recherche locale avec des voisinages définis par les divergences de la recherche arborescente (CDS, voir le chapitre VI pour plus de détail). Deux versions de cet algorithme ont été proposées (CDS-V1, CDS-V2). La différence entre les deux versions est que les divergences sont spécifiées par rapport à des critères différents. Dans CDS-V1, une divergence peut être considérée comme la modification de la position d'un client dans une liste donnée (voir figure III.5 b). En revanche, dans CDS-V2, une divergence implique la modification dans cette

liste d'un bloc de clients (voir figure III.5 c). Les figures III.5 b et III.5 c illustrent les deux versions pour la manière de définir les divergences à partir d'une solution de référence décrite sur la figure III.5 a. Nous constatons que deux solutions identiques apparaissent pour les voisinages de CDS-V1 et CDS-V2, mais la troisième solution explorée est différente.

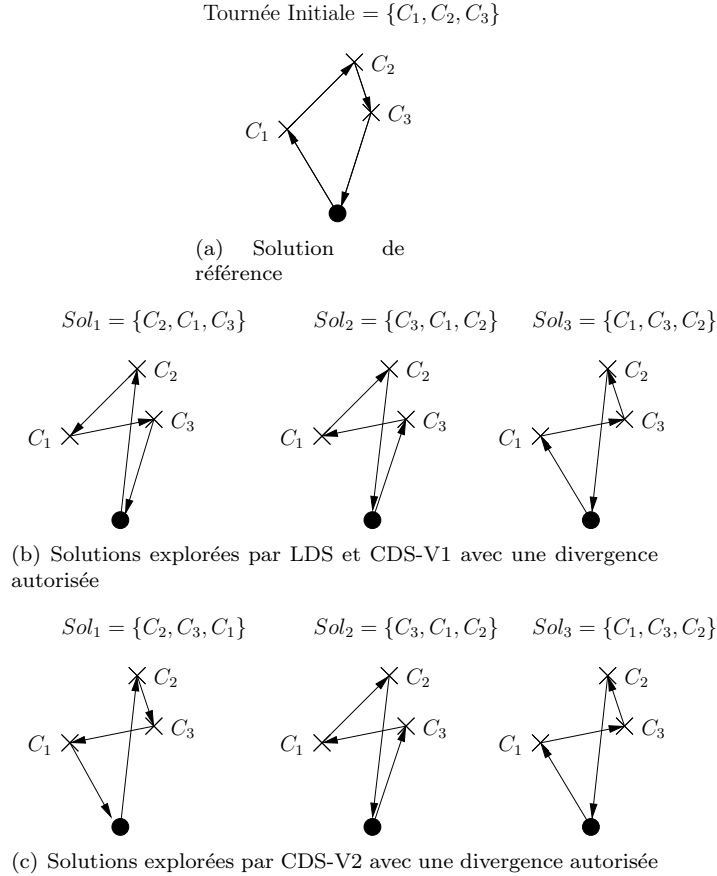


FIG. III.5 – Comparaison des voisinages explorés par CDS-V1 et CDS-V2

Notons que les voisinages définis par les divergences de la recherche ont une forte relation avec les voisinages classiques proposés dans la littérature pour la résolution du problème de tournées de véhicules (voir Gendreau and Tarantilis (2010) pour un état de l'art sur les voisinages décrits dans la littérature pour les problèmes de tournées de véhicules). Par exemple, la recherche des solutions à une divergence d'une route fixée est un sous-ensemble du voisinage connu sous le nom de "Or-opt" (Or, 1976) lorsque le nombre des clients des sections modifiées est égale à 1. Ce voisinage consiste à extraire chacun des clients de la séquence et essayer de les ré-insérer

dans le reste des positions possibles. La différence avec ce voisinage est que LDS autorise les ré-insertions des clients vers l'avant mais limite la distance de la ré-insertion vers l'arrière (voir la figure III.5 b). Dans la figure, nous observons que la tournée $R = \{C_3, C_2, C_1\}$ qui fait partie du voisinage Or-opt pour une Tournée Initiale = $\{C_1, C_2, C_3\}$ n'est pas explorée dans la recherche arborescente LDS avec une divergence autorisée. Ceci est dû à ce que la ré-insertion du client C_1 dans le séquence de clients est limitée par le nombre de divergences de la recherche.

Le deuxième algorithme (voir algorithme 4) est un algorithme d'optimisation de la solution complète. Pour cela, nous utilisons un schéma de recherche locale (voir le paragraphe VI.5.3 pour une description de ce schéma) où, pour chaque itération, un opérateur de réaffectation des clients aux véhicules basé sur des critères géographiques est appliqué. Pour chaque nouvelle affectation des clients, une optimisation des tournées qui ont été modifiées est lancée. Pour cette optimisation des tournées, nous utilisons l'algorithme qui vient d'être présenté dans le paragraphe précédent. Le nombre de divergences autorisées est limitée. Si pendant cette optimisation une nouvelle solution est trouvée, nous remplaçons la meilleure solution et le voisinage de cette nouvelle meilleure solution est alors explorée.

Il reste donc à définir quels sont les opérateurs pour le changement d'affectation des clients entre les différentes tournées. Dans la littérature, nous trouvons les voisinages connus avec le nom de Réaffectation et Echange proposés par Savelsbergh (1992) et le Cross-échange défini dans Taillard et al. (1997). Dans ce but, nous proposons deux opérateurs basés sur des critères géographiques. L'objectif de ces voisinages est de trouver le plus rapidement possible de bonnes affectations des clients aux véhicules. Pour le premier des voisinages, nous nous basons sur le calcul de la distance entre les clients et les centres de gravité des autres tournées. Cette distance est comparée avec la distance entre le client et le centre de gravité de sa propre tournée. Il s'agit de déterminer tous les clients (C_i) qui sont plus près du centre de gravité d'un autre tournée (G_{R_j}) que du centre de sa propre tournée (G_{R_q}). Le deuxième des voisinages consiste à comparer pour chaque client la distance par rapport aux autres clients qui ne font pas partie de la même tournée avec la distance par rapport au centre de gravité de sa propre route. S'il existe un client C_i de la route R_q qui est situé plus près d'un client $C_{i'}$ de la route R_j que du centre de gravité de sa propre route G_{R_q} , alors nous envisageons d'affecter ce client C_i au véhicule de la route R_j . Une fois déterminée la liste de nouvelles affectations spécifiées par le triplet $\{C_i, R_q, R_j\}$, où C_i représente le client qui, à l'origine, fait parti de la tournée R_q et qui doit être affecté à la tournée

R_j ($dist(C_i, G_{R_j}) < dist(C_i, G_{R_q})$ pour le premier des voisinages et $dist(C_i, C_{i'}) < dist(C_i, G_{R_q})$ pour le deuxième des voisinages), nous proposons alors d'explorer les voisinages définis par :

- (1) toutes les nouvelles réaffectations des clients. Ce voisinage consiste à affecter le client C_i au véhicule de la tournée R_j ;
- (2) tous les échanges d'affectations possibles entre deux clients. S'il existe dans la liste un triplet $\{C_i, R_q, R_j\}$ et un triplet $\{C_{i'}, R_j, R_q\}$, nous échangeons les affectations de ces deux clients de manière à ce que le client C_i soit affecté au véhicule de la tournée R_j et le client $C_{i'}$ soit affecté au véhicule de la tournée R_q .

Remarquons que ces voisinages sont des sous-voisinages des opérateurs Réaffectation et Echange (Savelsbergh, 1992).

Algorithme 4: Algorithme d'optimisation des tournées

Pas 1 : L'opérateur humain ou l'algorithme pour la construction d'une solution proposent une solution initiale, *Solution*

pour chaque tournée de Solution faire

- Pas 2 : Appliquer l'algorithme de recherche locale pour l'optimisation de la tournée (LDS ou CDS avec limitation du nombre de divergences autorisées)
- si une meilleure solution *MSolution* est trouvée alors**
 - └ Pas 3 : Actualiser la solution, $Solution \leftarrow MSolution$

tant que le temps spécifié par l'opérateur n'est pas atteint faire

- Pas 4 : Déterminer les réaffectations $\{C_i, R_q, R_j\}$ définies pour chaque critère
- pour chaque nouvelle affectation spécifiée par le voisinage (1) et (2) faire**
 - Pas 5 : Appliquer l'algorithme de recherche locale pour l'optimisation de la tournée R_q
 - Pas 6 : Appliquer l'algorithme de recherche locale pour l'optimisation de la tournée R_j
 - si une meilleure solution *MSolution* est trouvée alors**
 - └ Pas 7 : Actualiser la solution et revenir au Pas 4, $Solution \leftarrow MSolution$

Mode interactif : finir la résolution.

Finalement, l'opérateur dispose aussi du mode de contrôle interactif. La logique de ce mode est la même que dans les autres étapes de la résolution. L'opérateur propose d'abord une solution partielle avec le support du mécanisme de vérification décrit dans le paragraphe de la description du mode consultatif. Ensuite, l'algorithme pour déterminer une solution réalisable ordonne le reste de clients dans la séquence de la tournée, tout en respectant les décisions proposées par l'opérateur humain.

III.5 Résultats expérimentaux

Nous présentons les résultats de trois expérimentations réalisées pour évaluer l'efficacité des algorithmes proposés. Les algorithmes ont été implémentés en C++ et les expérimentations ont été réalisées sur un ordinateur de 2 GHz sous le système d'exploitation Linux Fedora 8 et avec 2 Go de mémoire RAM. Nous avons utilisé les instances de la littérature pour le VRPTW proposées par Solomon (1983) (25, 50 et 100 clients) et les extensions de ces instances (200 et 400 clients) (Gehring and Homberger, 1999). Ces instances sont regroupées par rapport à deux critères. Le premier est la position relative entre les clients : instances avec clients groupés en clusters (C), instances avec clients placés aléatoirement (R) et les instances mixtes (RC). Le deuxième critère concerne l'horizon de l'ordonnancement et les capacités des véhicules. Pour les instances des groupes C2, R2 et RC2 l'horizon de temps est plus large et les capacités de véhicules plus grandes que pour les instances des groupes C1, R1 et RC1, ce qui implique que chaque véhicule peut servir un plus grand nombre de clients.

Nous avons d'abord évalué la performance de la borne inférieure pour le nombre de véhicules nécessaires pour servir les clients. Nous avons comparé les meilleures solutions connues avec la borne proposée pour les instances avec 100, 200 et 400 clients. Notons que ces instances peuvent être utilisées pour évaluer la borne liée à la contrainte de capacité, la borne liée aux fenêtres de temps des clients (BI_{nv}^{FTC}) et la borne liée à la fenêtre de temps du dépôt (BI_{nv}^{FTDe}). La première colonne du tableau III.1 ($NbOpt$) indique, pour chaque groupe d'instances, le nombre de fois où il existe une solution réalisable avec le nombre de véhicules proposés par la borne ; le nombre d'instances de chaque groupe est indiqué entre parenthèses. Les deuxième et troisième colonnes indiquent respectivement le nombre de véhicules utilisés en moyenne dans les meilleures solutions réalisables décrites dans la littérature (Prescott-Gagnon et al., 2009; Nagata et al., 2010; Schrimpf et al., 2000) ($NVMoy$) et le nombre de véhicules en moyenne proposés par la borne inférieure ($NVMoyBI$). Finalement, la dernière colonne ($TCPU (s)$) affiche le temps de calcul moyen en secondes pour déterminer la valeur de la borne inférieure.

Nous pouvons affirmer que la borne proposée est relativement efficace. La valeur proposée par la borne est en effet égale à la valeur optimale pour la minimisation du nombre de véhicules pour la plupart des instances et les temps de calcul sont acceptables. Ces résultats renforcent l'idée déjà annoncée que cette étape de sélection des véhicules peut être très intéressante pour

Instances	$NbOpt$	$NVMoy$	$NVMoyBI$	$TCPU$ (s)
<i>C1-100</i>	9 (9)	10	10	0.02
<i>C2-100</i>	8 (8)	3	3	0.01
<i>R1-100</i>	3 (12)	11.83	10	0.03
<i>R2-100</i>	5 (11)	2.64	2	0.00
<i>RC1-100</i>	0 (8)	11.5	9.38	0.02
<i>RC2-100</i>	0 (8)	3.25	2	0.00
<i>C1-200</i>	7 (10)	18.8	18.2	0.1
<i>C2-200</i>	10 (10)	6	6	0.03
<i>R1-200</i>	9 (10)	18.1	18	0.13
<i>R2-200</i>	10 (10)	4	4	0.03
<i>RC1-200</i>	10 (10)	18	18	0.13
<i>RC2-200</i>	8 (10)	4.3	4	0.03
<i>C1-400</i>	6 (10)	37.6	36.4	1.41
<i>C2-400</i>	3 (10)	11.7	11	0.11
<i>R1-400</i>	9 (10)	36.2	36	1.91
<i>R2-400</i>	10 (10)	8	8	0.12
<i>RC1-400</i>	10 (10)	36	36	3.43
<i>RC2-400</i>	7 (10)	8.5	8	0.12
Total	124 (176)			

TAB. III.1 – Résultats de l'évaluation de la borne inférieure pour le nombre de véhicules à utiliser

les problèmes où la planification des véhicules doit être faite à l'avance, *e.g.*, les problèmes avec location de véhicules.

Dans la deuxième expérimentation, nous avons comparé la recherche arborescente basée sur les divergences classique (LDS, *Limited Discrepancy Search*) et les deux versions de l'algorithme hybride de recherche locale avec des voisinages définis par les divergences de la recherche arborescente (CDS-V1, CDS-V2). Pour cela, nous avons testé les trois méthodes sur les instances de Solomon (25, 50 et 100 clients). Une solution initiale est proposée grâce à une heuristique. Les tournées sont ensuite optimisées de manière indépendante par les algorithmes. Le temps de calcul pour l'optimisation de chaque tournée est limité à 10 secondes. Les résultats sont affichés dans le tableau III.2. Pour chaque méthode, nous calculons le nombre de fois où la méthode trouve la meilleure solution ($NbMeilleure$) et l'écart moyen ($EMoy$) entre la solution trouvée par la méthode et la meilleure solution.

Nous observons que l'algorithme CDS-V2 est le plus efficace. Ce résultat peut être dû au caractère cyclique des tournées. Nous croyons cependant que la performance de CDS-V2 peut diminuer dans le cas où les contraintes temporelles sont plus restrictives, car CDS-V2 exploite plutôt les aspects physiques du problème au détriment des aspects temporels. L'autre résultat ob-

Instances	LDS		CDS-V1		CDS-V2	
	<i>NbMeilleure</i>	<i>EMoy</i>	<i>NbMeilleure</i>	<i>EMoy</i>	<i>NbMeilleure</i>	<i>EMoy</i>
<i>C1-25</i>	8 (9)	1.65 %	8 (9)	2.12 %	9 (9)	0.00 %
<i>C2-25</i>	2 (8)	15.98 %	4 (8)	7.28 %	6 (8)	12.55 %
<i>R1-25</i>	12 (12)	0.00 %	12 (12)	0.00 %	12 (12)	0.00 %
<i>R2-25</i>	1 (11)	14.99 %	3 (11)	10.20 %	10 (11)	7.79 %
<i>RC1-25</i>	8 (8)	0.00 %	8 (8)	0.00 %	8 (8)	0.00 %
<i>RC2-25</i>	1 (8)	11.87 %	1 (11)	9.58 %	8 (8)	0.00 %
<i>C1-50</i>	8 (9)	2.43 %	8 (9)	1.32 %	9 (9)	0.00 %
<i>C2-50</i>	1 (8)	6.94 %	5 (8)	4.37 %	7 (8)	8.52 %
<i>R1-50</i>	11 (12)	0.78 %	12 (12)	0.00 %	12 (12)	0.00 %
<i>R2-50</i>	0 (11)	9.78 %	3 (11)	3.15 %	9 (11)	0.86 %
<i>RC1-50</i>	7 (8)	0.53 %	7 (8)	0.53 %	8 (8)	0.00 %
<i>RC2-50</i>	0 (8)	5.64 %	3 (8)	4.73 %	7 (8)	2.83 %
<i>C1-100</i>	7 (9)	0.31 %	8 (9)	0.02 %	9 (9)	0.00 %
<i>C2-100</i>	2 (8)	4.18 %	5 (8)	2.77 %	6 (8)	1.97 %
<i>R1-100</i>	8 (12)	0.44 %	10 (12)	0.52 %	12 (12)	0.00 %
<i>R2-100</i>	0 (11)	7.13 %	6 (11)	1.64 %	5 (11)	2.86 %
<i>RC1-100</i>	6 (8)	0.17 %	6 (8)	0.18 %	8 (8)	0.0 %
<i>RC2-100</i>	0 (7)	7.53 %	3 (7)	2.76 %	4 (7)	3.02 %
Total	82 (167)		112 (167)		149 (167)	

TAB. III.2 – Résultats de la comparaison des méthodes pour l’optimisation des tournées

tenu confirme un des résultats du chapitre VI, à savoir que la méthode CDS est plus performante que LDS également pour ce type de problèmes.

Finalement, nous avons évalué l’efficacité de l’algorithme de recherche locale pour l’optimisation de la solution complète (algorithme 4 ou RL+CDS). Les solutions obtenues sont comparées avec les solutions optimales par rapport au critère de minimisation de la distance parcourue. Le temps de calcul est limité à 120 secondes pour les instances avec 25 et 50 clients et à 300 secondes pour les instances avec 100 clients. La colonne *NV* spécifie le nombre moyen de véhicules utilisés, *DIST* représente la distance moyenne parcourue, *EMoy* est l’écart moyen par rapport à la solution optimale ou à la meilleure solution connue (MSC) pour les instances avec 100 clients, *NbMeilleure* représente le nombre de fois où la solution optimale ou la meilleure solution connue est atteinte pour l’algorithme proposé et *TMeilleure* est le temps de calcul moyen pour atteindre la meilleure solution.

Nous pouvons conclure que l’algorithme est assez efficace pour les instances de petite taille. De bonnes solutions sont obtenues assez rapidement, notamment pour les instances de type 1. L’écart entre les solutions proposées par l’algorithme et la solution optimale est en moyenne

Instances	Sol. Opt. ou MSC		RL+CDS				
	<i>NV</i>	<i>DIST</i>	<i>NV</i>	<i>DIST</i>	<i>EMoy</i>	<i>NbMeilleure</i>	<i>TMeilleure</i>
<i>C1-25</i>	3	190.59	3	190.59	0.00 %	9 (9)	0.81
<i>C2-25</i>	2	214.45	1.63	231.89	8.13 %	0 (8)	33.15
<i>R1-25</i>	4.92	463.37	4.92	472.43	1.12 %	3 (12)	2.37
<i>R2-25</i>	2.73	382.15	2.09	411.47	7.62 %	0 (11)	64.59
<i>RC1-25</i>	3.25	350.24	3.25	354.99	9.24 %	7 (8)	0.55
<i>RC2-25</i>	2.88	319.28	2.38	384.5	20.32 %	0 (8)	62.3
<i>C1-50</i>	5	361.69	5	368.99	9.05 %	8 (9)	21.48
<i>C2-50</i>	2.75	339.75	2.13	560.23	69.56 %	0 (8)	65.89
<i>R1-50</i>	7.75	766.13	8.42	821.32	7.02 %	0 (12)	35.2
<i>R2-50</i>	3.82	615.41	2.27	895.87	45.03 %	0 (11)	60.01
<i>RC1-50</i>	6.5	730.31	7.25	807.43	11.83 %	1 (8)	11.07
<i>RC2-50</i>	3.88	571.68	2.5	904.66	58.07 %	0 (8)	55.16
<i>C1-100</i>	10	826.7	10.44	936.18	13.24 %	0 (9)	97.49
<i>R1-100</i>	13.25	1173.61	14.17	1329.17	14.31 %	0 (12)	264.05
<i>RC1-100</i>	11.13	1341.33	13.25	1411.05	5.45 %	0 (8)	128.51

TAB. III.3 – Résultats de la comparaison de l’algorithme d’optimisation d’une solution complète

inférieur à 10 %, sauf pour les instances de la classe RC2-25 et les instances du groupe 2 à 50 clients. Notons que pour ces instances le nombre de véhicules utilisés en moyenne est plus petit que dans les solutions optimales. Il est évident que l’efficacité de l’algorithme s’éloigne des niveaux de performance des métaheuristiques proposées dans la littérature à mesure que le nombre de clients augmente. Toutefois, l’algorithme reste toujours compétitif pour les instances du groupe 1 avec 100 clients (pour les instances du groupe 2 les résultats restent pour l’instant insatisfaisants). Le but de cet algorithme est uniquement de proposer une solution de bonne performance à partir d’une solution initiale, parfois proposée par l’opérateur humain. A des fins de robustesse, il est alors tout à fait souhaitable de conserver au mieux la structure de cette solution initiale.

III.6 Conclusion

Dans ce chapitre nous avons proposé une architecture générique pour un système d’aide à la décision de résolution des problèmes de tournées de véhicules. Nous pensons que cette architecture est bien adaptée aux caractéristiques de ce type de problèmes, particulièrement pour arriver à bien gérer les aspects liés au contexte dynamique du domaine. Un autre avantage de cette architecture est que l’humain est au centre de processus de prise de décisions. En effet, le système permet à

l'opérateur de participer au processus de modélisation et au processus de prise de décisions lors de la résolution du problème, mais aussi de décider quel est le mode de contrôle à utiliser pour chaque étape de la résolution. Les différentes tâches à réaliser ont été identifiées et analysées de manière à pouvoir identifier quels sont les outils (interfaces et algorithmes) et quelles doivent être les caractéristiques de ces outils pour une résolution efficace du problème.

Ensuite, le mécanisme de résolution et les algorithmes intégrés dans le système ont été décrits. Des algorithmes conçus pour aider l'opérateur dans la réalisation de chacune des sous-tâches de la résolution du problème ont été proposés. Le mécanisme de résolution et les algorithmes peuvent être encore enrichis sous l'angle des mécanismes de propagation de contraintes. Ces mécanismes n'ont pas été entièrement développés dans ce travail. Dans cette même logique, d'autres techniques de résolution qui ont été démontrées être bien adaptées aux caractéristiques de certaines des variantes du problème de tournées de véhicules peuvent aussi aider à améliorer ce mécanisme de résolution. Enfin, nous avons également proposé des algorithmes de recherche locale dédiés à l'optimisation d'une solution. Les voisinages explorés lors de la recherche sont définis avec l'idée de ne pas introduire de modification majeure dans la solution initiale. Cette décision est motivée par le fait que les solutions initiales peuvent être proposées par l'opérateur. Plutôt que de proposer une solution avec une meilleure qualité mais que l'opérateur n'arrive pas à rapprocher de la solution initiale qu'il avait proposée, nous avons décidé de préserver au mieux les associations géographiques que l'opérateur humain a pu détecter lors de la proposition de cette solution initiale. Toutefois, il serait aussi intéressant d'intégrer dans le système des algorithmes plus efficaces pour l'optimisation de la solution globale, de manière à proposer à l'opérateur une résolution efficace avec une approche complètement automatisée.

Chapitre IV

Les interfaces homme-machine

IV.1 Introduction

Dans ce chapitre, nous présentons les interfaces homme-machine du système d'aide à la décision que nous proposons pour l'ordonnancement des transports. Pour la conception des interfaces, nous nous sommes basés sur la méthode EID (Ecological Interface Design) (Vicente, 1999a; Vicente and Rasmussen, 1992) qui s'appuie sur la méthode d'analyse du domaine de travail (WDA) présentée dans le paragraphe II.2. Ces interfaces sont souvent reconnues sous le nom d'interfaces écologiques. La principale caractéristique de ces interfaces est que les contraintes et les relations complexes du système de travail sont montrées à l'utilisateur de manière perceptiblement (visible, audible) évidente.

Dans les paragraphes suivants, nous introduisons d'abord des aspects génériques sur l'EID et les interfaces écologiques. Ensuite, nous décrivons de manière détaillée les interfaces proposées. Finalement, nous présentons l'expérience réalisée afin d'évaluer l'efficacité d'une de ces interfaces.

IV.2 Interfaces écologiques

La méthode de conception d'interface écologique ou EID (Ecological Interface Design) a été initialement proposée pour la conception d'interfaces dans les systèmes complexes (Rasmussen and Vicente, 1989; Vicente and Rasmussen, 1992). L'idée est d'étendre les avantages des interfaces

Une partie de ces travaux ont été publiés dans Gacias et al. (2010b) et Cegarra et al. (2010)

de manipulation directe (DMI) (*e.g.*, Shneiderman (1983); Hutchins et al. (1986)) aux systèmes complexes. EID emprunte le terme “écologique” de la psychologie écologique. Les auteurs font une similitude avec ce courant de la psychologie. La psychologie écologique fonde l’analyse de la perception sur les rapports entre l’organisme vivant et son environnement. En même temps, dans le cadre de conception d’interfaces écologiques, les contraintes et les relations de l’environnement de travail dans un système complexe se reflètent perceptiblement à travers l’interface et agissent ainsi sur le comportement humain.

La motivation principale de l’EID est d’arriver à proposer des interfaces capables de fournir un support à l’utilisateur lorsque celui-ci est confronté à un événement qui n’a pas été anticipé par le concepteur. L’interface écologique permet de représenter la hiérarchie d’abstraction comme un modèle mental externe pour la résolution du problème. L’information physique et fonctionnelle du domaine de travail est affichée par l’interface, ce qui permet à l’opérateur humain de facilement percevoir les limites structurelles du domaine. Ceci implique que les ressources cognitives de l’opérateur humain soient dédiées aux processus cognitifs plus complexes tels que la résolution du problème ou la prise de décisions. L’EID est susceptible d’améliorer la performance de l’opérateur et la fiabilité de l’ensemble du système dans la maîtrise des événements qui ont été prévus ou non lors de la conception du système. Cet objectif est atteint par une diminution de la charge mentale nécessaire pour résoudre le problème et en promouvant un raisonnement basé sur les connaissances de l’opérateur humain.

L’EID est basée sur deux concepts clés issus de la recherche en ingénierie cognitive : la hiérarchie d’abstraction et la taxonomie SRK (Skills, Rules, Knowledge). La hiérarchie d’abstraction (voir le chapitre II) est l’outil d’analyse de systèmes de travail qui permet de décrire la complexité (les contraintes, les relations entre objets, les limites structurelles, ...) du domaine de travail. La taxonomie SRK (Rasmussen, 1983) est le cadre théorique sur lequel s’appuie l’EID pour déterminer la manière la plus efficace de communiquer aux opérateurs les informations sur le domaine via une interface homme-machine.

La taxonomie SRK propose une modélisation du traitement de l’information par l’opérateur (Rasmussen, 1986b). Elle définit trois types de comportement : Automatismes, Règles et Connaissances. Le comportement basé sur les automatismes représente un type de comportement qui requiert très peu voire aucun contrôle conscient pour exécuter une action. Ce type de comportement est généralement très rapide et efficace. Le comportement basé sur les règles est caractérisé

par l'utilisation des règles et de procédures pour la sélection d'une séquence d'action dans une situation de travail familière. Ces deux types de comportement peuvent être définis par le couplage perception-action. Finalement, le comportement basé sur les connaissances est un comportement qui se caractérise pour des actions qui dérivent d'une analyse réalisée à partir d'une représentation symbolique du système. Les opérateurs ont besoin de connaître les principes fondamentaux et les lois qui gouvernent le système pour réussir l'analyse, pour arriver ainsi à établir des objectifs explicites qui vont diriger les décisions à prendre. La charge mentale est typiquement plus élevée que dans les comportements basés sur les habiletés ou sur les règles. Ceci augmente le risque de faire des erreurs d'analyse ce qui peut dériver en une mauvaise prise de décisions. Ce comportement du traitement de l'information est habituel lorsque l'humain affronte des situations non attendues et non familières.

L'idée de l'EID est de développer des interfaces qui supportent simultanément les trois niveaux de contrôle cognitif. L'interface écologique devra donc favoriser le couplage entre perception et action de manière à promouvoir un comportement basé sur les habiletés (automatismes). Les interfaces écologiques permettent alors aux utilisateurs novices d'acquérir plus facilement des modèles mentaux qui généralement prennent un certain temps d'expérience et de formation pour se développer (Burns and Hajdukiewicz, 2004). L'interface doit aussi fournir des signes uniques et cohérents avec les contraintes ou les règles du domaine. Ces signes doivent être clairs et univoques (les signes ambigus doivent être supprimés de l'interface) de manière à ce que l'opérateur humain puisse établir une séquence manifeste d'actions à prendre. Enfin, les interfaces écologiques soutiennent aussi le comportement basé sur les connaissances. Pour faire ceci, une représentation du domaine de travail à travers une hiérarchie d'abstraction est fournie par l'interface avec le but de servir à l'opérateur de modèle mental externe pour l'aider dans son analyse. Une fois que les contraintes du domaine ont été identifiées par le WDA, le principe de conception des interfaces est de fournir les informations nécessaires pour la résolution du problème d'une manière directement perceptible. Ce fait favorise un comportement de traitement de l'information basé sur les automatismes (perception-action) qui est plus efficace et qui exige une charge mentale plus faible que le niveau des connaissances.

Dans la littérature, nous trouvons plusieurs travaux expérimentaux avec des interfaces écologiques (Vicente and Rasmussen, 1992; Watanabe et al., 1995; Burns and Hajdukiewicz, 2004; Burns et al., 2008; Blanche, 2010). Certains de ces travaux montrent que les interfaces écologiques

améliorent la performance des opérateurs notamment face à l'imprévu et face à des situations non familières. Dans Burns et al. (2008), les auteurs remarquent cependant que dans certaines situations familières les participants qui utilisent les interfaces écologiques sont moins performants. Ceci renforce l'idée que l'EID ne peut pas remplacer les approches classiques de conception d'interfaces homme-machine tels que la conception centrée utilisateur et l'analyse de la tâche, mais que l'approche permet de les compléter.

IV.3 Les interfaces du système

Dans ce paragraphe, nous présentons les interfaces du système d'aide à la décision pour l'ordonnancement des transports. Ces interfaces ont été conçues en suivant le principe de l'EID (Ecological Interface Design). L'idée est de fournir à l'opérateur humain les informations nécessaires pour faciliter la résolution de la tâche. Ces informations vont être affichées de manière à ce qu'elles soient directement perceptibles par l'opérateur humain. Si l'information fournie à l'opérateur est pertinente, certaines ressources cognitives sont libérées (stockage d'information, calculs mentaux, ...). Ces ressources peuvent ainsi être dédiées à l'analyse du problème et à la prise de décisions.

Nous nous sommes servis de l'identification des tâches réalisée dans le paragraphe III.3 pour définir la structure du système d'aide à la décision. Le menu principal du système d'aide à la décision affiche les tâches principales pour la résolution du problème : la modélisation du problème, la résolution du problème, la sélection de la solution et la gestion en temps-réel des tournées. Ci-dessous nous décrivons les interfaces proposées avec la finalité de fournir une aide à l'opérateur humain pour la réalisation de chacune des tâches.

IV.3.1 Interfaces pour la modélisation du problème

La modélisation du problème est la première tâche à réaliser. Nous proposons deux types d'interfaces pour aider l'opérateur à réaliser cette modélisation.

Le premier des deux types consiste en un ensemble d'interfaces qui permettent de spécifier les objets physiques du problème et l'information importante relative à chacun des objets (caractéristiques physiques et contraintes spécifiques). Une interface est proposée pour chaque type d'objet physique identifié dans la hiérarchie d'abstraction (véhicules, conducteurs, dépôts, marchandises et clients). L'analyse des différentes variantes des problèmes d'ordonnancement des

transports a permis d'identifier quelles sont les caractéristiques et les contraintes des objets qui peuvent avoir une influence sur la résolution du problème. Par exemple, dans le cas des véhicules, comme nous l'avons déjà identifié dans le paragraphe II.3.1, nous considérons comme caractéristiques physiques pertinentes (voir la figure IV.1) : la capacité de transport du véhicule (volume, poids et longueur), la vitesse et le coût d'utilisation du véhicule.

La figure IV.1 montre l'ensemble des interfaces pour chaque type d'objet. Une des caractéristiques des interfaces est leur flexibilité. D'un côté, il n'est pas nécessaire de remplir toute l'information demandée pour continuer la résolution du problème. D'un autre côté, les interfaces proposent des outils d'interaction avec l'opérateur humain. Ces outils permettent d'intégrer d'autres contraintes, non considérées dans un premier temps. Ainsi, par exemple l'opérateur humain a la possibilité de considérer plusieurs fenêtres de temps pour les clients ou des nouvelles contraintes liées aux véhicules comme la durée (ou la distance ou le nombre de clients) maximale de la tournée pour un type de véhicule.

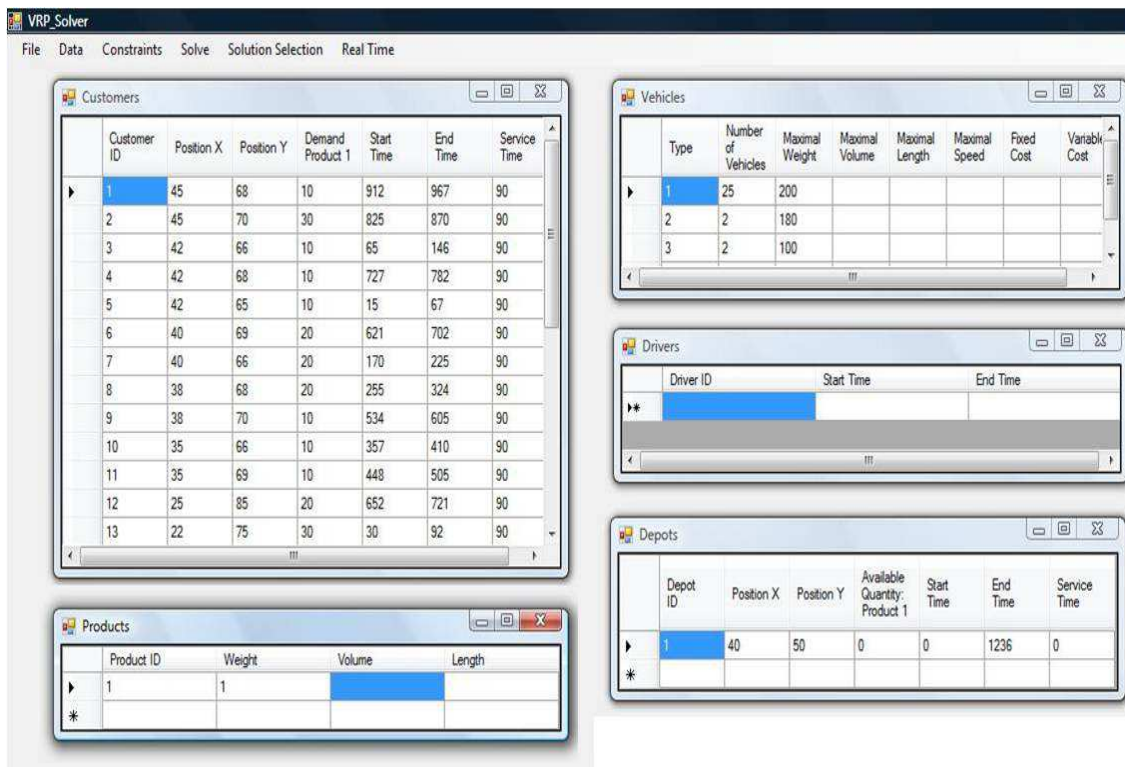


FIG. IV.1 – Ensemble d'interfaces pour la modélisation des objets physiques du problème

La deuxième interface qui fait partie des interfaces homme-machine de support à l'opérateur humain dans la modélisation du problème est affichée dans la figure IV.2. Cette interface assiste l'opérateur pour la spécification des contraintes du problème. La hiérarchie d'abstraction énumère de manière exhaustive les contraintes du problème. Une partie de ces contraintes sont spécifiques aux objets physiques (ces contraintes ont été définies dans les interfaces de la figure IV.1) et l'autre partie des contraintes sont définies entre différents objets physiques (contraintes d'inclusion ou d'exclusion, contraintes de précedence, ...). Ce dernier type de contraintes peut être introduit dans le modèle grâce à l'interface de la figure IV.2.

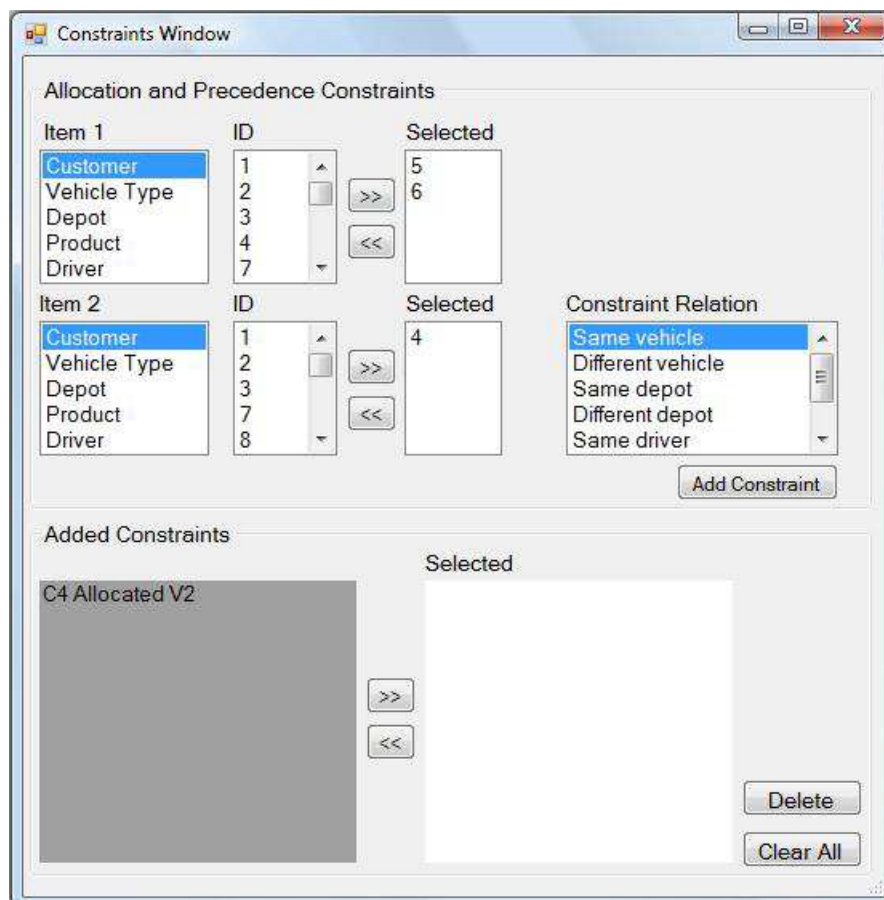


FIG. IV.2 – Interface de support pour la modélisation des contraintes entre les objets physiques

Le fonctionnement de l'interface est le suivant. D'abord, l'opérateur humain sélectionne deux objets ou deux ensembles d'objets (de même nature ou de nature différente). Une fois les ob-

jets sélectionnés, l'interface affiche quel type de contraintes peut être considéré entre ces deux objets ou ensembles d'objets (les types possibles de contraintes entre objets ont été identifiés de manière exhaustive lors de l'énumération des variantes des problèmes de tournées de véhicules). L'opérateur sélectionne la nature de la relation entre les objets et la (les) contrainte(s) définie(s) est (sont) ajoutée(s) au modèle.

Les principaux avantages des interfaces proposées pour la modélisation du système sont :

- L'opérateur manipule de manière directe les objets physiques et les contraintes du système.
- Les contraintes et les caractéristiques des objets sont présentées à l'opérateur avec un langage naturel et compréhensible. Il n'est pas nécessaire d'être un expert pour pouvoir modéliser le système. Le système n'exige pas non plus l'apprentissage d'un langage de modélisation.
- Les interfaces sont flexibles et d'utilisation facile. Les objets et les caractéristiques des objets peuvent être facilement introduits et modifiés. Les contraintes ou ensembles de contraintes peuvent être aussi ajoutées ou supprimées du modèle avec facilité.

IV.3.2 Interfaces pour la résolution du problème

La résolution du problème est la deuxième des tâches principales à réaliser. Cette résolution du problème a été divisée en trois sous-tâches indépendantes : la sélection des véhicules, l'affectation des clients et la conception des itinéraires. Nous pensons en effet que ces trois sous-tâches sont souvent réalisées de manière indépendante par les planificateurs (voir le paragraphe III.3). Le système de travail analysé par de Snoo et al. (2008) est un autre exemple de problème réel qui souligne l'importance d'une sélection de véhicules efficace dans un système d'ordonnement des transports. A notre avis, il est nécessaire que le système d'aide à la décision dispose de modules spécifiques (algorithmes et interfaces dédiés) de manière à aider l'opérateur humain lors de la réalisation de ces sous-tâches.

Interface pour la sélection des véhicules

La sélection des véhicules est une des principales sous-tâches pour la résolution du problème. Cette tâche consiste à déterminer quels sont les véhicules parmi l'ensemble de véhicules disponibles qui vont être utilisés pour servir les clients. Les contraintes pour déterminer la faisabilité du problème ainsi que les algorithmes intégrés dans le mécanisme de résolution sont présentés

dans le paragraphe III.4.2.

L'interface d'interaction homme-machine pour la sélection des véhicules est affichée dans la figure IV.3. L'interface fournit à l'opérateur les informations suivantes sur les véhicules : le nombre de véhicules initialement disponibles, les limites de capacité (poids maximal, volume maximal et longueur maximale) et les coûts fixes et variables des véhicules. L'opérateur dispose aussi d'information sur la satisfaction des contraintes du problème. L'interface affiche une borne inférieure sur le nombre de véhicules minimal nécessaire (voir le paragraphe III.4.2 pour les détails du calcul de la borne). Cette borne est comparée à chaque instant au nombre de véhicules de la solution courante (en bas à gauche sur la figure). Dans le même temps, l'information sur les contraintes liées à la limite de capacité des véhicules (poids et volume) est aussi affichée sur l'interface. L'utilisateur sait donc à chaque instant si la capacité des véhicules qui vont être utilisés est suffisante pour servir les clients et la marge de satisfaction de ces contraintes. Cette information permet à l'opérateur humain de se faire une idée sur la flexibilité apportée pour la solution mise en place.

En ce qui concerne les modes de contrôle, le système, d'une part, permet d'évaluer une solution proposée par l'opérateur humain, et d'autre part, le système calcule et propose des solutions qui satisfont les contraintes considérées tout en minimisant le nombre de véhicules utilisés (avec les algorithmes présentés dans le paragraphe III.4.2). Dans ce dernier cas, le système peut soit proposer un ensemble de solutions complètes et l'opérateur en choisit une, soit compléter une solution qui a été partiellement spécifiée par l'opérateur. L'opérateur peut ainsi participer à la construction de la solution et peut imposer la présence de certains véhicules. Par exemple, dans certaines situations l'opérateur peut préférer un véhicule avec une très grande capacité pour assurer que le service d'un ensemble de clients soit fait avec un seul véhicule.

Les avantages de ces outils d'interaction sont les suivants :

- L'information nécessaire pour trouver une solution réalisable est affichée par l'interface. Ceci permet à l'opérateur de se focaliser sur l'analyse du problème et sur la prise de décisions.
- L'opérateur humain peut participer à la construction de la solution. Il peut ainsi prendre les décisions en se basant sur son expérience et sur des connaissances sur le problème.
- Le système permet d'anticiper des besoins futurs qui n'ont pas été identifiés. Par exemple, l'opérateur peut prévoir une marge pour la satisfaction des contraintes de capacité plus grande, de manière à assurer le service de commandes de dernière minute. Cette caracté-

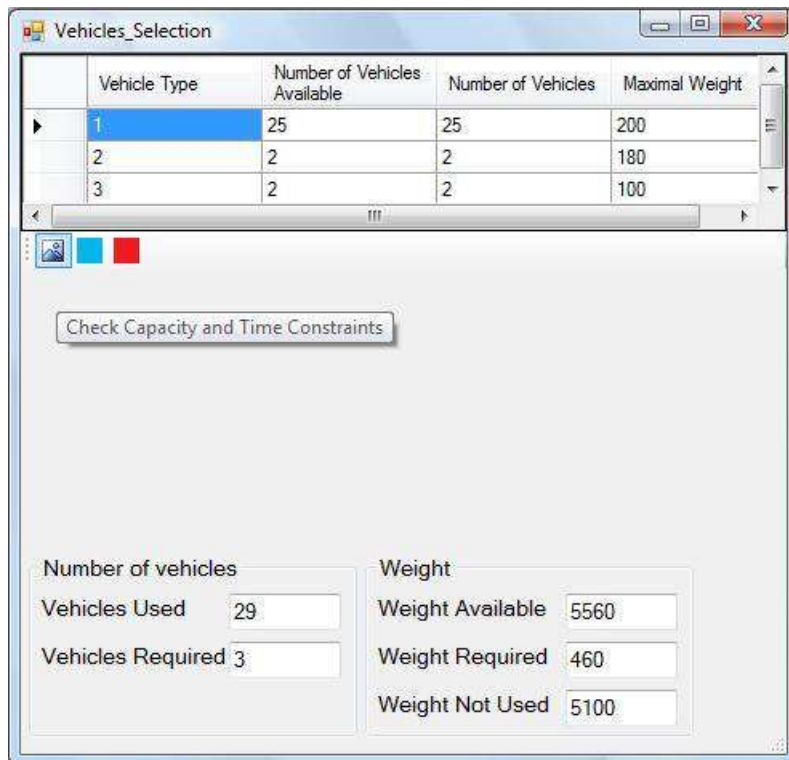


FIG. IV.3 – Interface de support à l’opérateur humain pour la sélection des véhicules

ristique est très importante pour cette phase de résolution du problème. En effet, dans les problèmes réels, la sélection des véhicules est souvent déterminée avant que le problème ne soit entièrement défini (systèmes avec location des véhicules, systèmes avec différentes planifications indépendantes, ...).

Interface pour l’affectation des clients

L’affectation des clients consiste à déterminer les clients qui vont être servis par chaque véhicule. Dans le système d’aide à la décision, la solution proposée dans cette étape de la résolution est toujours réalisable. Ceci implique que toutes les contraintes qui ont été spécifiées lors de la phase de modélisation doivent être satisfaites. Les algorithmes spécifiques pour trouver une solution pour cette phase de résolution sont présentés dans le paragraphe III.4.2.

L’interface a été conçue pour favoriser la participation de l’opérateur à la construction de la solution. Les informations physiques du système identifiées lors de la hiérarchie d’abstraction

(le niveau des objets physiques et le niveau des processus physiques) sont affichées de manière évidente. L'interface est composée de deux parties. Dans la partie de gauche, une carte avec le réseau routier défini par les localisations des clients et des dépôts est affichée. Ormerod and Chronicle (1996) démontrent que l'humain est capable de proposer des solutions très proches des solutions optimales à partir d'un support visuel avec les localisations des clients. L'affichage d'une carte est une pratique très habituelle dans les systèmes d'aide à la décision pour l'ordonnement des transports, notamment dans les systèmes qui permettent un réaménagement des solutions (Tarantilis and Kiranoudis, 2002; Santos et al., 2008; Mendoza et al., 2009; Yu and Liu, 2009). Dans la partie de droite, l'opérateur humain dispose de l'information sur les objets physiques et sur les contraintes du système de travail. Nous trouvons donc des informations sur la capacité (disponible et utilisée) des véhicules, les demandes des clients et les contraintes temporelles spécifiées par les clients (fenêtres de temps, temps de service, ...). Finalement, l'interface affiche aussi une information partielle sur les routes qui vont être créées (clients servis, des bornes simples sur la distance minimale à parcourir et sur la date minimale de fin de la tournée, ...). L'opérateur dispose ainsi de l'information sur les routes sous forme visuelle (carte) et sous forme numérique. Cette information permet, d'une part, d'orienter l'opérateur sur les décisions à prendre et d'autre part, de réduire les processus cognitifs liés au traitement de l'information (stockage de l'information, calcul mental, ...).

L'interface dispose aussi des outils d'interaction nécessaires pour que l'opérateur humain participe à la résolution du problème. Dans ce contexte, l'opérateur a la possibilité d'affecter un client ou un ensemble de clients à un véhicule et de défaire les décisions qui ont été déjà prises. L'opérateur dispose de différents modes de contrôle pour construire une solution. D'abord, l'opérateur peut construire une solution de manière itérative. Après chaque affectation d'un client à un véhicule, le système lance un test pour déterminer la faisabilité de la décision prise et une recherche complète de manière à trouver une solution réalisable avec les décisions qui ont été déjà prises (voir les algorithmes dans le paragraphe III.4.2). L'interface offre la possibilité à l'opérateur de fixer une limite de temps pour cette recherche de la solution réalisable. L'interface laisse aussi la possibilité à l'opérateur d'utiliser des heuristiques pour construire la solution de manière rapide et automatique. Dans cette logique, les algorithmes proposés peuvent être utilisés même après que certaines décisions aient été déjà prises par l'opérateur humain. Ces mécanismes d'interaction permettent à l'opérateur de participer à la construction de la solution de deux manières. D'une

part, l'opérateur humain peut imposer certaines décisions qui n'ont pas été spécifiées lors de la modélisation du problème et, d'autre part, l'opérateur humain peut contribuer à augmenter la qualité de la solution obtenue avec la modification de certaines décisions prises par les algorithmes.

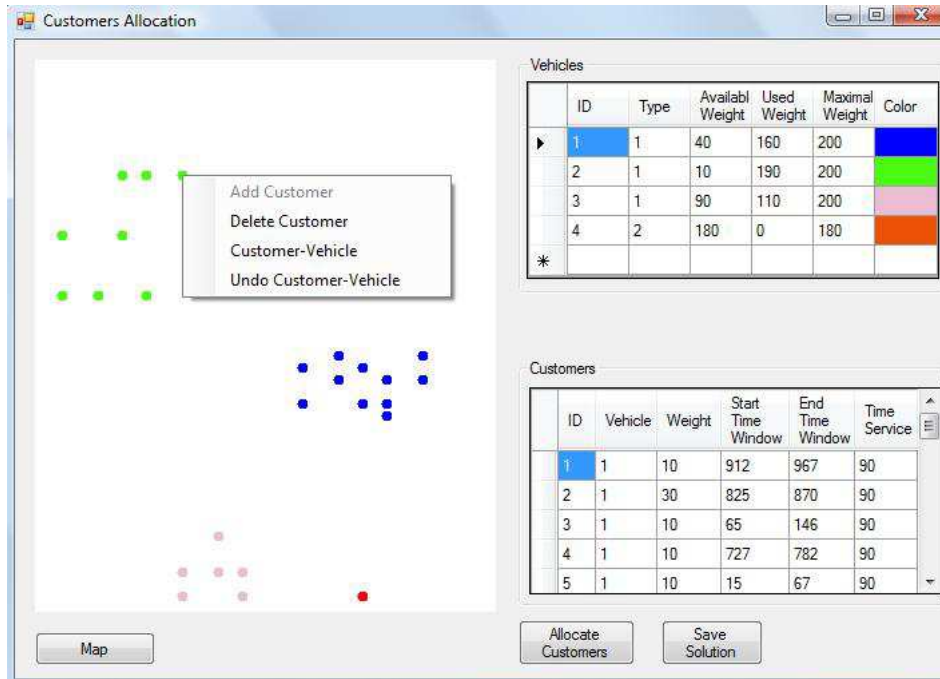


FIG. IV.4 – Interface de support à l'opérateur humain pour l'affectation des clients

Interface pour la construction des itinéraires

Deux types d'interfaces différentes sont proposées pour la construction des itinéraires. La première (voir figure IV.5 a) reprend le même principe de l'interface proposée pour l'affectation des clients. L'objectif de cette étape de la résolution est de déterminer les itinéraires à suivre par les véhicules. Il est alors nécessaire d'établir l'ordre de service des clients pour chaque véhicule. La complexité des décisions à prendre est plus élevée, ce qui implique que l'opérateur humain dispose de plus de mécanismes d'interaction avec le système. Les décisions que peut prendre l'opérateur humain ne sont pas uniquement l'affectation d'un client à un véhicule. Dans cette étape de la résolution, l'opérateur humain peut aussi décider de la séquence de service des clients.

Une autre différence par rapport à l'interface dédiée à l'affectation des clients est que les informations affichées sur les tournées ne sont plus des bornes sur des critères de performances. L'in-

terface affiche l'information réelle sur la tournée à mesure que celle-ci est en train de se profiler. Dans cette même logique, nous nous basons sur le système d'aide à la décision décrit dans Matsatsinis (2004) pour proposer une nouvelle interface qui affiche une représentation temporelle et des informations sur les aspects fonctionnels de chaque tournée (voir la figure IV.5 b). Cette interface facilite la tâche de comparaison de tournées. Nous avons noté dans le paragraphe III.3 que les planificateurs sont souvent confrontés à la tâche de modification des tournées, afin d'équilibrer les tournées pour le plus nombre de critères possibles (distance parcourue, durée de la tournée, nombre de clients servis, temps morts de la tournée, ...). Cette interface fournit à l'opérateur une information complémentaire à l'interface de la figure IV.5 a (interface surtout centrée sur l'information qui concerne les aspects physiques du problème) de sorte que l'opérateur dispose de plus d'éléments pour définir les décisions à prendre.

Cette deuxième interface n'est pas la seule différence avec les outils de support à l'opérateur pour l'affectation des clients aux véhicules. Les trois phases de résolution du problème sont indépendantes. Ceci implique que les algorithmes qui font partie de cette troisième étape doivent être capables de proposer une solution complète et réalisable pour le problème. Ces algorithmes sont présentés dans le paragraphe III.4.2.

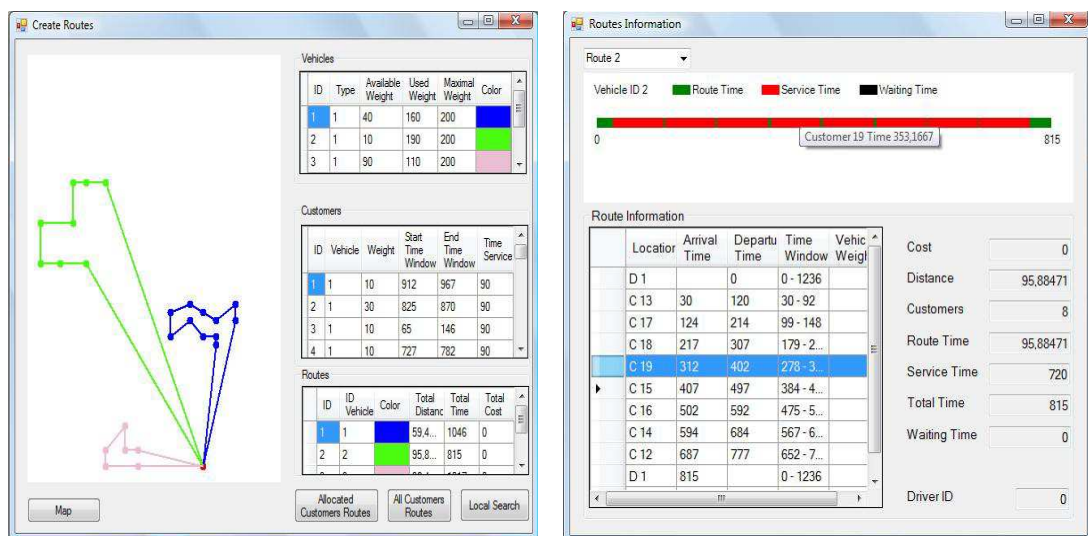


FIG. IV.5 – Interfaces de support à l'opérateur humain pour la construction des itinéraires

Les différents modes, du mode le plus manuel au mode le plus automatique, pour la construc-

tion d'une solution sont listés ci-dessous :

- L'opérateur propose de manière itérative une séquence de clients. Des réaffectations des clients peuvent être aussi faites.
- L'opérateur propose une séquence partielle de clients et un algorithme s'occupe de séquencer les clients qui ont une affectation à un véhicule déjà faite.
- L'opérateur propose une séquence partielle de clients et un algorithme s'occupe de séquencer le reste des clients (avec affectation ou non à un véhicule).
- Un algorithme intégré propose une solution complète au problème.

Finalement, l'opérateur humain dispose aussi d'algorithmes pour l'optimisation des solutions. Le critère à minimiser est la distance parcourue par les tournées. Deux types d'optimisation sont proposées :

- Optimisation de la solution globale : Un algorithme d'optimisation est lancé avec une limite de temps de calcul. Le système propose à l'opérateur la meilleure solution trouvée à la fin du temps limite.
- Optimisation locale des tournées : Un algorithme pour l'optimisation de la tournée sélectionnée est lancé. L'algorithme cherche uniquement à optimiser la séquence des clients de la tournée. Le véhicule affecté à la tournée et les clients qui sont affectés au véhicule ne sont donc pas modifiés.

IV.3.3 Interface pour la sélection de la solution finale

Cette interface a été conçue pour aider l'opérateur lors de la sélection de la solution retenue. Le système permet de garder en mémoire les solutions proposées. Cette interface facilite la comparaison de ces solutions de manière à aider l'opérateur à choisir la solution la mieux adaptée au problème réel. Les problèmes réels d'ordonnancement sont souvent multicritères. Certains de ces critères sont pris en compte par l'opérateur humain ou par les algorithmes lors de la résolution des problèmes. Cependant, il existe toujours des critères ou contraintes qui n'ont pas été identifiés lors de la modélisation du problème. L'idée de cette interface est de pouvoir prendre en compte ces aspects lors du choix de la solution finale.

L'interface représente les multiples solutions du problème dans des graphiques bidimensionnels. Le nombre de graphiques et les axes de chaque graphique sont spécifiés par l'opérateur humain. Les critères d'évaluation des solutions sont spécifiés par l'opérateur humain sur les axes

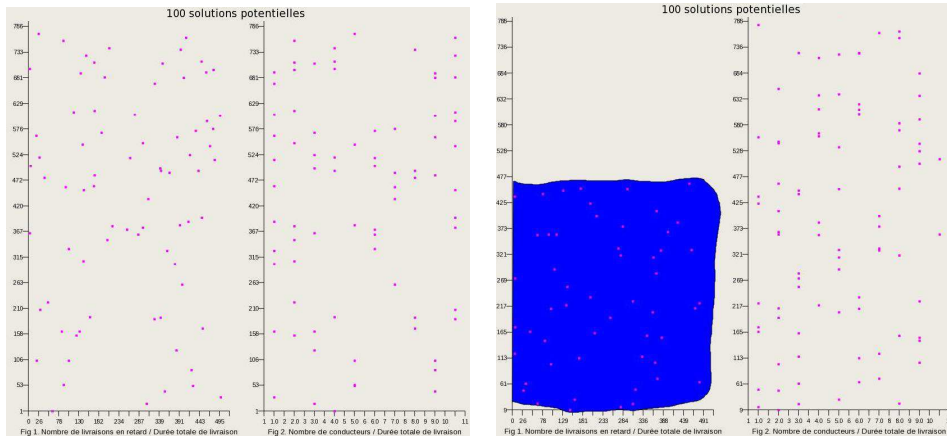
des graphiques. Les solutions sont ainsi évaluées et comparées par rapport aux critères établis par l'opérateur humain. Ceci permet de prendre en compte des critères secondaires ou des préférences de l'opérateur humain. Ce dernier joue un rôle décisif pour les décisions qui concernent les aspects fonctionnels du système de travail. En effet, la tâche principale de l'opérateur humain est plus souvent de choisir la bonne solution par rapport aux critères établis que d'élaborer les solutions possibles. La participation de l'humain dans le processus de sélection de la solution est donc nécessaire en raison du grand nombre de solutions réalisables.

L'interface permet de réduire la charge mentale nécessaire du processus de sélection de la solution. Pour cela, les solutions sont affichées de manière simultanée. Cette nouvelle forme de représentation des solutions encourage à ce que les processus cognitifs liés au traitement de l'information soient fondés plutôt sur la perception que sur l'analyse. Ceci facilite la tâche de comparaison de solutions de manière à arriver à déterminer la solution avec un meilleur compromis par rapport aux critères fixés par l'opérateur.

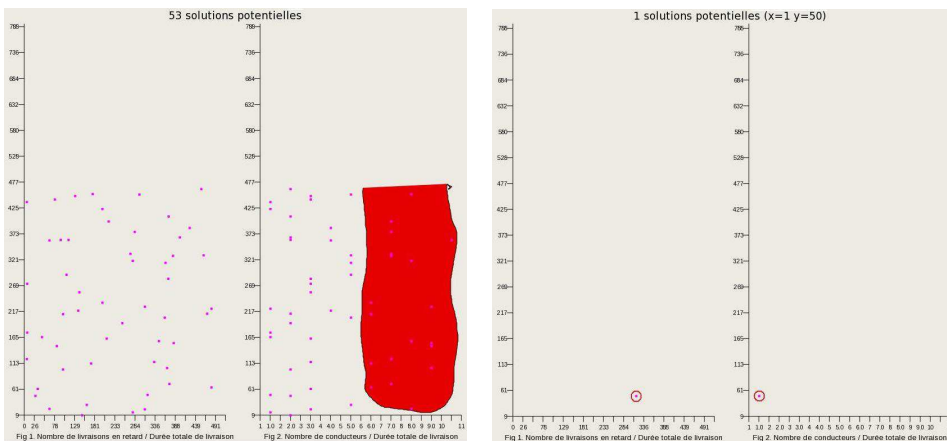
Le processus à suivre pour choisir la solution la mieux adaptée selon les critères de l'opérateur humain est le suivant :

- L'opérateur humain définit les graphiques en associant les critères d'évaluation aux axes. Par exemple, dans la figure IV.6 (a), deux graphiques sont affichés. Les abscisses de chaque graphique représentent respectivement le nombre de services en retard (services où les fenêtres de temps des clients ne sont pas respectées) et le nombre de véhicules utilisés pour chaque solution. Les ordonnées représentent la durée maximale de la solution dans les deux graphiques.
- Les solutions possibles du problème sont représentées par des points dans les différents graphiques. L'opérateur humain peut donc comparer la qualité des solutions pour chaque paire de critères spécifiés.
- L'opérateur humain peut sélectionner (ou rejeter) l'ensemble de solutions désirées (non-désirées) dans chaque graphique. Cet outil permet à l'opérateur de conserver les solutions qui atteignent un certain niveau de performance pour les critères spécifiés dans les graphiques. Par exemple, dans la figure IV.6 (b), les solutions avec une durée plus courte qu'un certain seuil sont considérées comme solutions acceptables. Dans la figure IV.6 (c), les solutions qui utilisent un nombre de véhicules plus élevé qu'un certain seuil sont rejetées par l'opérateur.

- Après chaque décision, les solutions qui ont été considérées comme non-acceptables disparaissent des autres graphiques.
- Le processus de sélection ou de rejet de solutions est répété de manière itérative jusqu'à ce que la solution avec le meilleur compromis pour les critères considérés est atteinte (figure IV.6 (d)).



(a) Graphiques qui affichent les solutions conservées (b) Sélection des solutions intéressantes (surface bleue)



(c) Les solutions moins intéressantes sont rejetées (surface rouge) (d) Graphiques avec la solution la plus intéressante

FIG. IV.6 – Interface dédiée au support à l'opérateur pour la sélection de la solution finale

Pour résumer, les avantages de l'interface pour la sélection de la solution finale du système d'aide à la décision sont :

- L'interface permet que les préférences de l'opérateur humain et certaines contraintes qui

n'ont pas été spécifiées lors de la modélisation et la résolution du problème soient considérées pour le choix de la solution finale.

- La nouvelle représentation des solutions favorise un traitement de l'information fondé sur la perception. De plus, l'interface affiche uniquement l'information qui est intéressante pour l'opérateur. Ceci permet de réduire la charge mentale nécessaire pour la comparaison des solutions par rapport aux interfaces de systèmes traditionnels.
- L'interface permet une manipulation directe des objets (dans ce cas des solutions) ce qui facilite l'interaction entre l'opérateur humain et le système. Il est démontré que cette manipulation directe permet de réduire la charge mentale et le temps nécessaire pour prendre les décisions.
- L'interface a été initialement conçue pour afficher les informations sur les aspects fonctionnels du système de travail (les niveaux les plus hauts de la hiérarchie d'abstraction). Malgré cela, l'interface peut être facilement adaptable pour afficher des informations sur les niveaux inférieurs de la hiérarchie. Il suffit de trouver le moyen de modéliser l'information de manière numérique. En effet, les graphiques permettent d'afficher des informations modélisées par des variables continues (*e.g.*, la durée maximale des tournées), des variables discrètes (*e.g.*, le nombre de véhicules utilisés) et des variables logiques (*e.g.*, si une route spécifique fait partie d'une des tournées de la solution ou non).

IV.3.4 Interface pour la poursuite en temps-réel des tournées

L'interface dédiée au support de l'opérateur lors de la gestion en temps-réel des tournées n'a pas pu être proposée par manque de temps. Nous pouvons toutefois imaginer cette dernière interface comme une extension de l'interface dédiée à la construction des itinéraires (voir la figure IV.5). Les outils d'interaction entre l'opérateur humain et les algorithmes de résolution doivent être cependant modifiés. Ceux-ci doivent en effet être capables d'aider l'opérateur dans le réaménagement des tournées en cas d'incident. Le système de transport décrit dans de Snoo et al. (2008) et les entretiens avec les planificateurs ont permis d'identifier les cas les plus fréquents qui exigent habituellement un réaménagement des tournées :

- L'arrivée d'un nouveau service ;
- L'annulation d'un service programmé ;
- Le retard d'un véhicule ;

- La congestion d'une route ou la présence d'un véhicule en panne.

A notre avis, il est nécessaire que le système dispose des outils d'interaction et des algorithmes spécifiques pour aider l'opérateur à agir en conséquence et pouvoir ainsi proposer une solution efficace lors de l'arrivée d'un de ces incidents.

IV.4 Évaluation de l'interface pour la sélection des solutions

L'interface proposée pour aider l'opérateur dans la tâche de sélection de la solution finale a été testée et comparée à deux autres interfaces plus traditionnelles sur un problème réel d'ordonnement de transports. Dans ce paragraphe, nous décrivons d'abord le problème réel. Ensuite, nous présentons l'algorithme proposé pour la résolution du problème. Enfin, nous présentons l'expérience et les résultats obtenus sur la performance de l'interface proposée.

IV.4.1 Description du problème

Un problème réel d'ordonnement des transports a été considéré pour le contexte de l'étude. Une organisation de producteurs de Seine-et-Marne produit chaque année plusieurs centaines de milliers de tonnes de betterave. La récolte est étalée de juillet à octobre et doit être terminée avant les gelées. Ces betteraves sont nettoyées pour en retirer le sable, la terre et les pierres. En ce qui concerne le sable, il est ensuite déposé aux frais des producteurs dans des décharges pour un coût important. En Seine-et-Marne, dans une logique d'écologie industrielle, l'UPSM s'est mise en relation avec des industries du bâtiment pour leur proposer le sable. Le coût de décharge est alors supprimé et les industries du bâtiment disposent d'une matière première à moindre coût. L'UPSM doit néanmoins organiser les transports de plusieurs véhicules pour transporter à moindre coût ce sable. Le planificateur doit planifier le transport de sable en visant comme objectif principal la minimisation des coûts de transport.

Dans l'expérience, 4550 tonnes de sable sont disponibles dans le dépôt. L'objectif est de livrer avec un coût minimum le maximum de tonnes possibles aux industries de manière à minimiser la quantité de tonnes qui doivent être déchargées au dépôt de collecte de déchets. Une quantité maximale de sable est demandée pour chacune des industries. Différents moyens de transport

sont disponibles : les camions, les péniches et les trains. Nous considérons une capacité limite, un coût de transport par tonne de sable ($\frac{\text{coût}}{\text{km} \times \text{tonne}}$) et une vitesse moyenne pour chaque moyen de transport. Les péniches et les trains sont des moyens de transport avec des itinéraires fixes, c'est-à-dire qu'ils peuvent être utilisés pour réaliser certains trajets. Les localisations d'origine et de fin de trajet possibles sont prédéfinies. De plus, ces deux moyens de transport ont une fréquence de passage journalière.

IV.4.2 Résolution du problème

Le problème que nous venons de décrire peut être modélisé comme un problème de transbordement (voir le paragraphe I.2.1). Il peut être résolu en temps polynomial par un algorithme de calcul d'un flot maxima à coût minimal dans un graphe orienté quelconque.

La figure IV.7 affiche un graphe très similaire au graphe du problème considéré. Dans ce graphe, le dépôt initial (D_i), les industries du bâtiment (I_j) et le dépôt de décharge (D_f) sont modélisés par des sommets. Le sable peut être transporté en camions depuis le dépôt initial D_i vers toutes les destinations. Le coût de transport du camion est de $\frac{c_c}{\text{km} \times \text{tonne}}$. Les localisations d'origine et de fin de trajet pour les itinéraires de la péniche et du train sont également modélisés par des sommets. Dans le graphe, T_1 est le point de départ de la péniche et l'industrie I_2 la localisation de fin de trajet, le coût de transport est de $\frac{c_p}{\text{km} \times \text{tonne}}$. T_2 est le point de départ du train, I_j et D_f les points d'arrivée et le coût de transport est de $\frac{c_t}{\text{km} \times \text{tonne}}$. Initialement, les arcs qui modélisent les possibilités de transport ne sont pas limités en capacité. Un sommet fictif D'_f permet de considérer le coût de décharge au dépôt final ($\frac{pD_f}{\text{tonne}}$). Ceci implique que les industries soient servies avant d'envoyer les tonnes au dépôt de décharge sauf si le coût de transport est très élevé (voir figure IV.7). Un sommet source (S) et un sommet puits (T) sont aussi considérés. Un arc d'une capacité de 4550 et avec un coût de transport nul est créé entre le sommet source et le dépôt de sable. Nous considérons aussi un arc avec un coût de transport nul entre chaque sommet de destination finale pour le sable (industries et dépôt de décharge) et le sommet puits. La capacité des arcs entre les industries et le sommet puits est égale à la quantité de tonnes de sable maximale sollicitée pour chaque industrie (d_{I_j}) et la capacité de l'arc entre le dépôt final et le puits est égale à 4550.

L'algorithme de calcul du flot maximal avec un coût minimum est lancé itérativement pour différentes valeurs de véhicules disponibles. La quantité de véhicules disponibles de chaque type

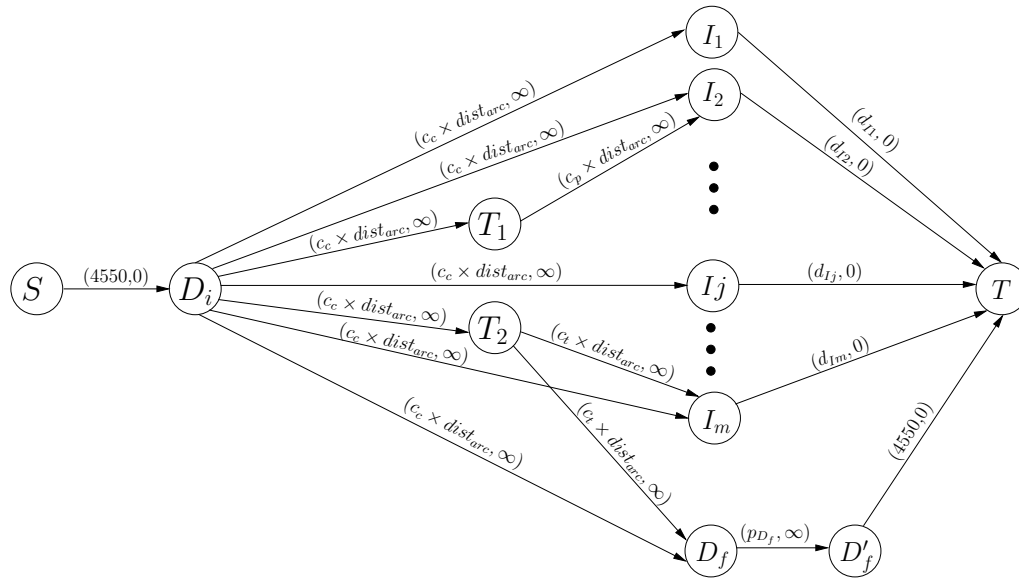


FIG. IV.7 – Graphe équivalent au problème de transbordement de l'expérience

de moyen de transport limite la capacité des arcs. Ceci permet de déterminer un ensemble de solutions réalisables.

IV.4.3 Présentation de l'expérience

Le problème réel d'ordonnement des transports présenté dans les paragraphes précédents précise le contexte de l'expérience. L'objectif de cette expérience est d'évaluer l'efficacité de l'interface proposée pour la sélection de la solution finale.

Les participants de l'expérience sont douze étudiants de Master en Ergonomie et Facteur Humains. Ces étudiants ont travaillé pendant neuf mois avec une entreprise de transports, notamment sur des tâches liées à l'ordonnement des transports. Dans l'expérience, les participants doivent sélectionner la meilleure solution pour chaque scénario. Chaque scénario est résolu avec trois interfaces différentes. Une phase d'entraînement et de familiarisation avec les interfaces a été prévue avant l'expérience.

L'interface présentée dans le paragraphe IV.3.3 est comparée avec une autre interface, plus traditionnelle. La première version de cette dernière interface, que nous appelons par la suite l'interface "analytique", affiche exactement la même information sur les solutions que notre interface (les graphiques disponibles lors de l'expérience sont établis au préalable). Cependant,

ces solutions sont affichées une par une sur l'interface en opposition à notre interface qui présente les solutions de manière simultanée, ceci afin de promouvoir un comportement plutôt fondé sur la perception que sur l'analyse. Pour la deuxième version de cette interface, nous avons ajouté une carte avec les itinéraires des véhicules. Cette interface que nous appelons "intégrale" affiche l'information relative à tous les niveaux de la hiérarchie d'abstraction (l'information fonctionnelle et l'information des niveaux des processus physiques et des objets physiques) en opposition à l'interface que nous avons proposée qui peut être considérée comme une interface abrégée, car elle a été initialement conçue pour afficher uniquement l'information relative aux niveaux supérieurs de la hiérarchie d'abstraction.

Les participants ont résolu neuf scénarios présentés de manière aléatoire. Pour chaque scénario, nous considérons trois paramètres de mesure pour évaluer l'efficacité des interfaces. Le premier est le temps nécessaire pour arriver à sélectionner la meilleure solution réalisable. Ce temps nécessaire pour arriver à prendre une décision est une mesure qui nous renseigne sur les processus cognitifs. Le deuxième paramètre mesuré est lié à la charge mentale nécessaire pour résoudre le problème. Nous avons utilisé le "NASA-TLX" (le National Aeronautics and Space Administration-Task Load Index, Hart and Staveland (1988)) pour évaluer la charge mentale. Le NASA-TLX est basé sur la mesure de six aspects relatifs à la charge mentale : demande cognitive, demande physique, demande temporelle, effort, performance et frustration. Finalement, le troisième paramètre proposé est une mesure liée à la qualité de la solution finale sélectionnée. Nous comparons le nombre de fois où la meilleure solution (il existe une unique solution optimale par scénario) a été sélectionnée par les participants .

Description des scénarios

Nous avons défini neuf scénarios comparables entre eux. Dans tous les scénarios, la solution à moindre coût doit être trouvée. Cependant, les participants doivent prendre en considération la satisfaction d'une contrainte particulière. Dans certains scénarios, une contrainte liée aux niveaux fonctionnels du système de travail (les niveaux supérieurs de la hiérarchie d'abstraction) doit être satisfaite (*e.g.*, la solution ne peut prendre plus de cinq jours). Dans d'autres, une contrainte liée aux niveaux de l'information physique de la hiérarchie d'abstraction doit être respectée (*e.g.*, la circulation est trop dense pour une route spécifique).

Le nombre de solutions pour chaque scénario est égal à vingt-cinq dont uniquement cinq sont

réalisables. Ce nombre a été choisi par rapport à la capacité des planificateurs à comparer des solutions. Le participant doit choisir la solution la moins coûteuse parmi les solutions réalisables. La nature des calculs nécessaires pour chaque scénario est la même, ce qui implique que s'il existe des différences significatives dans les mesures de performance, celles-ci sont dues aux conditions expérimentales (l'interface ou la typologie du scénario).

Hypothèses de l'étude

Pour cette étude, nous faisons les trois hypothèses suivantes :

- (1) Pour tous les scénarios, une interface qui affiche les solutions de manière simultanée est plus efficace (*i.e.*, temps plus court pour prendre les décisions, charge mentale plus faible et meilleure qualité des solutions obtenues) qu'une interface qui affiche les solutions de manière indépendante.
- (2a) Si le planificateur doit considérer les contraintes liées à l'information des niveaux inférieurs de la hiérarchie d'abstraction, une interface qui affiche uniquement l'information liée aux niveaux supérieurs (interface abrégée) est une interface moins efficace que l'interface qui affiche l'information liée aux niveaux supérieurs et inférieurs (interface intégrale).
- (2b) En même temps, si le planificateur doit considérer les contraintes liées à l'information des niveaux supérieurs de la hiérarchie d'abstraction, une interface qui affiche uniquement l'information liée aux niveaux supérieurs (interface abrégée) est une interface plus efficace que l'interface qui affiche l'information liée aux niveaux supérieurs et inférieurs (interface intégrale).

IV.4.4 Résultats expérimentaux

Nous présentons ci-dessous les résultats des comparaisons entre les interfaces. Nous avons calculé l'analyse de variance à mesures répétées ("Repeated Measures ANOVA"). La p -valeur qui détermine si la différence est significative entre deux populations est descendue jusqu'à $p = 0.012$ de manière à contrôler les erreurs de type 1. Par ailleurs, la part de variance expliquée η^2 est un indicateur de la force des différences observées entre la variable indépendante manipulée (par exemple, le type de scénario) et la variable dépendante (par exemple, le temps). Ainsi, une différence entre deux modalités de la variable indépendante peut être significative mais cette différence peut être faible lorsque η^2 est compris entre 0.2 et 0.5, moyenne pour des valeurs entre

0.5 et 0.8 ; au-delà, l'effet est considéré comme important (Cohen, 1992).

Interface analytique *vs.* Interface perceptive

Dans ce paragraphe, l'interface analytique et l'interface considérée comme perceptive sont comparées. Les figures IV.8, IV.9 et IV.10 montrent les résultats des comparaisons pour les différents paramètres de mesure (le temps nécessaire pour sélectionner une solution, l'index NASA-TLX et le paramètre lié à la performance de la solution sélectionnée, respectivement) pour les différents types de scénarios (avec considération d'une contrainte supplémentaire liée aux objectifs ou liée aux moyens du système).

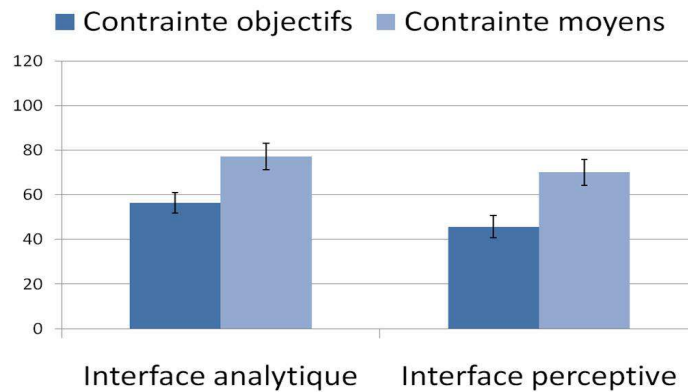


FIG. IV.8 – Comparaison entre l'interface analytique et l'interface perceptive – Résultats du temps (secondes) nécessaire pour la sélection d'une solution

La figure IV.8 montre que le temps nécessaire pour prendre une décision est plus faible avec l'interface perceptive qu'avec l'interface analytique, notamment pour le scénario où la contrainte à satisfaire est liée aux objectifs du système. Cette perception est corroborée par les résultats statistiques qui démontrent qu'il existe une différence significative entre les deux interfaces, indépendamment de la typologie du scénario considéré (résultats du test de Fischer, $[F(1; 23) = 13.104; p = 0.001; \eta^2 = 0.363]$). De plus, il existe également une différence significative si nous analysons les résultats pour les différents types de scénarios, indépendamment des interfaces ($[F(1; 17) = 36.109; p = 0.000; \eta^2 = 0.611]$). Ceci indique que les participants arrivent à prendre une décision plus rapidement lorsque la contrainte est liée aux objectifs que si la contrainte est liée aux aspects physiques du système.

L'analyse de la charge mentale (voir figure IV.9) montre qu'il n'existe pas de différence significative entre les deux interfaces en ce qui concerne la charge mentale nécessaire pour arriver à prendre une décision. En revanche, nous repérons une petite différence en ce qui concerne la charge mentale exigée selon la typologie du scénario ($[F(1; 23) = 19.125; p = 0.000; \eta^2 = 0.454]$). La charge mentale augmente pour le scénario de la contrainte liée aux moyens pour l'interface analytique et elle diminue pour l'interface perceptive.

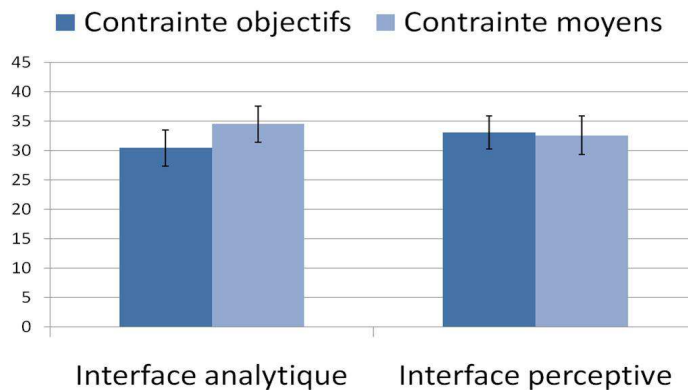


FIG. IV.9 – Comparaison entre l'interface analytique et l'interface perceptive – Résultats de la charge mentale (NASA-TLX) requise pour la sélection d'une solution

Finalement, nous observons que la performance des participants est généralement très bonne (voir la figure IV.10). La meilleure solution est atteinte dans 95 % des cas avec l'interface perceptive. Ce pourcentage est autour de 80 % avec l'interface analytique. L'analyse statistique (nous avons utilisé le test chi-carré qui est plus pertinent pour ce type de variables avec uniquement deux valeurs possibles; 1 si la solution choisie par le participant est la solution optimale et 0 sinon) ne révèle pas de différence significative sur la mesure de performance, ni pour le type d'interface ($[\chi(1) = 0.381; N.S.]$) ni pour la typologie du scénario ($[\chi(1) = 0.000; N.S.]$).

Le premier résultat obtenu suit la logique des hypothèses réalisées sur les interfaces écologiques. Les processus cognitifs fondés sur la perception sont généralement plus rapides et plus efficaces que les processus fondés sur l'analyse. L'hypothèse annoncée tout au long du chapitre postulant que les interfaces écologiques exigent une charge mentale moins élevée que les interfaces traditionnelles n'a pu être vérifiée dans cette expérience. Cependant, il est intéressant de remarquer que l'interface perceptive favorise (la charge mentale est réduite et le pourcentage de réussite est

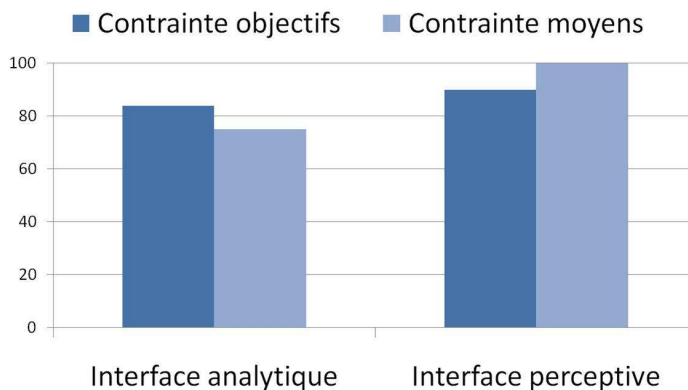


FIG. IV.10 – Comparaison entre l’interface analytique et l’interface perceptive – Résultats de performance (pourcentage de réussite) pour la sélection d’une solution

de 100 %) les processus de prise de décisions pour les scénarios où la contrainte considérée est liée aux moyens (niveaux physiques de la hiérarchie d’abstraction), et ce, bien que l’information nécessaire pour trouver la meilleure solution ne soit pas directement accessible à partir de l’interface. Dans ces situations, les planificateurs doivent faire preuve de leur capacité d’analyse en se basant sur des connaissances sur le problème pour trouver la bonne solution. Les résultats montrent que l’interface perceptive facilite ce processus d’analyse.

Interface intégrale vs. Interface abrégée

Nous comparons dans ce paragraphe l’interface intégrale et l’interface abrégée. Les figures IV.11, IV.12 et IV.13 montrent les résultats des comparaisons pour les différents paramètres de mesure pour les différents types de scénarios (avec considération d’une contrainte supplémentaire liée aux objectifs ou liée aux moyens du système).

La figure IV.11 montre les résultats de la comparaison en ce qui concerne le temps nécessaire pour prendre une décision. Nous observons que le temps exigé pour prendre la décision est plus important avec l’interface intégrale (interface qui affiche l’information relative à tous les niveaux de la hiérarchie d’abstraction) qu’avec l’interface abrégée. L’analyse statistique démontre l’existence d’une différence significative pour les résultats des deux interfaces, indépendamment de la typologie des scénarios ($[F(1; 23) = 27.212; p = 0.000; \eta^2 = 0.546]$). En ce qui concerne la comparaison entre les différents types de scénario, nous ne pouvons pas considérer qu’il existe

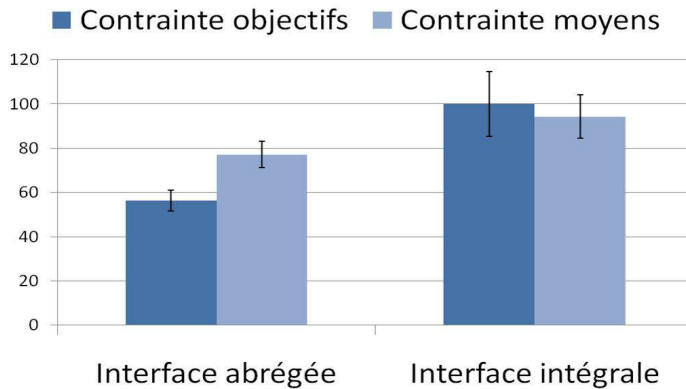


FIG. IV.11 – Comparaison entre l’interface intégrale et l’interface abrégée – Résultats du temps (secondes) nécessaire pour la sélection d’une solution

une différence claire entre les deux types de scénarios.

Dans l’analyse de la charge mentale, nous observons une différence assez importante entre les deux types d’interface (voir figure IV.12). L’analyse statistique renforce l’idée que l’interface intégrale demande une charge mentale plus élevée que l’interface abrégée ($[F(1; 23) = 11.317; p = 0.003; \eta^2 = 0.330]$). La figure IV.12 montre aussi une différence par rapport à la typologie du scénario. Dans l’analyse statistique, nous détectons que la charge mentale demandée aux participants est significativement plus élevée pour résoudre les scénarios où les contraintes à prendre en compte sont liées aux moyens que pour les scénarios avec des contraintes liées aux objectifs ($[F(1; 23) = 8.769; p = 0.007; \eta^2 = 0.276]$).

La figure IV.13 montre le pourcentage de cas où le candidat a trouvé la meilleure solution. A partir de l’analyse statistique, nous pouvons déduire qu’il n’existe pas de différence significative ni entre les interfaces ni entre les différents types de scénarios pour cette mesure.

Les résultats obtenus pour la mesure du temps total nécessaire pour la sélection de la solution sont étonnants. En accord avec les hypothèses de l’EID, l’interface intégrale devrait favoriser la prise de décisions, notamment pour les scénarios où la contrainte est liée aux niveaux inférieurs de la hiérarchie d’abstraction, car cette interface fournit à l’opérateur de manière directe l’information relative à ces niveaux. Pourtant, l’interface demande plus de temps pour sélectionner la solution et avec un pourcentage de réussite plus faible (encore que statistiquement cette différence n’est pas significative).

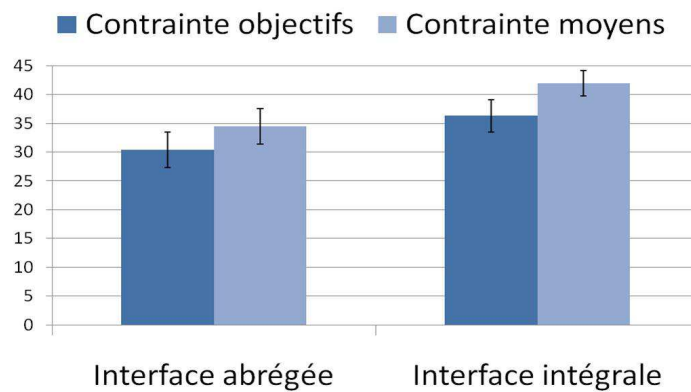


FIG. IV.12 – Comparaison entre l'interface intégrale et l'interface abrégée – Résultats de la charge mentale (NASA-TLX) requise pour la sélection d'une solution

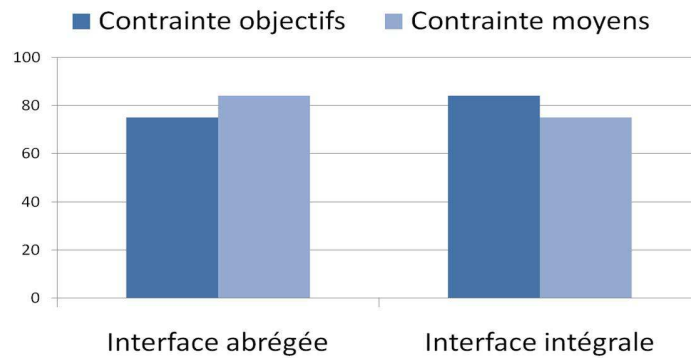


FIG. IV.13 – Comparaison entre l'interface intégrale et l'interface abrégée – Résultats de performance (pourcentage de réussite) pour la sélection d'une solution

Nous trouvons deux explications possibles à ces résultats, la première est qu'il est possible que la manière choisie pour afficher les informations relatives aux niveaux inférieurs de la hiérarchie d'abstraction ne soit pas la plus adéquate. Cependant, Ormerod and Chronicle (1996) montrent que les humains sont capables de déterminer des solutions proches de la solution optimale pour le problème du voyageur de commerce (problème voisin du problème de tournées de véhicules) à partir de la perception visuelle (matérialisée comme dans notre interface par une carte avec les localisations des clients). Ces résultats nous encouragent à continuer le développement de l'interface proposée pour la sélection de véhicules de manière à ce que l'information physique du système de travail puisse aussi être affichée dans les graphiques.

La seconde explication est qu'il est possible que les planificateurs n'aient pas vraiment besoin de l'information sur les niveaux inférieurs pour les processus de comparaison des solutions. Lorsque les niveaux inférieurs sont affichés, il est tout à fait possible que l'excès d'information fasse augmenter la charge mentale et le temps nécessaire pour intégrer toute cette information. Dans la littérature, nous trouvons plusieurs travaux expérimentaux qui montrent que l'information non nécessaire fait augmenter la charge mentale des planificateurs et le temps nécessaire pour prendre les décisions (Moray et al., 1991; Davis and Kottemann, 1995; Cegarra and Hoc, 2008).

IV.5 Conclusion

Dans ce chapitre, les interfaces du système d'aide à la décision pour l'ordonnement des transports ont été présentées. Nous nous sommes basés sur les concepts proposés dans la méthode de conception d'interfaces écologiques présentée dans le paragraphe IV.2. Ce sont des interfaces susceptibles d'aider de manière efficace l'opérateur lors du processus de résolution d'un problème d'ordonnement de transport. Les interfaces permettent une manipulation directe de la part de l'opérateur humain des objets du système.

Dans ce contexte, des interfaces dédiées à la modélisation du problème, à la résolution du problème et à la sélection de la solution finale ont été proposées. Ces tâches, conjointement à la poursuite des tournées en temps-réel, ont été identifiées auparavant comme les grandes tâches devant être accomplies pour arriver à résoudre un problème d'ordonnement de transports (voir le paragraphe III.3). En plus des caractéristiques spécifiques aux interfaces écologiques, les interfaces proposées sont aussi des interfaces flexibles et d'utilisation facile. En effet, les interfaces ne demandent pas une formation complémentaire pour être utilisés et facilitent les processus de prise de décisions.

Finalement, nous avons présenté l'expérience réalisée afin de tester l'efficacité des caractéristiques de l'interface proposée (perceptive et abrégée). Les résultats obtenus montrent l'efficacité de l'interface proposée par rapport à d'autres interfaces plus traditionnelles. Quelques interrogations restent cependant sans réponse. L'inefficacité de l'interface qui contient toute l'information de la hiérarchie d'abstraction n'est pas complètement expliquée. De nouvelles expériences sur ce sujet et sur l'efficacité des autres interfaces proposées seraient nécessaires pour une validation

complète des mécanismes de coopération homme-machine proposés dans ce chapitre.

Chapitre V

Assistance à l'opérateur dans la phase de résolution

V.1 Introduction

Dans ce chapitre, nous présentons les outils du système d'aide à la décision destinés à aider l'opérateur lorsque le problème est non-réalisable ou le devient à la suite de la prise de certaines décisions. Cette question a été très peu abordée dans la littérature. D'abord, nous analysons plus profondément les caractéristiques du problème auquel est confronté l'opérateur humain lorsqu'il n'existe pas de solution réalisable pour le problème traité. Ensuite, nous décrivons l'approche que nous avons adoptée, appelée inversion du modèle, de manière à aider l'opérateur dans la tâche de relaxation des contraintes. Nous proposons également des méthodes de classification et d'analyse de données sur lesquels s'appuient ces mécanismes d'inversion. L'efficacité de ces propositions a été testée sur des instances de la littérature. Les résultats obtenus sont ainsi présentés en fin de chapitre.

V.2 Problèmes non-réalisables

Un problème d'ordonnancement des transports peut être non-réalisable en raison des contraintes spécifiées dans le modèle initial ou peut le devenir au cours de sa résolution. Ceci est

Une partie de ces travaux ont été publiés dans Gacias et al. (2010c)

le cas quand une séquence de décisions (qui correspond à l'instanciation d'un certain nombre de variables de décision) entraîne la non-existence d'une valeur consistante (*i.e.*, qui ne viole aucune des contraintes du problème) pour une des variables qui n'a pas encore été instanciée.

Les différents types de problèmes sont illustrés sur la figure V.1. Il s'agit de réaliser l'affectation de deux clients aux véhicules. Ces derniers ont chacun un poids maximal de transport (100 *kg.* pour le véhicule V_1 et 80 *kg.* pour le véhicule V_2). Le poids de la demande du client C_1 est de 60 *kg.* et de 30 *kg.* pour le client C_2 . Dans le problème de la figure V.1 (a), le client C_1 peut être affecté indifféremment aux deux véhicules, en revanche le client C_2 peut être servi uniquement par le véhicule V_2 . Ce problème est réalisable, car l'affectation du client C_1 au véhicule V_1 et du client C_2 au véhicule V_2 est une solution qui ne viole aucune contrainte du problème. Dans l'exemple de la figure V.1 (b) (où les deux clients doivent être servis par le véhicule V_2), le problème est cependant non-réalisable car le poids limite du véhicule est dépassé. Enfin, la figure V.1 (c) illustre un problème initialement réalisable (le problème de la figure V.1 (a)) qui à cause d'une décision qui a été prise devient non-réalisable. La décision d'affecter le client C_1 au véhicule V_2 entraîne en effet qu'il n'existe plus d'affectation possible pour le client C_2 sans violer une des contraintes du problème.

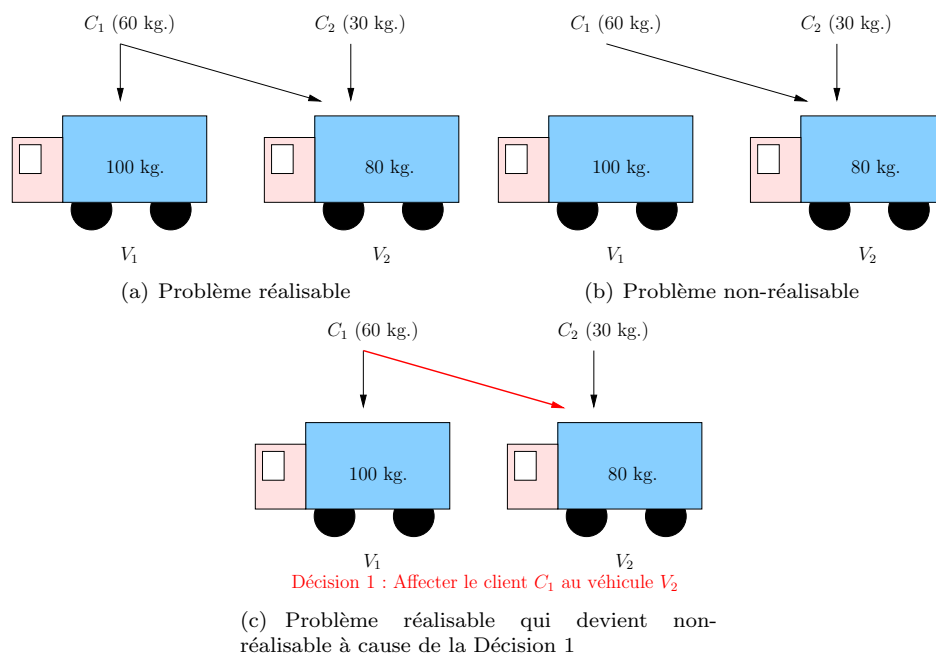


FIG. V.1 – Exemple de différents types de problèmes

D'après Higgins (1996, 1999) ce problème d'infaisabilité peut être surmonté par l'opérateur humain essentiellement de deux manières : avec l'autorisation d'une relaxation forte des contraintes ou avec l'autorisation d'une relaxation légère des contraintes (voir figure V.2).

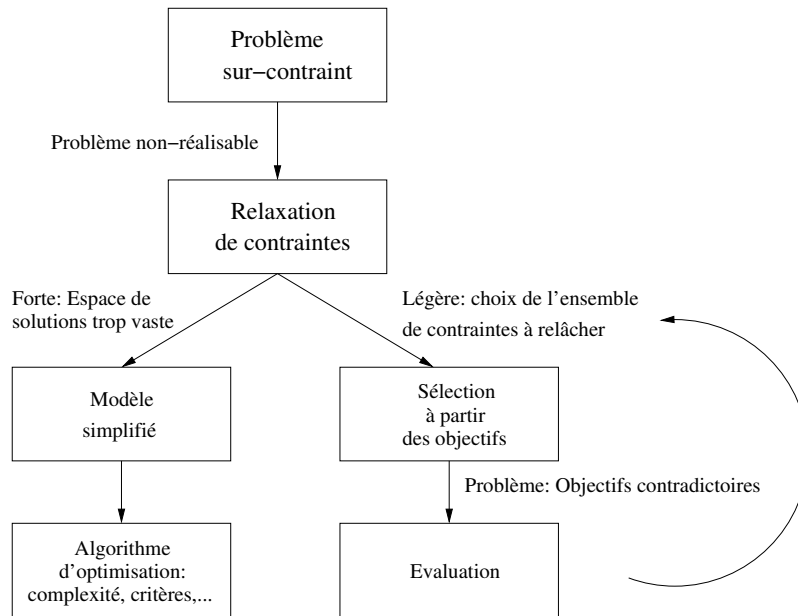


FIG. V.2 – Analyse des options proposées dans Higgins (1996) pour surmonter un problème non-réalisable

Une relaxation forte de contraintes consiste à relâcher un ensemble conséquent de contraintes de manière à élargir sensiblement l'espace de solutions qui au début était vide. Le modèle considéré pour la résolution est ainsi, encore une fois, simplifié. Le principal inconvénient de cette approche est que, malgré l'hypothétique existence d'un algorithme d'optimisation efficace qui permette de résoudre le problème généralement de complexité non polynomiale, la solution proposée risque d'être loin des exigences de l'opérateur. Ceci peut impliquer que l'opérateur humain favorise une approche plus manuelle au détriment du système d'aide à la décision, car l'opérateur est toujours obligé de vérifier et/ou modifier la solution proposée par le système. De plus, la conception de l'algorithme d'optimisation n'est pas évidente. Par exemple, en ce qui concerne le choix du critère à optimiser, dans les méthodes de recherche opérationnelle il est habituel d'inclure des termes liés aux coûts des violations de certaines contraintes dans la fonction objectif. Un des problèmes ici est que ni l'opérateur humain ni le concepteur ne sont

capables de pouvoir spécifier de manière numérique quels sont les coûts associés à la violation d'une contrainte. Par exemple, le retard dans la livraison pour certains clients peut n'avoir aucune conséquence négative et pour d'autres peut, en revanche, impliquer la résiliation du contrat ou induire une mauvaise publicité, situations qui peuvent entraîner des pertes inestimables à l'avance. Nous concluons donc que cette approche est loin d'être envisageable pour un système censé faciliter les tâches liées à la résolution du problème.

L'autre possibilité est une relaxation légère des contraintes qui permet de retrouver la faisabilité d'un petit ensemble de solutions. Cette approche est, selon nous, plus adéquate pour être intégrée dans un système d'aide à la décision. D'après Higgins (1999), l'opérateur réalise un cycle sélection-relaxation-évaluation jusqu'à parvenir à une solution satisfaisante. La programmation par contraintes propose un cadre générique pour résoudre ce type de problème grâce au formalisme des CSP pondérés (weighted Constraint Satisfaction Problem). Dans ce type de problèmes, un ensemble de contraintes qui ne définissent pas des restrictions strictes est considéré. Il s'agit donc de minimiser le nombre de contraintes violées (problème dit "max-CSP"). La principale interrogation de cette approche réside dans le fait de déterminer quel est le plus pertinent choix des contraintes à relâcher. De plus, la complexité de ce type de problèmes est aussi généralement non polynomiale. Jussien (2001) propose des mécanismes algorithmiques basés sur le concept d'"explication" pour déterminer les contraintes ou ensembles de contraintes qui doivent être relâchées afin de rendre le problème réalisable. L'approche reste cependant très algorithmique et difficilement intégrable sous sa forme actuelle dans un système coopératif.

Nous nous sommes inspirés de cette approche et du concept d'inversion du modèle pour la conception d'un mécanisme qui permet de proposer à l'opérateur humain une liste de contraintes à relâcher en priorité pour que le problème devienne réalisable. L'idée est de faciliter à l'opérateur la tâche de sélection des contraintes à relâcher pour ainsi rompre le plus vite possible le cycle sélection-relaxation-évaluation de la figure V.2.

V.3 Inversion de modèle

Nous proposons une inversion de modèle pour offrir des choix à l'utilisateur sur les contraintes à relâcher, lorsque le problème devient non-réalisable en raison de la violation d'une des contraintes. L'inversion de modèle consiste en l'échange des rôles des variables de décision et des paramètres

endogènes (propres à un centre de décision donné) et exogènes (extérieurs à ce centre). Ainsi les variables deviennent paramètres, qui contraignent l'espace de décision, et les paramètres deviennent variables de décision sur lesquels des inférences peuvent être déduites.

Dans ce travail, l'inversion de modèle est proposée pour l'étape de sélection des véhicules. L'identification des paramètres qui sont susceptibles d'être modifiés est assez simple pour cette étape de la résolution du problème, car le nombre de contraintes considérées est très faible (voir le paragraphe III.4.2 pour une identification des contraintes qui sont prises en compte dans cette étape de résolution du problème). Nous rappelons que l'objectif de cette étape est de déterminer les véhicules qui vont être utilisés pour servir les clients.

Pour certaines contraintes, si le nombre de paramètres qui intervient est limité, nous pouvons déterminer la valeur limite de ce paramètre et faire une première proposition à l'opérateur. Lorsqu'il s'agit d'un ensemble de paramètres, nous proposons des techniques de classification de données (centres mobiles, nuées dynamiques) basées sur des critères jugés pertinents, afin d'offrir à l'utilisateur des contraintes à relâcher en priorité.

Le premier pas de la méthode consiste à identifier les paramètres qui interviennent dans chacune des contraintes considérées. Cette étape est réalisée lors de la conception du système. Nous proposons de diviser les paramètres selon l'objet physique d'appartenance (clients, véhicules, conducteurs, dépôts et marchandises). Après la violation d'une contrainte ou d'un ensemble de contraintes, le système d'aide à la décision interroge l'opérateur sur l'ensemble de paramètres qui sont candidats pour être modifiés. Ceci permet de réduire le nombre de paramètres à inverser et le nombre de calculs à réaliser.

Le tableau V.1 est le résultat de cette première phase d'identification des paramètres qui interviennent dans les contraintes considérées pour l'étape de sélection de véhicules. Par exemple, si nous considérons la contrainte liée à la limite de poids transportable pour les véhicules $(\sum_{i=1}^{nc} \sum_{k=1}^{np} d_i^k \times P_k^p \leq \sum_{\tau=1}^{tv} nv_{\tau} \times C_{\tau}^p)$, nous identifions quatre ensembles de paramètres qui interviennent dans la contrainte : la demande de produit pour chaque client (d_i^k), le poids de chaque produit (P_k^p), le nombre de véhicules de chaque classe (nv_{τ}) et le poids limite transportable pour chaque type de véhicule (C_{τ}^p).

En plus des contraintes analysées dans le tableau, il existe d'autres contraintes qui sont aussi considérées dans cette étape de résolution. Ce sont les contraintes entre les objets physiques. Pour chaque contrainte liée à un type de véhicule, il faut déterminer quels sont les paramètres

Contraintes						
	Contraintes Capacité 1. Poids (eq. III.4) 2. Volume (eq. III.5) 3. Longueur (eq. III.6)	Borne FTC (BI_{nv}^{FTC})	Borne FTD (BI_{nv}^{FTDe} , eq. III.11)	Borne Dis- tance (BI_{nv}^D , eq. III.8)	Borne Temps (BI_{nv}^T , eq. III.9)	Borne Clients (BI_{nv}^C , eq. III.10)
Clients	Demande (d_i^k) ou la suppression du service	Fenêtres de temps (FTC_i) ou la suppression du service	Fenêtres de temps (FTC_i) ou temps de service ou la suppression du service	Suppression du service	Temps de service (Ts_i) ou la suppression du service	Suppression du service
Véhicules	Poids, volume et longueur limites 1. C_τ^p 2. C_τ^v 3. C_τ^l ou le nombre de véhicules (nv_τ)	Nombre de véhicules (nv_τ)	Vitesse moyenne ou le nombre de véhicules (nv_τ)	Distance max. (V_j^D) ou le nombre de véhicules (nv_τ)	Durée max. (V_j^T) ou vitesse moyenne ou le nombre de véhicules (nv_τ)	Nombre de clients max. (V_j^C) ou le nombre de véhicules (nv_τ)
Dépôts			Fenêtres de temps ($FTDe_t$)			
Marchandises	Poids, volume et longueur 1. P_k^p 2. P_k^v 3. P_k^l					

TAB. V.1 – Identification des paramètres qui interviennent dans les contraintes considérées dans la phase de sélection de véhicules

qui peuvent être modifiés afin d'avoir un problème réalisable. Par exemple, pour les contraintes d'affectation entre un client et un type de véhicule particulier, uniquement trois solutions sont possibles : (1) supprimer la contrainte du problème, (2) ajouter un véhicule du type demandé à la solution, ou (3) supprimer le client du problème.

Une fois que les paramètres qui interviennent dans les contraintes ont été identifiés, il faut proposer les algorithmes et/ou les calculs nécessaires pour l'inversion de modèle. Ces mécanismes seront déclenchés après la violation d'une des contraintes. Nous observons qu'un traitement individualisé pour chaque type de contrainte s'impose. Chaque contrainte dispose ainsi de sa propre procédure d'inversion. De plus, un algorithme dédié est nécessaire pour déterminer la modifi-

cation de chaque paramètre ou ensemble de paramètres qui interviennent dans la contrainte. Ce mécanisme d'inversion calcule pour chaque paramètre la valeur limite pour que le problème devienne réalisable. Dans le cas où un ensemble de paramètres peut être modifiés, une liste des contraintes à relâcher en priorité est proposée. Le but est d'essayer de minimiser l'impact de la relaxation et d'essayer de proposer des solutions qui facilitent la résolution du problème, ainsi que l'atteinte des objectifs du système de travail.

Dans ce travail, les algorithmes spécifiques de chacune des contraintes ne sont pas présentés. L'objectif de ce chapitre est d'introduire le cadre de l'inversion de modèle et de proposer des méthodes génériques basés sur des techniques de classification et d'analyse de données qui peuvent être intégrées dans ces mécanismes d'inversion. Bien que ces méthodes de classification ne puissent pas être utilisées pour la plupart des contraintes telles qu'elles sont présentées dans ce chapitre, elles peuvent cependant être combinées avec d'autres algorithmes de manière à s'adapter au mieux à la contrainte violée.

Exemple pour illustrer le fonctionnement du mécanisme d'inversion de modèle

Les 7 clients C_i de la figure V.3 ont demandé à être servis le même jour. 3 véhicules avec une grande capacité de transport sont disponibles. Il existe cependant un contrainte sur la durée maximale des tournées : cette durée ne peut pas dépasser un certain seuil V^T (durée de la journée de travail).

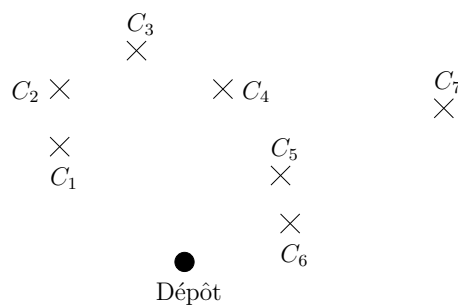


FIG. V.3 – Localisation géographique des clients à servir

Lors de la phase de la sélection des véhicules à utiliser, l'opérateur propose une solution avec uniquement deux véhicules, car il observe que les besoins de capacité sont largement couverts avec ces deux véhicules. Le mécanisme de vérification intégré dans le système d'aide à la décision

détecte que la contrainte liée à la durée maximale des véhicules est violée. Le rapport entre la borne inférieure BI_T du temps nécessaire pour servir les clients (temps de trajet + temps de service) et le nombre de véhicules de la solution est alors plus grand que le seuil V^T ($\frac{BI_T}{2} > V^T$).

Le mécanisme d'inversion de la contrainte violée entre en fonctionnement. D'abord, l'opérateur peut sélectionner le type de paramètres qu'il préfère modifier. Dans notre exemple, nous observons que l'opérateur dispose uniquement de deux choix pour agir sur les paramètres : les paramètres liés aux véhicules ou les paramètres liés aux clients (voir dans le tableau V.1 la colonne de la contrainte liée à la borne de la durée maximale BI_{nv}^T qui est la contrainte violée dans l'exemple).

Dans le cas où l'opérateur choisit de faire varier les paramètres liés aux véhicules, le mécanisme d'inversion propose trois solutions :

1. Augmenter le nombre de véhicules à utiliser jusqu'à $x = \lceil \frac{BI_T}{V^T} \rceil$.
2. Augmenter la vitesse moyenne des véhicules de v_m jusqu'à v_m^2 de manière à réduire la borne pour le temps de trajet pour que $\frac{BI_T^2}{2} < V^T$.
3. Augmenter le seuil fixé pour la durée maximale des tournées (V_2^T) jusqu'à $V_2^T = \frac{BI_T}{2}$.

Dans le cas où l'opérateur choisit de faire varier les paramètres liés aux clients, le mécanisme d'inversion propose deux autres solutions :

4. Réduire le temps de service chez les clients afin que $\frac{BI_T^3}{2} < V^T$.
5. Supprimer un ou plusieurs clients du problème pour que $\frac{BI_T^4}{2} < V^T$.

Nous observons que, bien que les propositions de relaxation dans cette étape de la résolution n'assurent pas la faisabilité du problème, le mécanisme peut cependant orienter efficacement l'opérateur vers les bons paramètres à modifier et comment les modifier. Le mécanisme cherche à minimiser l'impact de la relaxation sur le problème. Ce critère peut parfois impliquer que la proposition de relaxation ne soit pas suffisante, mais elle assure que dans l'étape de résolution courante la contrainte considérée n'est plus violée. Dans les autres étapes de la résolution où les décisions prises sont toutes réalisables, ce phénomène d'infaisabilité du problème malgré la relaxation n'apparaît pas, c'est-à-dire, que les propositions des mécanismes d'inversion assurent que le conflit est résolu.

Le deuxième aspect que nous remarquons est qu'il manque les algorithmes pour définir quelles sont les contraintes à relâcher lorsque l'inversion doit être faite sur un ensemble de paramètres, comme c'est le cas dans la proposition numéros 4 et 5. Par exemple, comment le système d'aide à

la décision peut aider l'opérateur à sélectionner efficacement les clients qui doivent être supprimés du problème ? ou comment le système peut proposer une solution sur les clients sur lesquels une réduction du temps de service doit être appliquée de manière à ce que cette réduction soit efficace pour arriver à atteindre le but qu'aucune des deux tournées considérées de manière individuelle ne dépasse la durée maximale fixée ?

Dans notre exemple, nous supposons maintenant que les temps de service sont très semblables pour tous les clients. Nous pouvons pressentir que les clients C_1 , C_2 , C_3 et C_4 sont de bons candidats pour une réduction du temps de service (proposition 4 du mécanisme d'inversion), car ces clients ont une forte probabilité d'être servis par le véhicule qui aura plus de difficultés pour satisfaire la contrainte liée à la durée maximale. Ce véhicule doit servir quatre clients, alors que l'autre véhicule ne doit en servir que trois. En revanche, si nous considérons la possibilité de supprimer un des clients du problème, les clients C_7 ou C_4 sont de bons candidats, car ils sont les clients les plus éloignés de leurs voisins respectifs. Supprimer ces clients du problème devrait normalement faciliter la satisfaction des contraintes temporelles du problème, en tout état de cause davantage que supprimer d'autres clients comme les clients C_1 ou C_5 .

Nous proposons donc un mécanisme d'inversion divisé en deux phases lorsqu'un ensemble de paramètres sont candidats à être modifiés. D'abord, un algorithme adapté à la contrainte violée doit identifier les paramètres qui permettent d'arriver à obtenir une solution réalisable. Une fois ces paramètres identifiés, un algorithme dédié permet de proposer une liste qui ordonne selon des critères jugés pertinents les meilleurs choix pour la relaxation. Nous proposons ci-dessous des méthodes d'analyse de données qui peuvent servir de base pour la conception de ces algorithmes dédiés.

Enfin, nous remarquons que le mécanisme d'inversion des contraintes est organisé à partir de la structure de la hiérarchie d'abstraction. L'opérateur identifie d'abord les contraintes qui sont candidates à être relâchées ; ces contraintes comme nous l'avons vu lors de l'analyse du domaine sont liées aux objets physiques du système de travail. Ensuite, l'opérateur et le système coopèrent pour déterminer la meilleure manière de relâcher ces contraintes vis-à-vis des critères d'évaluation identifiés dans la hiérarchie d'abstraction (gestion efficace du temps et de la capacité).

V.3.1 Méthodes de classification de données

L'analyse de données propose un ensemble de méthodes destinées à structurer l'information contenue dans les données pour ainsi arriver à identifier des relations existant entre individus. Nous nous intéressons particulièrement aux méthodes de classification dont l'objectif est de construire des groupes d'individus homogènes.

Les méthodes de classification utilisent une mesure de ressemblance pour construire des classes d'individus les plus homogènes possibles. Une mesure de ressemblance est une application qui fait correspondre pour toute paire d'individus un réel positif quantifiant la ressemblance.

Nous présentons dans les paragraphes qui suivent deux méthodes de classification couramment rencontrées : la méthode des centres mobiles et la méthode des nuées dynamiques. Ces méthodes sont utilisées pour regrouper les clients selon un critère géographique et un critère temporel, respectivement. Lorsqu'une contrainte est violée, nous allons nous appuyer sur une analyse des groupes des clients spécifiés par les méthodes de classification pour déterminer quelles sont les contraintes les plus pertinentes à relâcher.

Notons par Ω l'ensemble de n individus à classifier. Chaque individu $\omega_i \in \Omega$ est caractérisé par un ensemble de v variables $z_i = (z_i^1 \dots z_i^v)$.

Critère géographique : centres mobiles

Nous proposons la méthode des centres mobiles pour classifier les clients selon un critère géographique. La méthode des centres mobiles a été mise en œuvre sous sa forme actuelle par Forgy (1965). La condition pour l'application de la méthode est que les ressemblances entre les paires d'individus $(\omega_i, \omega_j) \in \Omega$ puissent être quantifiées par une distance euclidienne $D(\omega_i, \omega_j)$. Soit E l'espace euclidien v -dimensionnel. Les individus $\omega_i \in \Omega$ sont alors vus comme points de E caractérisés par v coordonnées. Nous pouvons alors calculer le centre de gravité G et l'inertie Φ de Ω .

L'inertie totale est constante. Lorsque l'ensemble d'individus est décomposé en classes, cette inertie se décompose additivement selon l'inertie interclasse (calculée entre les classes et le centre de gravité de Ω) et l'inertie intraclasse (calculée entre les individus et le centre de gravité de la classe). Il est démontré que l'algorithme des centres mobiles converge en minimisant localement l'inertie intraclasse pour un nombre de classes fixé, ce qui revient à former les classes en moyenne

les plus homogènes possibles (voir *e.g.*, Saporta (1990)).

Nous proposons l'algorithme des k -means (Lloyd, 1982) pour classifier les nc clients en K clusters ($P = P_1 \cup P_2 \cup \dots \cup P_K$) les plus homogènes possibles par rapport à une mesure de similarité définie pour chaque paire de clients. Nous proposons la localisation géographique des clients comme critère de ressemblance. La condition pour appliquer la méthode des centres mobiles est ainsi satisfaite, car la distance euclidienne entre deux clients est par définition une mesure de dissimilarité. Les deux variables qui caractérisent les individus sont alors l'ordonnée x_i et l'abscisse y_i de la localisation de chaque client. Le nombre de groupes de clients (K) est le nombre de véhicules de la solution non-réalisable.

L'équation de la mesure de dissimilarité correspond donc à la distance entre deux individus (clients) :

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (\text{V.1})$$

L'algorithme des k -means est un algorithme itératif qui permet, dans notre cas, de grouper géographiquement les clients (voir l'algorithme 5). Le premier pas consiste à choisir les K centres initiaux. Dans la méthode originale, les centres sont choisis au hasard et l'algorithme est lancé plusieurs fois avec des ensembles de centres différents. L'objectif est d'arriver à identifier des regroupements d'individus qui restent constants, indépendamment du choix des centres initiaux. Pour ce problème, nous avons opté pour la proposition de centres uniformément répartis entre les clients de manière à favoriser une bonne formation des clusters depuis le premier et seul lancement de l'algorithme. L'analyse des groupes qui permet de déterminer les clients les moins homogènes des clusters est néanmoins faite dans une étape ultérieure. Les centres sont alors distribués de manière uniforme le long d'un des axes (horizontal ou vertical). La direction de la distance la plus longue entre deux clients détermine la direction de distribution des centres et le point d'origine de la distribution est le centre de gravité G de l'ensemble des clients. Ensuite, pour chaque itération de l'algorithme la distance entre chaque client et les centres est calculée. Puis, les clients sont affectés au cluster du centre le plus proche. Enfin, les centres sont remplacés par les centres des nouveaux clusters qui viennent d'être formés. La condition d'arrêt de l'algorithme est que les clusters restent inchangés pour deux itérations successives.

Une des astuces que nous avons utilisée est l'actualisation des centres de chaque cluster après chaque affectation d'un client. Ceci permet d'accélérer la convergence de l'algorithme. La

Algorithme 5: Algorithme de centres mobiles pour grouper les clients géographiquement

Pas 1 : Choisir l'ensemble de K centres initiaux

répéter

 Pas 2 : Calculer les distances entre chaque client et les centres (equation V.1)

 Pas 3 : Affecter chaque client au cluster du centre qui est le plus proche

 Pas 4 : Actualiser les centres de chaque cluster P_k

jusqu'à les clusters restent inchangés d'une itération à une autre;

figure V.4 (a) illustre la position des centres initiaux (deux centres car le nombre de véhicules de la solution proposée est égal à deux) distribués autour du centre de gravité le long de l'axe le plus long selon la position des clients (dans l'exemple l'axe horizontal) et la procédure itérative de l'algorithme pour l'exemple de la figure V.3.

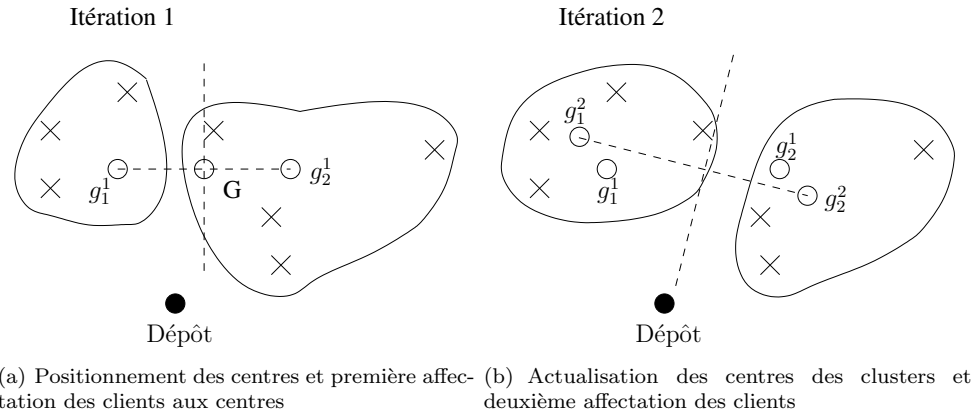


FIG. V.4 – Illustration du fonctionnement de l'algorithme des centres mobiles pour la classification géographique des clients

Une fois que les clients ont été groupés, une analyse de ces groupes est proposée de manière à déterminer quelles sont les meilleures propositions de modification des paramètres.

Les clients qui font partie du même cluster ont une grande probabilité de faire partie de la même route. Pour chaque client, nous proposons le calcul d'un paramètre qui nous donne une idée de l'homogénéité du client par rapport à son groupe. Ce paramètre peut être utilisé aussi comme indicateur pour décider par exemple quel est le meilleur client à supprimer du problème (la suppression d'un client du problème est la seule proposition qui peut être une solution pour toutes les contraintes considérées dans la phase de sélection de véhicules, voir le tableau V.1). Le paramètre proposé (dm_i) est la moyenne des distances entre le client i et les autres clients qui font partie du même cluster ($i \in P_k$, $|P_k|$ est le nombre de clients qui font partie du cluster

k) et la distance entre le client et le dépôt le plus proche de manière à favoriser les clients qui se trouvent près du dépôt.

$$dm_i = \frac{\min_{l \in Depot} d_{il} + \sum_{j \in P_k} d_{ij}}{|P_k|} \quad (V.2)$$

Critère temporel : nuées dynamiques

Le critère temporel pour classifier les clients est fondé sur l’algorithme des nuées dynamiques (Diday, 1971). L’algorithme est une extension de l’algorithme des centres mobiles de manière à pouvoir traiter des mesures de ressemblance qui ne peuvent pas être quantifiées par des distances euclidiennes. Une fonction d’affectation (une mesure de dissimilarité) et une fonction de représentation des clusters doivent donc être définies, car les individus ne peuvent plus être représentés comme des points dans un espace euclidien et les centres de gravité ne peuvent donc être calculés.

Nous utilisons la mesure de dissimilarité proposée par Levy (1996). Elle est calculée à partir des fenêtres de temps des clients ($FT_i = [r_i, d_i]$) et permet de mesurer le degré de centrage entre deux fenêtres de temps. La figure V.5 montre les principales relations entre deux fenêtres de temps et la valeur de la mesure de dissimilarité δ pour chaque configuration.

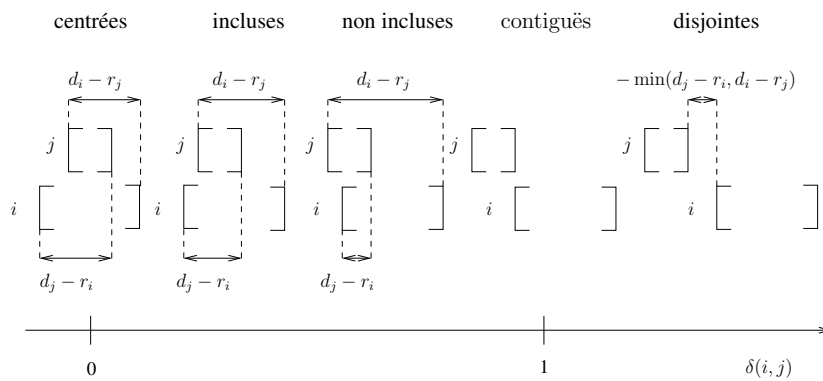


FIG. V.5 – Principales relations entre deux fenêtres de temps

L'équation de la mesure de dissimilarité δ est :

$$\delta(i, j) = \begin{cases} 1 - \frac{\min(d_j - r_i, d_i - r_j)}{\max(d_j - r_i, d_i - r_j)} & \text{si } \min(d_i, d_j) \geq \max(r_i, r_j), \\ 1 - \frac{\min(d_j - r_i, d_i - r_j)}{\frac{1}{n} \sum_{i=1}^n (d_i - r_i)} & \text{sinon.} \end{cases} \quad (\text{V.3})$$

Lorsque les fenêtres temporelles des tâches sont non disjointes, l'indice de dissimilarité mesure la position relative entre les deux fenêtres par rapport à leurs centres. Ainsi, δ s'annule lorsque les fenêtres sont centrées. Il augmente progressivement à mesure que les centres des deux fenêtres s'écartent.

Lorsque les fenêtres de temps sont disjointes, la valeur de δ est supérieure à l'unité. Le terme du diviseur est constant et représente la moyenne des largeurs des fenêtres. δ mesure la dissimilarité de deux tâches de manière proportionnelle à l'intervalle de temps qui sépare les deux fenêtres. Dans le cas limite, la date de fin d'une des fenêtres coïncide avec le début de l'autre (fenêtres contiguës), δ est alors égal à l'unité. Il existe donc une continuité des valeurs de δ .

Une fois l'indice de dissimilarité défini, nous avons besoin d'une fonction d'affectation des clients aux clusters et d'une fonction de représentation des clusters. Nous définissons le centre ou noyau d'un cluster P_k comme le client C_k avec la date de début au plus tôt r_k qui s'approche le plus de la moitié de l'intervalle de temps $[\min_{j \in P_k} r_j, \max_{j \in P_k} r_j]$. A chaque itération de l'algorithme, les clients C_i sont affectés au cluster P_k dont le client-centre (C_k) minimise la valeur de δ . Comme dans l'algorithme des centres mobiles, nous devons calculer à chaque itération les distances entre les fenêtres de temps des clients et les clients-centres de chaque cluster ($\min_{k=1..K} \delta(C_k, C_i)$), puis affecter les clients au cluster avec le client-centre le plus proche (plus petite valeur de δ) et finalement actualiser les centres à partir des clusters qui viennent d'être définis.

Pour l'initialisation des K centres, nous proposons une distribution tout le long de l'intervalle de temps défini par les dates de début au plus tôt des clients (voir l'algorithme 6).

Algorithme 6: Initialisation des centres des clusters pour l'application de l'algorithme des nuées dynamiques

Pas 1 : Calculer l'horizon de temps HT , $HT \leftarrow \max_{i=1..nc} r_i - \min_{i=1..nc} r_i$

Pas 2 : Calculer le pas , $pas \leftarrow \frac{HT}{K}$

Pas 3 : Calculer la position du premier centre, $t_1 \leftarrow \min_{i=1..nc} r_i + \frac{pas}{2}$

Pas 4 : Déterminer le client-centre du premier cluster C_1 , $C_1 \leftarrow$ client i avec la r_i la plus proche de t_1

Pas 5 : Initialiser le compteur des clusters, $k \leftarrow 2$

pour ($k \leq K$) **faire**

 Pas 5 : Calculer la position du centre du cluster k , $t_k \leftarrow t_k + pas$

 Pas 6 : Déterminer le client-centre du cluster k , $C_k \leftarrow$ client i avec r_i le plus proche de t_k et le client n'est pas déjà un centre

 Pas 7 : Incrémenter le compteur des clusters, $k \leftarrow k + 1$

Une fois que les clients ont été groupés par rapport à leur localisation temporelle, nous proposons une analyse des clusters de manière à identifier quels sont les clients les plus conflictuels. Ceci permet de proposer des choix de relaxation qui promettent de faciliter la résolution du problème. Pour chaque cluster, nous calculons un paramètre qui permet de mesurer le niveau de criticité des clients qui font partie du cluster. L'indice de criticité IC_k est calculé comme le rapport entre le nombre de clients du cluster et le nombre de véhicules de la solution. Une grande valeur de IC_k peut alors indiquer des problèmes pour servir les clients du cluster en raison du compactage des fenêtres de temps des clients. Cet indice peut être considéré pour l'utilisation de l'algorithme dédié à proposer des clients à supprimer du problème.

Nous observons que les temps de trajet entre clients, paramètres qui peuvent avoir une influence importante, ne sont pas considérés dans le critère temporel. Nous proposons la classification temporelle et l'indice de criticité pour pointer l'ensemble ou ensembles de clients qui peuvent être conflictuels. Une fois les ensembles de clients conflictuels identifiés, nous proposons l'utilisation de l'algorithme des centres mobiles (algorithme 5) pour chacun de ces sous-ensembles de clients. L'algorithme 7 illustre la procédure suivie pour la proposition d'un client à supprimer. Cette articulation avec le critère géographique permet de prendre en compte aussi les temps de trajet à travers la considération de la distance entre clients.

Enfin, nous devons décider de la valeur de K (le nombre de groupes de clients à définir avec l'algorithme). Pour le critère géographique, cette valeur est assez intuitive ; il s'agit du nombre de véhicules de la solution. De cette manière, chaque groupe de clients peut être vu comme une future tournée. Dans le cas du critère temporel, le choix du nombre de groupes est moins évident.

Algorithme 7: Identification d'un client à supprimer du problème pour le critère temporel

Pas 1 : Initialisation des clients-centres des nuées dynamiques (algorithme 6)

répéter

- Pas 2 : Calculer la valeur de δ entre chaque client et les clients-centres
- Pas 3 : Affecter chaque client au cluster du client-centre avec la plus petite valeur de δ
- Pas 4 : Actualiser les client-centres de chaque cluster P_k

jusqu'à *les clusters restent inchangés d'une itération à une autre;*

Pas 5 : Identifier les clusters de clients conflictuels ($IC_k > \text{seuil}$)

pour *chaque cluster conflictuel* **faire**

- Pas 6 : Application de l'algorithme des centres mobiles pour les classifier géographiquement (algorithme 5)
- Pas 7 : Introduire le client du cluster conflictuel avec une plus grande valeur de dm_i dans une liste L

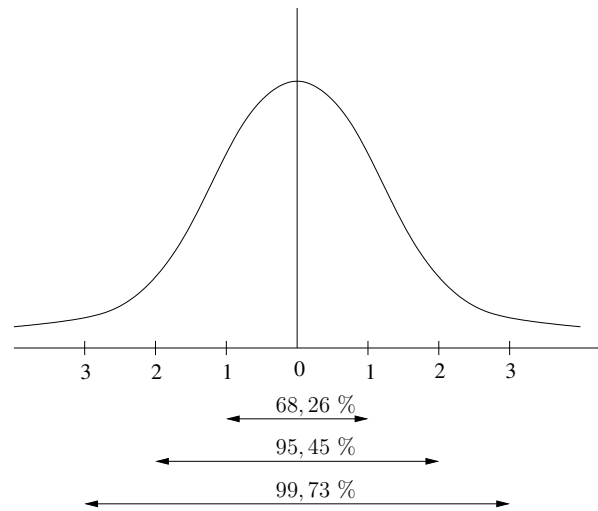
Pas 8 : Proposer le client de la liste L avec une plus grande valeur de dm_i

Dans les tests que nous avons réalisés, nous avons choisi $K = \frac{d_{max} - r_{min}}{\frac{1}{nc} \sum_i (d_i - r_i)}$. Remarquons que si un seul groupe est considéré ($K = 1$), nous revenons à l'application du critère géographique car l'algorithme des centres mobiles est appliqué uniquement à un ensemble qui comprend tous les clients.

Mesure d'évaluation de la proposition du mécanisme d'inversion

Nous proposons dans ce paragraphe une analyse statistique de manière à déterminer le degré de pertinence du choix proposé par les algorithmes de classification. Cette analyse donne comme résultat un paramètre susceptible d'aider l'opérateur lors de la sélection de la contrainte à relâcher. Ce paramètre peut être vu comme une mesure de la qualité de la proposition.

Nous faisons l'hypothèse que la variable de décision X pour chaque critère suit une distribution de probabilité normale $N(\mu, \sigma)$. Cette distribution peut être approchée par $N(\bar{X}, S)$, où \bar{X} et $S = \sqrt{\frac{(X - \bar{X})^2}{n-1}}$ représentent respectivement la moyenne et la déviation de l'échantillon. La distribution est ensuite normalisée ($N(0, 1)$). Pour cela, nous utilisons la fonction $z = \frac{X - \bar{X}}{S}$. La distribution normale standard $N(0, 1)$ ou distribution z permet de comparer les mesures qui ont déterminé le choix proposé à l'opérateur pour un même critère et aussi entre différents critères. Par exemple, une proposition qui a été faite à partir d'un paramètre (par exemple la distance d_i) qui, après la conversion, présente une valeur de z proche de zéro, indique que la décision n'est pas pertinente du point de vue statistique, car la valeur du paramètre n'est pas statistiquement différente des autres (la valeur est placée au centre de la courbe de probabilité, voir la figure V.6).

FIG. V.6 – Distribution normale standard ou distribution z

Le principal avantage de l'utilisation de ce paramètre est que l'opérateur peut avoir une idée de la qualité de la proposition réalisée par le mécanisme d'inversion de modèle par rapport aux autres propositions.

V.4 Résultats expérimentaux

Les algorithmes de classification que nous venons de présenter doivent être adaptés à la contrainte qui a été violée et au type de paramètre qui est inversé. Cependant, dans l'état actuel des développements, ces algorithmes ne peuvent être utilisés que pour déterminer le meilleur client à supprimer du problème lorsque la contrainte liée à la distance ou à la durée maximale des tournées est violée.

L'efficacité de ces algorithmes est testée dans ce paragraphe. Pour cela, nous proposons de résoudre optimalement des instances de petite taille (9 clients et 1 seul dépôt) du problème de tournées de véhicules avec fenêtres de temps (VRPTW). Une énumération complète des solutions possibles est faite. L'idée est d'évaluer la performance des algorithmes proposés à l'heure de choisir le client à supprimer du problème. Chaque instance est donc résolue 9 fois ; à chaque fois, un client différent est supprimé du problème.

Nous comparons les solutions des critères proposés (le critère géographique CG , le critère temporel CT et un troisième critère qui consiste à supprimer le client qui se trouve le plus loin du dépôt DD) avec la solution optimale. La fonction objectif à minimiser est la minimisation de la distance parcourue. CS (Client Sélectionné) correspond à la proposition des trois critères qui présente une valeur de z la plus éloignée du zéro après l'analyse statistique, normalement la meilleure des propositions. Cette analyse n'est pas très pertinente pour les petites instances, car le nombre d'individus de l'échantillonnage est très faible.

Le critère géographique propose comme client à supprimer du problème, le client avec une plus grande valeur de dm_i après application de l'algorithme des centres mobiles. En ce qui concerne l'application du critère temporel, nous considérons que les ensembles de clients conflictuels sont les groupes avec un indice de criticité plus grand que 2 après l'application de l'algorithme des nuées dynamiques. Puis nous appliquons l'algorithme des centres mobiles uniquement aux sous-ensembles conflictuels ($IC_k = \frac{|P_k|}{nv} \geq 2$). Enfin, le client avec une plus grande valeur de dm_i est proposé pour suppression du problème.

Les instances ont été générées à partir des instances de petite taille (25 clients) proposées par Solomon (1983) pour le problème de tournées de véhicules avec fenêtre de temps (VRPTW). Des clients ont été supprimés (pour arriver à des instances de seulement 9 clients) selon l'ordre croissant des dates de début de livraison r_i . Les résultats sont affichés dans le tableau V.4. Dans les instances de Solomon, nous trouvons trois types d'instances selon la position relative entre les clients : instances avec clients groupés en clusters (C), instances avec clients placés aléatoirement (R) et les instances mixtes (RC).

La première colonne ($NbOptDist$) du tableau correspond au nombre de fois où la solution avec une distance minimale est trouvée lorsque le client proposé par un des trois critères est supprimé du problème. Nous indiquons entre parenthèses le nombre d'instances de chaque classe. La deuxième colonne ($DevMoy$) spécifie la moyenne de la déviation par rapport à la solution optimale de la meilleure solution trouvée après avoir supprimé le client proposé par chacun des critères. $MoyPos$ affiche la moyenne de la position de la proposition par rapport à la meilleure solution. Par exemple, si la meilleure solution trouvée, lorsqu'on suit la décision d'un des critères, est la deuxième meilleure solution par rapport aux solutions trouvées, alors la proposition est classée en deuxième position. Enfin, $NbSol$ représente le nombre de fois où la proposition faite par le critère permet de trouver un plus grand nombre de solutions réalisables. Cette mesure

peut être vue comme un indicateur de la flexibilité apportée par la proposition du critère. Cette flexibilité est importante car le problème original est non-réalisable et ainsi l'espace de solutions réalisables est élargi.

56 instances				
$nc = 9$	$NbOptDist$	$DevMoy$	$MoyPos$	$NbSol$
<i>CG-C</i>	13 (17)	8.73 %	2.7	6
<i>CG-R</i>	9 (23)	5.99 %	3.5	17
<i>CG-RC</i>	9 (16)	5.01 %	3.2	5
<i>CT-C</i>	5 (17)	14.98 %	4.5	7
<i>CT-R</i>	3 (23)	11.46 %	5.4	7
<i>CT-RC</i>	4 (16)	9.30 %	5.3	7
<i>DD-C</i>	8 (17)	15.76 %	4.3	8
<i>DD-R</i>	5 (23)	10.97 %	4.7	11
<i>DD-RC</i>	3 (16)	9.12 %	4.5	4
<i>CS</i>	32	6.83 %	3.54	30
Total (3 critères)	37			38

TAB. V.2 – Résultats de la comparaison des propositions des différents critères

Nous pouvons déduire que le critère *CG* est le plus efficace pour les petites instances. La solution optimale est atteinte pour plus de la moitié des instances. La position moyenne est aussi assez bonne (environ 3), ce qui veut dire que la proposition de ce critère reste toujours acceptable pour les instances dont elle ne propose pas la meilleure décision. Ces résultats sont assez logiques car la fonction objectif à minimiser (la distance parcourue) est liée à un critère géographique. Nous remarquons cependant que les résultats sont plus performants que pour le critère de la plus grande distance par rapport au dépôt, bien que ce critère soit aussi un critère géographique.

Nous observons également dans l'analyse statistique que ce critère paraît efficace. Il est capable de choisir la meilleure proposition de suppression de client entre les trois critères de manière à faire augmenter le nombre de fois où la meilleure solution est atteinte. Nous devons toutefois rester méfiants face à ses résultats car l'analyse statistique est faite sur un très petit échantillon d'individus.

Enfin, nous remarquons dans la dernière ligne du tableau la grande performance des trois critères considérés ensemble. La meilleure solution est atteinte sur 37 des 56 instances pour un des trois critères. La proposition qui permet d'obtenir le plus de solutions réalisables est atteinte aussi pour un grand nombre d'instances (la différence entre les nombres de solutions réalisables d'une proposition à un autre peut être de l'ordre de dizaine de milliers). Finalement,

nous avons observé que pour pratiquement toute la totalité des instances (48 sur 56), une des trois propositions permet d'atteindre la solution optimale ou le plus grand nombre de solutions réalisables.

V.5 Conclusion

Dans ce chapitre, nous avons décrit les principes d'un cadre nouveau et original pour aider l'opérateur humain à réaliser la relaxation de contraintes lorsque le problème est ou devient non-réalisable. Nous nous sommes basés sur le concept d'inversion de modèle. L'idée est que chaque contrainte dispose d'un mécanisme d'inversion qui permet de proposer à l'opérateur des paramètres à modifier pour que le problème devienne réalisable.

Dans ce contexte, nous avons proposé des méthodes génériques issues de l'analyse de données qui peuvent être intégrés dans ces mécanismes d'inversion. Les algorithmes doivent cependant être adaptées à la contrainte violée et au type de paramètre considéré pour être modifié. Par exemple, dans le cas où la contrainte violée est liée à la limite du poids transportable par les véhicules, l'algorithme des centres mobiles peut être aussi utilisé pour classifier les clients. Les propriétés des clients à considérer pour la classification devraient cependant être modifiées. Nous pouvons considérer, en plus des coordonnées géographiques, une troisième caractéristique pour la classification des clients, par exemple le poids de leurs demandes. De cette manière, les clients seraient classifiés en groupes les plus homogènes possibles par rapport à la localisation géographique mais aussi par rapport au poids de leurs demandes. Après la classification, une analyse sur les groupes permettrait d'identifier les clients les plus conflictuels (ensembles de clients avec une grande demande et qui sont proches géographiquement) pour, par exemple, les supprimer du problème.

La performance des algorithmes a été testé sur des instances de petite taille. Les résultats obtenus sont très encourageants. Une validation sur des instances de taille plus conséquente doit être cependant réalisée. Nous remarquons aussi la nécessité de la réalisation d'une expérimentation de manière à évaluer l'efficacité de la coopération homme-machine, car l'inversion de modèle a été conçue dans le but principal d'aider l'opérateur humain à relâcher les contraintes du problème (tâche qui doit être impérativement réalisée par l'humain et pour laquelle il n'existe pas, à l'heure actuelle, d'outils de support).

Chapitre VI

Résolution d'un problème d'ordonnancement

VI.1 Introduction

Dans ce chapitre, nous présentons une étude sur un problème particulier d'ordonnancement : le problème à machines parallèles avec contraintes de précédence et temps de préparation entre tâches. Ce problème peut être plus particulièrement vu comme un problème d'ordonnancement lié à l'optimisation de la production. Cela dit, les restrictions spécifiques considérées dans ce problème (contraintes de précédence et temps de préparation) sont tout à fait cohérentes avec les contingences des problèmes de transport.

On peut donc voir l'étude focalisée consignée dans ce chapitre comme une contribution pour la résolution de certains sous-problèmes du problème global d'ordonnancement des transports traité dans les chapitres précédents.

VI.2 Problèmes d'ordonnancement

Les problèmes d'ordonnancement se posent dans des domaines aussi variés que la gestion de production et les transports, l'exploitation de systèmes informatiques ou la conduite de projets. Ils consistent à coordonner l'exécution de tâches ou opérations soumises à des contraintes de

Une partie de ces travaux ont été publiés dans Gacias et al. (2010a)

temps et de ressources, de manière à réaliser un travail global en satisfaisant un ou plusieurs objectifs.

Dans ce qui suit, nous emploierons le terme générique de tâche, bien que le terme d'opération est généralement utilisé en ordonnancement de la production. Le problème peut donc se poser comme suit : étant donné un horizon de temps, un ensemble de ressources ainsi que leur disponibilité sur l'horizon, un ensemble de tâches et un ensemble de contraintes de précédence entre ces tâches, il s'agit de déterminer les dates de début et de fin de chaque tâche, ainsi que la ressource réalisant son exécution.

De nombreux travaux de recherche ont été consacrés à la résolution des problèmes d'ordonnancement. Classiquement, on recherche une solution exacte ou approchée au problème d'optimisation combinatoire ainsi formulé (Baker, 1974). La complexité du problème à traiter (NP-difficulté) incite souvent à l'utilisation de méthodes approchées (heuristiques) au détriment de méthodes exactes, garantissant l'optimalité des solutions trouvées mais au prix d'un temps prohibitif dans un contexte réaliste (taille des problèmes, "richesse" des contraintes réelles). Dans cette étude, nous allons nous intéresser aux méthodes de recherche arborescente et locale pour résoudre le problème de façon efficace.

VI.2.1 Définitions

Précisons la définition des principaux concepts des problèmes d'ordonnancement.

Les tâches

Les tâches sont les entités élémentaires du problème. Elles sont localisées dans le temps par une date de début et nécessitent pour leur réalisation un certain nombre d'unités de temps (durée de la tâche) et de ressources. Selon les problèmes, les tâches peuvent être exécutées par morceaux (problèmes préemptifs) ou doivent être exécutées sans interruption (problèmes non préemptifs).

Les ressources

Une ressource est un moyen, humain ou technique, de disponibilité (limitée ou non) connue a priori. On distingue parmi les ressources à disponibilité limitée les ressources renouvelables et les ressources consommables. Une ressource est renouvelable si elle est à nouveau disponible en même quantité après avoir été allouée à une ou plusieurs tâches, dans le cas contraire elle

est consommable. A titre d'exemple, les machines sont des ressources renouvelables, les matières premières des ressources consommables.

La fonction objectif

La fonction objectif est l'outil dont on dispose pour quantifier la qualité d'une solution. Plusieurs critères peuvent définir cette fonction, comme des critères liés au temps (temps total de l'ordonnancement, retards des tâches lorsque des dates limites sont fixées pour leur exécution, ...), des critères liés aux coûts de production ou liés aux ressources (l'équilibrage de la production dans les ressources, ...). Notons que certains problèmes de décision ne sont pas associés à l'optimisation d'une fonction objectif, mais consistent à prouver l'existence d'une solution afin de démontrer l'admissibilité (respect des contraintes) du problème ; on parle alors de problème de satisfaction de contraintes (*Constraint Satisfaction Problem*, CSP).

Les différents types de contraintes

- Contraintes temporelles

Le problème présente par exemple ce type de contraintes lorsque les tâches sont liées par des contraintes de précédence. Ainsi, si la tâche i précède la tâche j , on ne peut commencer l'exécution de j avant la fin de l'exécution de la tâche i . Ces contraintes de précédence sont ce qu'on appelle des contraintes de potentiels dans le sens où elles établissent des relations entre dates de début (ou de fin) de tâches.

Pour les problèmes n'intégrant que des contraintes de potentiels, le calcul des dates de début au plus tôt et d'un ordonnancement de durée minimale est de résolution aisée, par l'utilisation de méthodes comme PERT (Program Evaluation and Review Technique) ou MPM (Méthode des Potentiels-Metra). En termes mathématiques, il s'agit de calculer le plus long chemin dans le graphe de contraintes de précédence. Ce type de problème peut être résolu avec un algorithme de complexité polynomiale (algorithme de Bellman-Ford).

- Contraintes sur les ressources

Les contraintes sur les ressources sont de nature différente :

- Contraintes d'allocation de ressources

Une ressource disponible doit être allouée à chaque tâche à exécuter.

- Contraintes de partage de ressources disjonctives

Une ressource disjonctive, une machine, ne peut exécuter deux tâches en parallèle. Dans le problème que nous traitons dans le cadre de ce chapitre, nous allons nous intéresser particulièrement aux problèmes présentant de telles contraintes de disjonction, ainsi que des temps de préparation entre l'exécution de deux tâches.

- Contraintes de partage de ressources cumulatives

Dans le cas où les ressources sont disponibles en quantité limitée, le nombre de ressources en train d'exécuter une tâche à un instant donné ne doit pas être supérieur au nombre de ressources disponibles. Ce type de contraintes n'est pas considéré dans le problème traité.

VI.3 Problème à machines parallèles avec contraintes de précedence et temps de préparation

Le problème étudié est l'ordonnancement de tâches sur machines parallèles avec contraintes de précedence et temps de préparation des machines entre l'exécution des différentes tâches. Nous considérons l'optimisation de deux critères différents : la minimisation de la somme des dates de fin de chaque tâche et la minimisation du retard algébrique maximal sur l'ensemble des tâches. Ces fonctions objectif présentent un intérêt particulier en ordonnancement de la production. La somme des dates de fin est un objectif qui maximise le débit de production et qui permet de minimiser les stocks des produits intermédiaires ou d'en-cours, produits non terminés, présents dans l'atelier. Dans la minimisation du retard algébrique maximal, les dates d'échéance peuvent correspondre aux dates de livraison des produits. C'est donc un objectif de respect des délais qui vise à pénaliser le moins possible le client livré le plus en retard. Ces problèmes sont *NP-difficiles* au sens fort (Graham et al., 1979).

Le problème à machines parallèles a été largement étudié (Cheng and Sin, 1990), notamment parce qu'il apparaît comme une relaxation de problèmes plus complexes comme le flow-shop hybride ou le RCPSP (Resource-Constrained Project Scheduling Problem). Dans la littérature, plusieurs méthodes ont été proposées pour la résolution de ce problème. Chen and Powell (1999) résolvent le problème par une méthode de génération de colonnes. Pearn et al. (2007) proposent un programme linéaire et une heuristique efficace pour la résolution du problème avec considération de contraintes de priorité et de temps de préparation par familles d'opérations. Salem et al. (2000)

proposent des méthodes basées sur la recherche arborescente. Dans ce même contexte et plus récemment, Néron et al. (2008) comparent deux schémas de branchement différents et plusieurs stratégies de branchement pour le problème de la minimisation du makespan avec considération de fenêtres temporelles pour les tâches à exécuter.

Baev et al. (2002) et van den Akker et al. (2005) traitent le problème à machines parallèles avec contraintes de précédence pour la minimisation de la somme des dates de fin d'exécution des tâches et la minimisation du retard algébrique maximal, respectivement. Le problème avec temps de préparation est considéré dans Schutten and Leussink (1996) et dans Ovacik and Uzsoy (1995) pour le problème de minimisation du retard algébrique maximal. Uzsoy and Velasquez (2008) traitent ce critère pour le problème à une machine avec considération de temps de préparation par familles de tâches. Finalement, Nessah et al. (2005) proposent une borne inférieure et une méthode arborescente pour le problème de minimisation de la somme des dates de fin d'exécution des tâches.

Cependant, le problème combinant contraintes de précédence et temps de préparation a été beaucoup moins étudié. La présence simultanée des deux types de contraintes augmente de façon sensible la complexité du problème. Les problèmes intégrant une seule de ces deux limitations peuvent être résolus par un algorithme de liste, c'est-à-dire qu'il existe une combinaison des tâches, qui, lorsqu'on suit une certaine règle d'affectation et d'ordonnancement, aboutit à une solution optimale (Schutten, 1994). Pour les critères réguliers, la règle à suivre est *Earliest Completion Time* (ECT), qui consiste à affecter la tâche à la machine qui permet de la finir au plus tôt. Malheureusement, ce raisonnement n'est plus valide lorsque des contraintes de précédence et de temps de préparation entre les tâches sont considérées, comme cela est démontré dans Hurink and Knust (2001). Cela oblige à modifier la façon de traiter le problème. Nous devons notamment considérer non seulement les décisions sur l'ordre des tâches à exécuter, mais aussi les décisions sur l'affectation des tâches aux machines.

VI.3.1 Relation avec les problèmes d'ordonnancement des transports

Compte tenu du thème majeur abordé dans la thèse, cette partie a pour but de souligner le lien entre le problème particulier d'ordonnancement étudié dans ce chapitre et le problème global d'ordonnancement des transports.

Le problème d'ordonnancement considéré peut être vu comme un problème de tournées de

véhicules ou Vehicle Routing Problem (VRP) (cf. paragraphe I.2). Les ressources du problème à machines parallèles peuvent être considérées comme des véhicules identiques de la flotte dans un problème de tournées de véhicules. Les tâches correspondent aux clients qu'il est prévu de servir par les véhicules. Les temps d'exécution de chaque tâche correspondent aux temps de service pour chaque client visité et les temps de préparation sont associés aux temps de trajet entre les différents clients (tâches). La considération des fenêtres de temps des clients est aussi une des contraintes typiquement considérée dans les problèmes de tournées de véhicules (VRPTW) pour définir l'intervalle de temps pendant lequel le client peut être servi. Finalement, les contraintes de précedence entre les clients peuvent aussi apparaître dans certains types de problèmes de tournées de véhicules où certains clients doivent être servis avant d'autres.

Le problème à machine parallèles avec contraintes de précedence et temps de préparation peut être donc considéré comme un sous-problème du problème global traité dans les chapitres antérieurs.

VI.3.2 Définition du problème

Il s'agit de réaliser l'affectation de n tâches sur l'ensemble de m machines disponibles. Les relations de précedence entre les différentes tâches doivent être respectées, ainsi que les temps de préparation rencontrés lorsque les tâches sont affectées de façon consécutive sur la même machine. Chaque machine exécute au maximum une tâche simultanément et chaque tâche ne peut pas être exécutée par plus d'une machine à un instant de temps donné. Dans notre problème, la préemption n'est pas autorisée, c'est-à-dire que chaque tâche i doit s'exécuter durant p_i unités de temps consécutives sur la même machine. Les variables de décision du problème sont les dates de début des tâches $\{S_i\}_{i=1..n}$. Ainsi, la date de fin d'exécution de la tâche i peut se définir comme $C_i = S_i + p_i$.

Pour toute tâche i , nous définissons une fenêtre de temps $[r_i, d_i]$, où r_i et d_i correspondent à la date de disponibilité (date de début au plus tôt) et à la date d'échéance (date maximale idéale de fin d'exécution) de la tâche i , respectivement. Les dates d'échéance sont seulement considérées pour le calcul du retard algébrique maximal. Nous notons E l'ensemble modélisant les contraintes de précedence entre les tâches. La relation $(i, j) \in E$ signifie que la tâche i précède la tâche j (ce qui se note formellement $i \prec j$), la tâche j doit débiter après la fin de la tâche i ($S_j \geq C_i$). Finalement, nous définissons s_{ij} comme le temps de préparation nécessaire lorsque la

tâche j est exécutée par la même machine immédiatement après la tâche i . Pour deux tâches i et j exécutées de façon consécutive sur la même machine, alors $S_j \geq C_i + s_{ij}$, si i précède j , ou $S_i \geq C_j + s_{ji}$, si j précède i .

Pour la minimisation de la somme des dates de fin d'exécution de l'ensemble des tâches, la fonction objectif se définit comme $\min(\sum_{i=1}^n C_i)$. On s'aperçoit que, dans ce cas-là, les d_i des différentes tâches n'affectent pas la résolution du problème. Par contre, pour la minimisation du retard algébrique maximal sur l'ensemble des tâches, ces dates limites d_i occupent un rôle important, car la fonction objectif se définit comme $\min L_{\max}$, où $L_{\max} = \max_{i \in n} L_i$ et $L_i = C_i - d_i$. En utilisant la notation classique (Graham et al., 1979), les deux problèmes sont respectivement notés $P|prec, s_{ij}|\sum C_i$ et $P|prec, s_{ij}|L_{\max}$.

Exemple

Un ensemble de 5 tâches ($n = 5$) doit être exécuté sur 2 machines différentes ($m = 2$). Pour chaque tâche i , on donne dans le tableau VI.1 la valeur des différentes grandeurs p_i , r_i , d_i et s_{ij} . De plus, pour cet exemple, nous avons les contraintes de précédence suivantes : $1 \prec 4$ et $2 \prec 5$.

(a)				(b)					
n	p_i	r_i	d_i	s_{ij}	1	2	3	4	5
1	4	1	7	1	0	2	3	4	5
2	3	0	5	2	7	0	6	1	3
3	4	3	8	3	2	4	0	7	1
4	3	3	10	4	4	4	8	0	1
5	2	1	5	5	3	4	8	5	0

TAB. VI.1 – Données de l'exemple illustratif

La figure VI.1 représente une des solutions possibles pour ce problème. L'ensemble de contraintes de précédence est respecté : $S_5 = 13 \geq 3 = C_2$ et $S_4 = 5 \geq 5 = C_1$. Remarquons que la tâche 4 se voit forcée à retarder son début d'une unité de temps à cause de la contrainte de précédence. D'un autre côté, il faut vérifier que, pour chaque tâche, $r_i \leq S_i$ et que les temps de préparation entre tâches consécutives sur une même machine sont aussi respectés. En ce qui concerne la fonction objectif, notre critère d'évaluation de la qualité de la solution, observons que pour la minimisation de la somme des dates de fin des tâches la fonction vaut $z = \sum C_i = 43$ et pour le cas de la minimisation du retard algébrique maximal, $z = L_{\max} = L_5 = 10$.

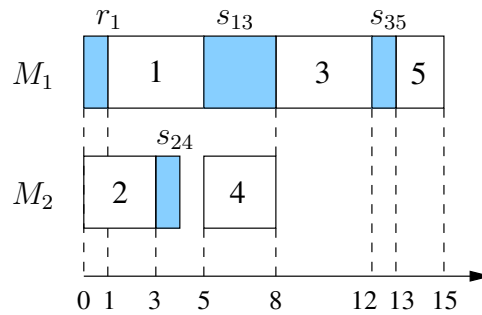


FIG. VI.1 – Exemple d'un ordonnancement réalisable

VI.3.3 Analyse des solutions

Dans Schutten (1994), les auteurs prouvent que le problème à machines parallèles avec considération de contraintes de précédence ou de temps de préparation peuvent être résolus optimalement par un algorithme de liste. Un algorithme de liste permet de représenter un ordonnancement uniquement sous la forme d'une liste de tâches sachant qu'il existe une règle dominante d'affectation des ressources aux tâches et d'ordonnancement des tâches dans l'ordre de la liste. La règle est dominante au sens où il existe une liste de tâches telle qu'une solution optimale est obtenue en suivant la règle. Schutten (1994) démontre que les ordonnancements produits par un algorithme de liste basé sur la règle d'affectation ECT (Earliest Completion Time) sont dominants.

Malheureusement, les problèmes d'ordonnancement qui présentent conjointement des contraintes de précédence et des temps de préparation ne peuvent pas systématiquement être résolus efficacement par un algorithme de liste (Hurink and Knust, 2001). C'est-à-dire qu'il peut ne pas exister de règle d'affectation des ressources aux tâches qui aboutisse à une solution optimale lorsqu'on énumère toutes les listes de tâches existantes. Ce fait est illustré à partir de l'exemple suivant :

Considérons la minimisation de la somme des dates de fin d'exécution pour un problème à 2 machines avec 4 tâches à ordonnancer. Le tableau VI.2 indique les données du problème.

(a)		
n	p_i	r_i
1	1	0
2	1	0
3	1	2
4	1	2

(b)				
s_{ij}	1	2	3	4
1	0	10	2	10
2	10	0	1	1
3	10	10	0	10
4	10	10	10	0

TAB. VI.2 – Données pour le deuxième exemple

Nous considérons tout d'abord le problème sans contraintes de précédence. Les listes de tâches $\{1, 2, 4, 3\}$ et $\{2, 1, 4, 3\}$ atteignent deux solutions optimales ($\sum C_i = 9$) lorsque les ressources sont affectées avec la règle Earliest Completion Time (voir figure VI.2a). Toutes les autres séquences de tâches aboutissent à des ordonnancements sous-optimaux. Considérons à présent le même problème auquel on ajoute la contrainte de précédence $3 \prec 4$. Dans ce cas, il n'existe pas de règle d'affectation qui aboutisse à une solution optimale pour les listes de tâches qui respectent les contraintes de précédence. La solution optimale ($\sum C_i = 11$) est atteinte pour la liste $\{1, 2, 3, 4\}$ lorsque la tâche 3 n'est pas exécutée par la machine qui permet de la finir le plus tôt (figure VI.2b).

Dans le même travail, les auteurs prouvent que même si un ordre total est imposé sur l'ensemble de dates de début des tâches, *i.e.* $S_1 \leq S_2 \leq \dots \leq S_n$, le problème reste *NP-difficile* au sens fort. Donc, déterminer la meilleure allocation de ressources s'avère nécessaire, en plus de trouver la meilleure liste de tâches.

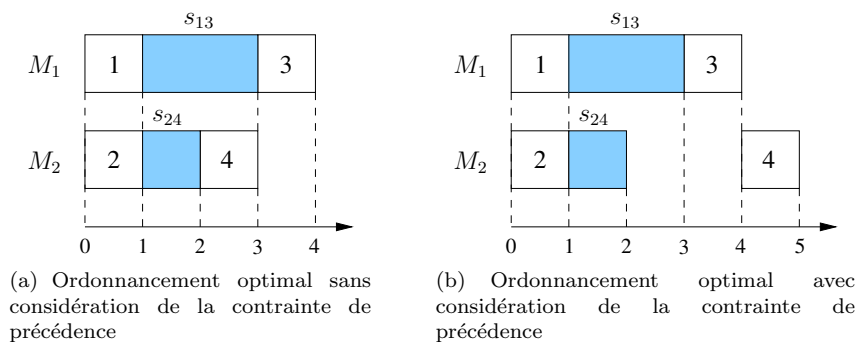


FIG. VI.2 – Exemple d'affectation de tâches

Neumann et al. (2002) démontrent que la réalisabilité d'un ordonnancement peut être prouvée en temps polynomial ($O(n^3)$) si l'on dispose de l'ensemble des dates de début des tâches. L'idée est de représenter l'ordonnancement par un graphe, puis de vérifier, à partir d'un calcul de flot maximum, l'exécution des tâches et la satisfaction des contraintes de ressource.

Le graphe est constitué de deux sommets par tâche, le premier représente la date de début i_t et le second la date de fin d'exécution i_s de la tâche. Pour représenter le transfert des unités de ressource entre les tâches, des arcs à capacité unitaire sont définis entre les sommets $i_s - j_t$. Si la tâche j peut être exécutée après la tâche i par la même machine ($S_j \geq S_i + p_i + s_{ij}$), alors un arc direct $i_s - j_t$ est créé. Finalement, quatre sommets fictifs sont définis. Deux sommets ($0_s, 0_t$), le sommet source S (origine du flot) et le sommet puits T (destination du flot). Les arcs $S-0_s$ et 0_t-T sont des arcs avec m unités de capacité. Ces arcs assurent la satisfaction de la contrainte due au nombre limité de ressources disponibles. Des arcs de capacité unitaire sont considérés entre $S-i_s$ et i_t-T avec la finalité d'assurer que toutes les tâches sont exécutées.

La figure VI.3 montre le graphe qui correspondent à l'ordonnancement du paragraphe VI.3.2, avec les dates de début des tâches ($S_1 = 1, S_2 = 0, S_3 = 8, S_4 = 5, S_5 = 13$). Le calcul du flot maximum détermine la réalisabilité de l'ordonnancement et propose une affectation des tâches aux machines qui respecte les dates de début des tâches. Nous observons qu'un flot de $m + p$ unités est nécessaire pour assurer l'exécution des tâches et satisfaire les contraintes de ressource.

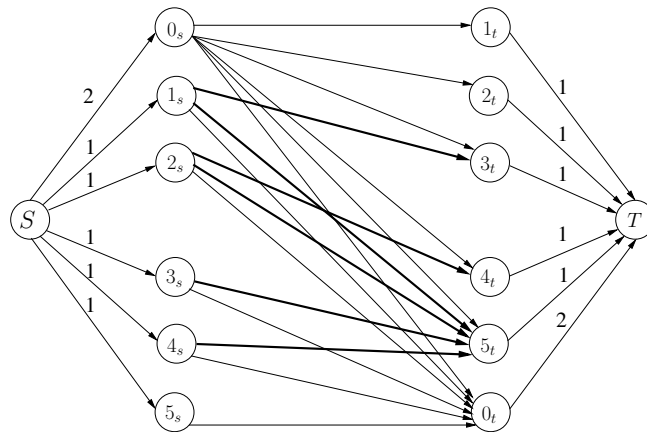


FIG. VI.3 – Graphe pour déterminer la réalisabilité d'un ordonnancement

De cette analyse, nous pouvons déduire que les solutions peuvent être spécifiées par un couple (liste de tâches, affectation tâche/machine) ou directement par un vecteur de dates de début dont

la réalisabilité de l'affectation aux machines peut être calculée en temps polynomial.

Les ordonnancements semi-actifs

Un ordonnancement réalisable représenté par un vecteur S de dates de début est semi-actif s'il n'existe pas un ordonnancement réalisable lorsqu'on fait avancer d'une unité de temps l'exécution d'une des tâches (Sprecher et al., 1995). Les ordonnancements semi-actifs sont dominants pour les problèmes que l'on traite ici. Il existe toujours un ordonnancement optimal qui est semi-actif; en conséquence, l'espace de recherche peut être limité à l'ensemble des ordonnancements semi-actifs. Ce principe de dominance des ordonnancements semi-actifs est utilisé pour la proposition de nouvelles règles de dominance intégrées dans des schémas de recherche arborescente et locale (cf. paragraphes VI.4.3 et VI.5.4).

Le calcul de flot maximum proposé dans le paragraphe VI.3.3 peut être utilisé pour vérifier en temps polynomial si un ordonnancement est semi-actif ou non. Pour un ordonnancement de n tâches, il s'agit de vérifier qu'il n'existe pas un ordonnancement réalisable lorsqu'on avance d'une unité la date de début de chaque tâche (voir algorithme 8).

Algorithme 8: Procédure de vérification de la propriété de semi-activité pour un ordonnancement $S = \{S_0, S_1, \dots, S_n\}$

Pas 1 : Construire le graphe qui correspond à l'ordonnancement S .

pour ($i=1..n$) **faire**

 Pas 2 : Etablir $S_i := S_i - 1$.

 Pas 3 : Actualiser le graphe en supprimant les arcs $j_s - i_t$ tels que $S_i < S_j + p_j + s_{ji}$.

 Pas 4 : Si l'ordonnancement est réalisable alors S n'est pas un ordonnancement semi-actif, retourner faux.

 Pas 5 : Etablir $S_i := S_i + 1$ et ajouter les arcs supprimés au Pas 3.

Pas 6 : S est un ordonnancement semi-actif, retourner vrai.

Implications sur les méthodes de résolution

Nous proposons une méthode de résolution exacte basée sur la recherche arborescente pour résoudre les instances de petite taille et une méthode de recherche locale qui s'avère efficace pour les instances de grande taille. Les méthodes se fondent sur la représentation des solutions; une liste de tâches et l'affectation des tâches aux machines. L'algorithme pour vérifier si un ordonnancement est semi-actif (algorithme 8) est utilisé comme règle de dominance pour supprimer les ordonnancements partiels non semi-actifs.

Dans le paragraphe VI.4, nous présentons une méthode de séparation et évaluation qui utilise conjointement cette règle de dominance, le calcul de bornes inférieures et des techniques de coupe fondées sur la propagation de contraintes. En ce qui concerne la recherche locale, plutôt que d'utiliser les opérateurs de voisinage classiques pour l'allocation des machines (permutation de tâches, re-allocations, etc.), nous proposons de les intégrer dans le schéma de recherche arborescente. Ceci permet de restreindre les voisinages explorés avec l'utilisation de bornes inférieures, de techniques de propagation de contraintes et de règles de dominance. Le schéma de résolution de la méthode CDS (Climbing Discrepancy Search) proposée par Milano and Roli (2002) est très pratique pour atteindre ce but. Cependant, nous devons toujours envisager des méthodes qui permettent de supprimer les solutions dominées seulement lorsque la solution dominante fait partie du voisinage exploré. Dans le paragraphe VI.5, nous présentons les variantes de la méthode CDS proposées pour résoudre les problèmes.

VI.4 Recherche arborescente

Une structure arborescente à deux niveaux de décision (ordre des tâches et allocation de ressources) est proposée dans le paragraphe VI.4.1. Les bornes inférieures et les mécanismes de propagation de contraintes sont présentés dans le paragraphe VI.4.2. Les règles de dominance sont présentées dans le paragraphe VI.4.3.

VI.4.1 Structure arborescente

Nous définissons un nœud comme un ordonnancement partiel $\sigma(p)$ de p tâches. Chaque nœud entraîne au plus $m \times (n - p)$ nœuds fils. Le terme $n - p$ correspond à la sélection de la prochaine tâche à ordonnancer. Seules sont candidates à être ordonnancées les tâches dont les tâches précédentes ont déjà été ordonnancées (*i.e.*, font partie de l'ordonnancement $\sigma(p)$). A tout moment, la liste construite respecte les contraintes de précédence entre tâches. Lorsqu'une tâche est sélectionnée, nous devons considérer les m affectations possibles aux machines. En pratique, pour homogénéiser le concept de nœud, les deux types de décision ont été intégrés : un branchement correspond à la sélection de la prochaine tâche à ordonnancer et au choix de la ressource allouée. Ainsi, une solution est atteinte lorsque le nœud représente un ordonnancement complet, *i.e.*, lorsque $p = n$.

VI.4.2 Evaluation d'un nœud

L'évaluation d'un nœud est différente selon la fonction objectif traitée. Nous proposons le calcul de bornes inférieures et supérieures pour les deux critères étudiés. Des mécanismes de propagation de contraintes sont utilisés comme techniques de coupe dans le problème de minimisation du retard algébrique maximal.

Bornes

D'abord, nous proposons un calcul simple de bornes inférieures. Pour chaque nœud (ordonnancement partiel), nous actualisons la date de début au plus tôt des tâches non ordonnancées en prenant en compte les décisions de branchement dérivées des contraintes de précédence entre tâches. Le temps de fin d'exécution minimal (pour le critère $\min \sum C_i$) et du retard algébrique minimal (pour le critère $\min L_{\max}$) est calculé pour chaque tâche non ordonnancée. La borne inférieure est comparée avec la meilleure solution courante.

Pour le critère $\min \sum C_i$, nous proposons une calcul de la borne spécifique présentée dans Nesaiah et al. (2005). La borne a été proposée pour le problème à machines parallèles, avec temps de préparation et la considération de dates de début au plus tôt ($Pm|s_{ij}, r_i|\sum C_i$). Le problème est une relaxation du problème avec contraintes de précédence; la borne inférieure est donc valide pour le problème étudié. Nous présentons ci-dessous la borne inférieure basée sur la relaxation de la contrainte de préemption.

Nous définissons S_* comme l'ordonnancement obtenu avec la règle SRPT (Shortest Remaining Processing Time) pour le problème relâché $1|r_i, (\frac{p_i}{m} + s_i^*), pmtn|\sum \max(C_i^* - s_i^*, r_i + p_i)$, où $s_i = \min_{j \neq i} s_{ij}$ et $s_i^* = \frac{s_i}{m}$. $C_{[i]}^*(S_*)$ est le temps de fin d'exécution de la tâche i avec considération d'un temps d'exécution $p_i + s_i^*$ pour chaque tâche i . Nous définissons $(a_{[1]}, a_{[2]}, \dots, a_{[n]})$ comme la suite obtenue si l'on ordonne (a_1, a_2, \dots, a_n) dans l'ordre décroissant, où $a_i = p_i + r_i + s_i^*$. On a alors $BI = \sum \max[C_{[i]}^*(S_*), a_{[i]}] - \sum s_i^*$ une borne inférieure du problème $Pm|prec, s_{ij}, r_i|\sum C_i$. La complexité de l'algorithme pour le calcul de la borne inférieure est $O(n \log n)$, la même que pour la règle SRPT.

Nous proposons aussi le calcul d'une borne supérieure. La borne est calculée à partir d'un algorithme de liste qui sélectionne la combinaison tâche non ordonnancée / machine qui permet de la débiter au plus tôt (*EST, Earliest Completion Time*). Le choix de ce critère est dû au fait

qu'il est intuitivement compatible avec la minimisation de la somme des temps de préparation, critère qui a globalement un impact positif pour la minimisation de critères réguliers (Artigues et al., 2005). Dans le cas où plusieurs paires tâche/machine sont sélectionnées par la règle, nous proposons comme deuxième critère pour la sélection l'utilisation de la règle *SPT* (*Shortest Processing Time*) pour le critère $\min \sum C_i$ et *EDD* (*Earliest Due Date*) pour le critère $\min L_{\max}$.

Raisonnement énergétique

Le raisonnement énergétique entre dans le cadre de la propagation de contraintes, ensemble de techniques de réécriture des contraintes qui facilitent l'exploration de l'ensemble des solutions. Ce type de raisonnement est lié au concept d'énergie; celui-ci permet d'effectuer des raisonnements quantitatifs intégrant les contraintes de temps et de ressources (Lopez and Esquirol, 1996).

Nous proposons un test de réalisabilité basé sur le raisonnement énergétique comme technique de coupe pour le critère $\min L_{\max}$. L'extension du raisonnement pour les problèmes avec temps de préparation est proposée. Nous appliquons le test de réalisabilité pour vérifier si la meilleure solution atteinte à partir du nœud évalué peut parvenir à être au moins aussi bonne que la meilleure solution courante.

L'énergie est produite par les ressources disponibles et consommée par les opérations. Nous déterminons l'énergie consommée par les tâches (E_{requis}) dans un intervalle $\Delta = [t_1, t_2]$, puis nous la comparons avec l'énergie disponible ($E_{produite} = m \times (t_2 - t_1)$). Nous devons aussi considérer l'énergie requise pour les temps de préparation (E_{setup}). Si $E_{requis} + E_{setup} > E_{produite}$ le nœud peut être coupé.

Pour un intervalle de temps Δ où un ensemble F de k tâches peuvent consommer de l'énergie, nous pouvons facilement démontrer que la quantité minimale d'intervalles de temps entre tâches (ou de temps de préparation) est de $\alpha = \max(0, k - m)$. On a alors que la somme des α plus petits temps de préparation de l'ensemble $\{s_{ij}\}, i, j \in F$ est une borne inférieure de l'énergie requise pour les temps de préparation.

L'énergie consommée dans un intervalle Δ est $E_{requis} = \sum_i \max(0, \min(p_i, t_2 - t_1, r'_i + p_i - t_1, t_2 - d'_i + p_i)) + \sum_l^\alpha s_{[l]}$, où $s_{[l]}$ sont les temps de préparation de l'ensemble $\{s_{ij}\}, i, j \in F$ ordonnés dans l'ordre décroissant et les fenêtres de temps $[r'_i, d'_i]$ pour chaque tâche i non ordonnancée sont déduites des ajustements des bornes des fenêtres temporelles, suivant :

$$r'_i = \max\{r_i, r_j + p_j; \forall j \in \Gamma_i^-\} \text{ et } d'_i = \min\{Z_{meilleure} + d_i, d'_j - p_j; \forall j \in \Gamma_i^+\},$$

où Γ_i^- et Γ_i^+ sont respectivement les ensembles de tâches prédécesseurs et de tâches successeurs pour la tâche i et $Z_{meilleure}$ est la valeur du critère L_{\max} pour la meilleure solution courante.

Nous illustrons sur la figure VI.4 la procédure pour le calcul de l'énergie requise pour les tâches non ordonnancées (les tâches 1 à 5 dans l'exemple) pour un problème avec 3 machines parallèles. Pour chaque tâche, nous déterminons la fenêtre de temps et la quantité minimale d'énergie requise dans l'intervalle de temps $\Delta = [t_1, t_2]$. Pour l'énergie requise par les temps de préparation E_{setup} , nous considérons les α plus petits temps de préparation. Dans l'exemple, $k = 4$ (la consommation de la tâche 1 est nulle) et $m = 3$; nous devons donc considérer seulement le plus petit des temps de préparation ($E_{setup} = s_{35} = 2$).

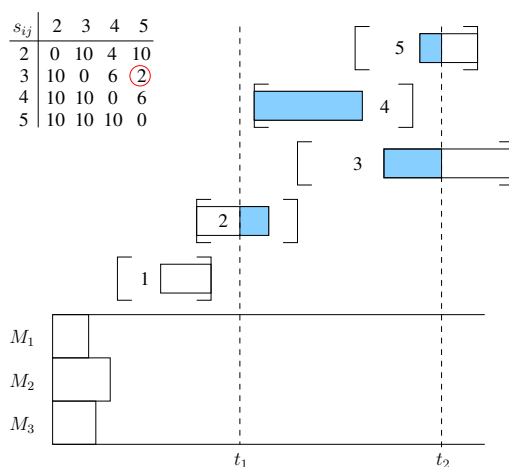


FIG. VI.4 – Énergie minimale requise pour un ordonnancement partiel

L'intervalle de temps $\Delta = [t_1, t_2]$ considéré pour le calcul de l'énergie est $t_1 = \min r'_i, \forall i \in F$ et $t_2 = d'_j$, où j est la tâche non ordonnancée avec la plus petite fenêtre de temps $\min(d'_j - r'_j), \forall j \in F$. La limitation de ne considérer qu'un unique intervalle de temps (l'algorithme n'est donc pas complet) permet de réduire la complexité du test énergétique à $O(n^2)$.

VI.4.3 Règles de dominance

Nous proposons une règle de dominance pour restreindre l'espace de solutions. Cette règle consiste à tenter de prouver l'existence d'un ordonnancement dominant qui permet de couper l'exploration à partir du nœud évalué. La règle exploite le fait que les ordonnancements semi-actifs sont dominants (cf. paragraphe VI.3.3). Nous définissons $J(\sigma(p))$ comme l'ensemble de tâches

d'un ordonnancement partiel $\sigma(p)$. Nous définissons également le *front* $F(\sigma(p)) \subseteq J(\sigma(p))$ de l'ordonnancement partiel $\sigma(p)$ comme l'ensemble des dernières tâches affectées sur chaque machine. Sur la figure VI.5, $F(\sigma(p)) = \{4, 5\}$.

Règle de dominance 1 : *Un ordonnancement partiel $\sigma(p)$ est dominé s'il existe un autre ordonnancement partiel $\sigma'(p)$ composé du même ensemble de tâches $J(\sigma'(p)) = J(\sigma(p))$, de même front $F(\sigma'(p)) = F(\sigma(p)) = F$ et avec une date de début d'exécution des tâches qui font partie du front plus petite ou égale que dans l'ordonnancement $\sigma(p)$, $S'_i \leq S_i, \forall i \in F$.*

Pour vérifier si un ordonnancement partiel $\sigma(p)$ est dominé nous proposons l'algorithme 8 (vérification de la propriété de semi-activité pour un ordonnancement) avec les modifications suivantes :

- Au Pas 1, lors de la construction du graphe de l'ordonnancement $\sigma(p)$ (comme dans le paragraphe VI.3.3), les arcs qui sortent des tâches qui font partie du front F ne sont pas créés de façon à toujours conserver le même front. En effet, sur la figure VI.5, la tâche 5 ($S_5 = 18$) peut être exécutée après la tâche 4, si l'arc $4_s - 5_t$ est créé, avec un temps de début d'exécution plus petit ($S_5 = 17$). Cependant, le nouvel ordonnancement $\sigma(p)$ est non-dominant (voir le contre-exemple du paragraphe VI.3.3).
- Au Pas 2, seules les tâches du front F sont sélectionnées pour la vérification de l'existence d'un ordonnancement réalisable (lorsque leur date de début d'exécution est réduite d'une unité de temps).

Si l'algorithme retourne faux, c'est-à-dire que l'ordonnancement $\sigma(p)$ n'est pas un ordonnancement semi-actif, le nœud est coupé.

Dans Demeulemeester and Herroelen (1997) des règles de dominance similaires ont déjà été proposées pour le problème d'ordonnancement de projet avec contraintes de ressources (qui est une extension du problème à machines parallèles avec contraintes de précédence, mais sans considération de temps de préparation). Toutefois, dans Demeulemeester and Herroelen (1997), les ensembles de coupes du graphe sont stockés entraînant des besoins importants de mémoire.

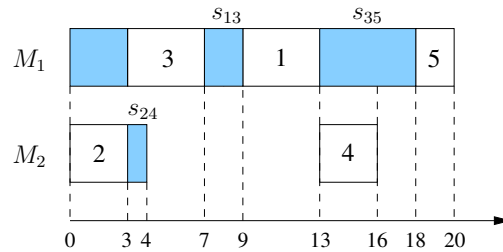


FIG. VI.5 – Ordonnancement partiel du nœud évalué

VI.5 Recherche arborescente basée sur les divergences

VI.5.1 Recherche arborescente à divergences limitées

Les méthodes standard de recherche arborescente s'avèrent inefficaces pour résoudre les problèmes de grande taille. Pour surmonter l'explosion combinatoire lors de l'exploration de l'espace de recherche, nous utilisons une recherche arborescente basée sur des divergences par rapport aux choix effectués initialement par une heuristique de référence. Ces méthodes sont fondées sur la bonne performance de l'heuristique choisie. De la solution donnée par cette heuristique, une recherche locale est réalisée de manière ordonnée, en examinant d'abord les solutions avec le moins de divergences par rapport à cette solution de référence, puis en s'en éloignant jusqu'à couvrir tout l'espace de recherche autorisé.

La méthode *LDS* (*"Limited Discrepancy Search"* ou *recherche à divergences limitées*) (Harvey and Ginsberg, 1996) a été conçue initialement comme une méthode itérative pour la résolution de problèmes à variables binaires par recherche arborescente, où chaque nœud de l'arbre a deux nœuds fils représentant respectivement les décisions d'affecter une variable aux valeurs 0 (non) ou 1 (oui). L'autre principe de base de cette méthode est d'explorer d'abord les solutions qui présentent des divergences dans les niveaux les plus hauts de l'arbre. En effet, les erreurs les plus importantes de l'heuristique sont celles réalisées dans les premiers niveaux où peu de décisions ont encore été prises.

La figure VI.6 montre le comportement de la méthode LDS pour un arbre binaire en indiquant le nombre de divergences associées à chaque nœud. Nous considérons la branche de gauche comme la décision donnée par l'heuristique de référence. A l'itération 0, l'exploration atteint la solution de l'heuristique, puis à l'itération 1 les solutions qui diffèrent au plus d'une décision (une divergence)

deux heuristiques de branchement. En ce qui concerne l'heuristique de sélection de la tâche à ordonnancer, le choix est de sélectionner la tâche qui peut débiter le plus tôt (règle EST ou Earliest Start Time). En cas d'égalité de date de début d'exécution, nous décidons de donner la priorité à la tâche avec une durée d'exécution plus petite (règle SPT ou Shortest Processing Time) pour le critère $\min \sum C_i$ et de donner la priorité à la tâche avec une date d'échéance d_i la plus petite (règle EDD ou Earliest Due Date) pour le critère $\min L_{\max}$. Le choix respectif de ces règles est dû au fait que ces règles sont optimales pour le problème à une machine pour les critères considérés. Nous utilisons une heuristique statique pour la sélection de la tâche à exécuter, c'est-à-dire que l'ordre des tâches est déterminé au début de la recherche arborescente. En revanche, l'heuristique d'affectation des tâches aux machines est dynamique (la tâche est affectée à la machine qui permet de la finir au plus tôt).

Pour chaque type de décisions, nous considérons deux types de divergences : une divergence pour la sélection de la tâche et une divergence pour l'affectation à la machine. Dans le cas des arbres non-binaires, les divergences peuvent être considérées de manière différente (voir la figure VI.7). Le mode de comptage binaire considère que suivre la décision de l'heuristique correspond à 0 divergence, et toutes les autres décisions sont comptabilisées comme 1 divergence. Le mode de comptage non-binaire considère que, plus la décision s'écarte par rapport à la décision proposée par l'heuristique, plus le nombre de divergences augmente. Nous proposons l'évaluation des deux modes de comptage pour la décision de la tâche à ordonnancer. La prise en compte de temps de préparation entre les tâches peut entraîner que l'ordre des tâches déterminé par la règle EST (Earliest Start Time) à partir d'un certain nombre de divergences ne soit plus une bonne heuristique. Nous proposons donc l'utilisation du comptage binaire pour améliorer la stratégie de branchement. D'un autre côté, nous utilisons le mode non-binaire pour le choix de la machine, car nous supposons que l'heuristique d'affectation commet très peu d'erreurs. En effet, l'efficacité de l'heuristique d'affectation est montrée dans le paragraphe VI.6.2.

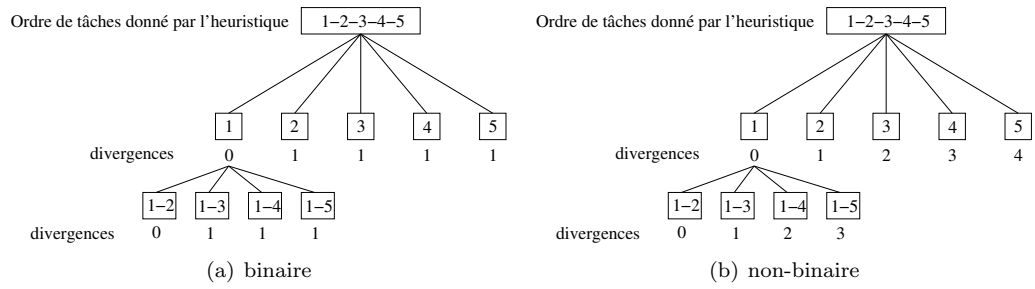


FIG. VI.7 – Exemple des modes de comptage de divergences pour la sélection de la tâche à ordonnancer

Nous proposons de tester trois schémas de branchement différents. Le *DBDFS* (Beck and Perron, 2000) correspond à une recherche en profondeur classique où les nœuds explorés sont limités par le nombre de divergences. Nous proposons en plus deux autres stratégies, *LDS-haut* et *LDS-bas*, qui considèrent le nombre de divergences pour déterminer l'ordre d'exploration des solutions. Le nœud exploré est le nœud avec le plus petit nombre de divergences. Pour un nombre de divergences donné, *LDS-haut* explore d'abord les nœuds avec un niveau de profondeur plus petit. *LDS-bas* contrairement à *LDS-haut* donne d'abord la priorité aux nœuds plus profonds dans l'arbre. La figure VI.8 montre, pour une recherche à 2 divergences, que les trois schémas explorent les mêmes solutions mais avec un ordre différent.

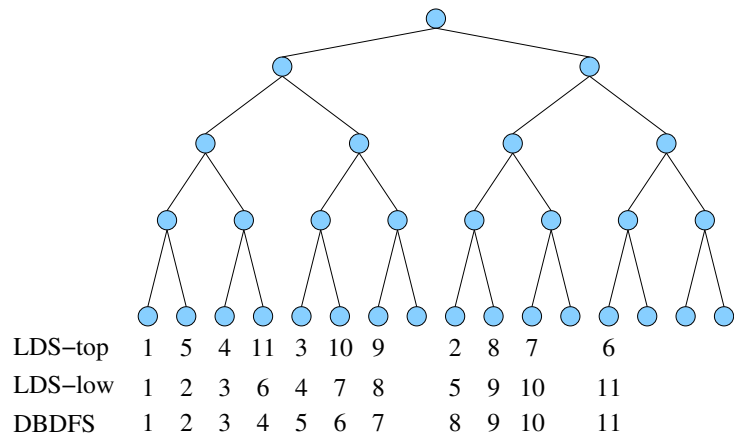


FIG. VI.8 – Ordre d'exploration des feuilles pour différent schémas de branchement

VI.5.3 Recherche locale à grands voisinages basée sur les divergences

La méthode LDS a été présentée comme une méthode de recherche arborescente complète ou tronquée. Dans ce paragraphe, nous proposons d'intégrer la méthode dans un schéma de recherche locale. Dans une méthode de recherche locale, nous définissons un voisinage $N_k(x)$ d'une solution x (où k spécifie les variations admissibles par rapport à la solution x). Si une solution x' meilleure que x est trouvée dans $N_k(x)$, alors le voisinage $N_k(x')$ de cette nouvelle solution est exploré. Pour les problèmes de grande taille, les voisinages à explorer deviennent trop vastes. Nous pouvons considérer l'exploration du voisinage $N_k(x)$ (où k est de grande taille) comme un sous-problème d'optimisation (Shaw, 1998). Dans ce contexte, une méthode basée sur les divergences pour définir le voisinage à explorer est envisageable. La méthode LDS permet de guider l'exploration du voisinage de façon ordonnée ; cet ordre doit normalement favoriser la recherche de meilleures solutions. L'autre avantage de LDS est que la méthode est compatible avec les techniques de coupe qui permettent de rejeter le plus rapidement possible les solutions non intéressantes (cf. paragraphe VI.3.3).

CDS (*Climbing Discrepancy Search*) (Milano and Roli, 2002) est la première méthode de recherche locale à grands voisinages basée sur LDS (voir algorithme 9). A chaque itération, une recherche avec k divergences autorisées par rapport à la meilleure solution courante est réalisée. Si une meilleure solution est trouvée, alors CDS explore le voisinage de cette nouvelle solution. Dans le cas où l'exploration s'achève sans trouver une meilleure solution, le nombre de divergences autorisées k est augmenté d'une unité.

Algorithme 9: Algorithme *CDS*

Pas 0 : Initialiser le compteur de divergences autorisées ($k := 1$) et le nombre maximal de divergences ($k_{max} := n$, pour un comptage binaire)
 Pas 1 : Déterminer la Sol_{ref} avec *Heuristique_Initiale()*
tant que $k \leq k_{max}$ **faire**
 | Pas 2 : Explorer l'ensemble de solutions S avec k divergences de Sol_{ref}
 | Pas 3 : Trouver la meilleure solution (Sol) de l'ensemble de solutions S
 | **si** Sol est meilleure que Sol_{ref} **alors**
 | | Pas 4 : Remplacer Sol_{ref} par Sol et réinitialiser le compteur de divergences autorisées ($k := 1$)
 | **sinon**
 | | Pas 5 : Augmenter le compteur de divergences autorisées d'une unité ($k := k + 1$)

Le principal inconvénient de CDS est l'accroissement rapide des voisinages explorés pour

les problèmes de grande taille. Pour surmonter ce fait, Hmida et al. (2007) proposent *CDDS* (*Climbing Depth-bounded Discrepancy Search*) qui mélange les principes de CDS et de DDS. Le voisinage de la meilleure solution est limité non seulement par le nombre de divergences mais aussi par le niveau de profondeur dans l'arbre. Cette méthode retarde l'explosion en taille du voisinage exploré et renforce l'idée que les erreurs les plus importantes de l'heuristique sont celles réalisées dans les premiers niveaux de décision.

Nous proposons deux variantes de CDS et CDDS pour les problèmes traités. Les méthodes sont fortement liées au concept de voisinages variables ou *Variable Neighborhood Search* (Hansen and Mladenovic, 2001), car la taille et la structure des voisinages explorés varient à mesure que la recherche avance. *HD-CDDS* (*Hybrid Discrepancy CDDS*) (voir algorithme 10) est aussi un mélange entre les principes de CDS et CDDS. La recherche débute avec un schéma de recherche CDS, mais si pour un nombre de divergences k_{limit} défini au préalable une meilleure solution n'a pas été trouvée, alors le nombre de divergences est incrémenté mais ces divergences sont autorisées seulement pour un certain nombre x de niveaux de profondeur, où $x = d_{max} - d_{min}$ (voir le Pas 6 de l'algorithme 10). Lorsque cette recherche limitée par les divergences et par les niveaux est achevée, nous proposons soit d'incrémenter le nombre de divergences autorisées tout en gardant le même nombre de niveaux où les divergences sont autorisées (Pas 8A dans l'algorithme 10), soit d'incrémenter le nombre de niveaux x et le nombre de divergences n'est pas modifié (Pas 8B de l'algorithme).

Cette méthode empêche l'explosion en taille du voisinage exploré et augmente la mobilité des tâches par rapport à CDDS (fait qui est particulièrement intéressant pour les problèmes avec considération de temps de préparation entre tâches). L'autre avantage est que les niveaux qui limitent le voisinage peuvent être sélectionnés en prenant en compte la structure des solutions. Par exemple, pour le problème de la minimisation du retard algébrique maximal, nous proposons de débiter avec les niveaux $[d_{min}, d_{max}]$ autour de la tâche critique de sorte à améliorer rapidement la solution. En contrepartie, les valeurs des paramètres de la recherche (k_{limit}, x) doivent être déterminés au préalable.

La deuxième méthode proposée, *MC-CDS* (*Mix Counting CDS*), consiste à appliquer la recherche *CDS* mais avec une modification au niveau du comptage de divergences. Nous proposons de considérer un comptage binaire pour les divergences des niveaux situés au plus haut de l'arbre et un comptage non-binaire pour le reste des niveaux. De cette façon, l'exploration des branches

Algorithme 10: Algorithme *HD-CDDS*

Pas 0 : Initialiser le compteur de divergences ($k := 1$), le nombre maximal de divergences autorisées (k_{limit}) et les niveaux autorisés ($d_{min} := 0, d_{max} := n$)

Pas 1 : Déterminer la Sol_{ref} avec *Heuristique_Initiale()*.

tant que conditions de fin de recherche **faire**

 Pas 2 : Explorer l'ensemble de solutions S avec k divergences (entre les niveaux $[d_{min}, d_{max}]$) de Sol_{ref}

 Pas 3 : Trouver la meilleure solution (Sol) de l'ensemble de solutions S .

si Sol est meilleure que Sol_{ref} **alors**

 Pas 4 : Remplacer Sol_{ref} par Sol , réinitialiser le compteur de divergences ($k := 1$) et les niveaux autorisés pour les divergences ($d_{min} := 0, d_{max} := n$)

sinon

si $k < k_{limit}$ **alors**

 Pas 5 : Augmenter le compteur de divergences autorisées d'une unité ($k := k + 1$)

sinon

si $d_{max} - d_{min} = n$ **alors**

 Pas 6 : Réinitialiser les niveaux autorisés ($d_{min} := 0, d_{max} := x$).

sinon

 Pas 7 : Actualiser les niveaux ($d_{min} := d_{max}, d_{max} := d_{min} + x$).

si $d_{min} > n$ **alors**

 Pas 8A : Augmenter le compteur de divergences d'une unité

 ($k := k + 1$) et réinitialiser les niveaux autorisés ($d_{min} := 0, d_{max} := x$)

 Pas 8B : Réinitialiser et augmenter les niveaux autorisés ($d_{min} := 0,$

$d_{max} := x + 1$)

des niveaux les plus hauts de l'arbre est favorisée, et, d'un autre côté, les divergences sont autorisées pour tous les niveaux sans que la taille du voisinage devienne trop vaste.

VI.5.4 Règles de dominance adaptées aux divergences

Nous proposons d'adapter la règle de dominance présentée dans le paragraphe VI.4.3 au principe de la recherche locale. L'application directe de la règle dans les schémas de recherche locale proposés peut être très inefficace. En effet, les meilleures solutions du voisinage peuvent ne pas être explorées à cause de l'existence d'un ordonnancement partiel dominant dont découlent des coupes pour l'exploration de ces solutions. Malgré l'existence d'une solution meilleure que les solutions obtenues à partir de l'ordonnancement dominé, il est donc possible que cette solution ne soit pas considérée dans le voisinage exploré.

A cet effet, nous proposons une variante de la règle de dominance adaptée au voisinage exploré

(déterminé par le nombre de divergences autorisées). Une fois connu le critère qui spécifie le voisinage (par exemple, k divergences autorisées par rapport à l'ordre de tâches L), nous n'avons qu'à vérifier que le nouvel ordre de tâches L' qui permet d'obtenir l'ordonnement dominant fait partie des nœuds visités lors de l'exploration du voisinage ($L' \in G$, où G est l'ensemble d'ordres de tâches avec k divergences de distance par rapport à l'ordre L).

La règle de calcul du flot maximum présentée dans le paragraphe VI.4.3 n'est pas adaptable à l'existence de divergences. La vérification que l'ordonnement partiel dominant $\sigma'(p)$ fait partie du voisinage exploré n'est pas possible, car la règle permet de montrer l'existence de $\sigma'(p)$ mais l'ordonnement n'est pas spécifié.

Ainsi, nous proposons une nouvelle règle de dominance basée sur la position des tâches qui font partie du front.

Règle de dominance 2 : *Pour un ordonnancement partiel $\sigma(p)$ lié à un ordre de tâches L , s'il existe une permutation de tâches dans $F(\sigma(p))$ de sorte que l'ordonnement résultant $\sigma'(p)$ et son ordre de tâches L' satisfont : $L' \in G$ et $S'_i \leq S_i, \forall i \in F(\sigma(p))$, alors l'ordonnement partiel $\sigma(p)$ est dominé et le nœud associé peut être supprimé de l'exploration.*

La figure VI.9 montre que l'ordonnement partiel conserve le même front ($F(\sigma'(p)) = \{1, 2, 3\}$) que l'ordonnement du nœud évalué, une des tâches débute plus tôt ($S'_1 < S_1$) et pour le reste des tâches qui font partie du front les dates de début ne sont pas retardées ($S_2 = S'_2$ et $S_3 = S'_3$). Si l'ordre des tâches ordonnancées qui font partie du front est $1 - 2 - 3$, toutes les permutations possibles qui satisfont les contraintes de précédence et qui respectent le nombre de divergences sont testées afin de trouver un ordonnancement partiel dominant qui permet de couper le nœud associé à l'ordonnement évalué.

Cette règle peut être appliquée par un algorithme de complexité $O(m!)$. Malgré la complexité exponentielle dans le pire des cas, l'utilisation de la règle de dominance s'avère intéressante lorsqu'on l'applique dans un schéma de recherche arborescente basée sur les divergences. La règle reste efficace pour un petit nombre de machines (cf. paragraphe VI.6.2). Si le nombre de machines m devient très grand, une énumération partielle peut être proposée afin de réduire le temps de calcul requis.

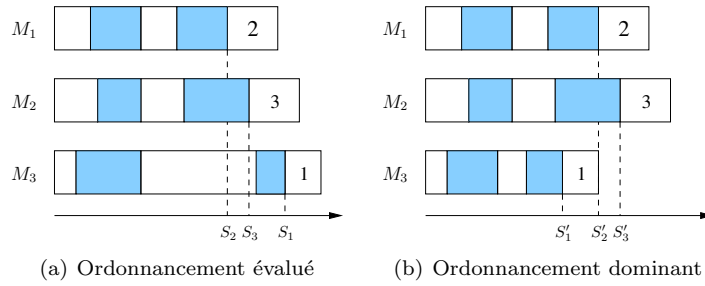


FIG. VI.9 – Exemple d'ordonnancement dominant

VI.6 Résultats expérimentaux

VI.6.1 Générateur d'instances

Les méthodes et techniques présentées dans ce chapitre ont été testées sur des instances générées aléatoirement. Ce choix est justifié par le manque d'instances proposées en benchmarking dans la littérature pour ce type de problème. Tout d'abord, les contraintes de précédence ont été générées par le logiciel RanGen (Demeulemeester et al., 2003) qui permet de créer des instances avec la possibilité de faire varier la densité du graphe de précédences. Ce logiciel a été conçu pour la génération d'instances pour le problème d'ordonnancement de projet à moyens limités (*RCPSP*). Les temps de préparation entre tâches et les fenêtres de temps ne peuvent pas être générées par le logiciel. Nous avons donc développé un générateur de temps de préparation et de fenêtres de temps pour les tâches. Les temps de préparation ont été générés à partir des distributions uniformes $U[1, 10]$ et $U[20, 40]$. La seule condition imposée aux temps de préparation est que l'inégalité triangulaire faible entre temps de préparation doit être respectée : $s_{ik} \leq s_{ik} + p_k + s_{kj}, \forall i, j, k$.

Pour les fenêtres de temps, nous avons utilisé la méthode habituelle proposée dans la littérature (Sourd, 2005). Les dates d'échéance d_i suivent une distribution uniforme $[max(0, P \times (1 - \tau - \rho/2)), P \times (1 - \tau + \rho/2)]$, avec $P = \sum(p_i + min_j(s_{ij}))$, $\tau \in [0, 1]$, $\rho \in [0, 1]$. Les dates de début au plus tôt r_i sont générées à partir des d_i et d'un paramètre α qui permet de faire varier la largeur des fenêtres de temps, $r_i = d_i - (p_i \times (2 + \alpha))$ où $\alpha \in [-0.5, +1.5]$.

VI.6.2 Tests et résultats

Dans ce paragraphe, nous décrivons les tests proposés pour l'évaluation des méthodes et nous analysons les résultats obtenus. Les algorithmes ont été implémentés en C++ et les expérimentations ont été réalisées sur un ordinateur de 2 GHz sous le système d'exploitation Linux Fedora 8 et avec 2 Go de mémoire RAM.

Un ensemble de 120 instances de petite taille (60 instances pour chaque critère) ont été générées ($n = 10, m = 3$ et $n = 15, m = 2$) pour l'évaluation de la règle d'affectation *ECT* et des règles de dominance. Puis, nous avons testé sur un ensemble de 120 instances de taille moyenne ($n = 40, m \in [2, 4]$) les règles de branchement (*LDS-haut*, *LDS-bas* et *DBDFS*) et les modes de comptage (*binnaire* et *non-binnaire*) afin de déterminer les meilleures options en vue de leur intégration dans la structure LDS pour la recherche locale. L'efficacité des bornes inférieures, des règles de dominance et du raisonnement énergétique proposés dans les paragraphes VI.3.3, VI.4.2 et VI.5.4 a été prouvée sur les instances de moyenne et grande taille ($n = 100, m \in [2, 4]$). Nous avons aussi comparé la méthode *CDS* et la méthode *HD-CDDS* avec les résultats obtenus dans Néron et al. (2008) sur ses instances difficiles du problème $Pm|r_i, q_i|C_{\max}$ (problème sans contraintes de précédence et sans temps de préparation). Finalement, nous avons comparé et évalué sur l'ensemble d'instances de grande taille les résultats obtenus avec IBM ILOG CP Optimizer 6.0, un logiciel commercial de résolution avec un algorithme de résolution spécifique pour les problèmes d'ordonnement (Laborie and Godard, 2007; Laborie, 2009).

Les instances de petite taille sont résolues optimalement. Le tableau VI.3 montre les résultats pour le critère de minimisation de $\sum C_i$ et le tableau VI.4 montre ceux obtenus pour la minimisation de L_{\max} . Pour l'évaluation des règles de dominance, nous comparons la recherche arborescente avec les bornes inférieures proposées et le raisonnement énergétique sans les règles de dominance (*Optimale*), avec la recherche arborescente lorsque la règle de dominance basée sur la permutation des tâches du front est intégrée (*Règle Front*) et avec la recherche arborescente lorsque la règle de dominance basée sur le calcul du flot maximum est intégrée (*Règle Flot*). La ligne *ECT* illustre les résultats quand on limite l'allocation des tâches à la machine qui permet de les finir au plus tôt (Earliest Completion Time).

60 Instances			
$n = 10, m = 3$	<i>Nb_Meilleure</i>	<i>Moy_nœuds</i>	<i>Moy_TCPU</i>
<i>Optimale</i>	60 (100.0 %)	484925	10.6
<i>Règle Front</i>	60 (100.0 %)	480444	12.3
<i>Règle Flot</i>	60 (100.0 %)	339541	27.7
<i>ECT</i>	53 (88.3 %)	61684	0.07

60 Instances			
$n = 15, m = 2$	<i>Nb_Meilleure</i>	<i>Moy_nœuds</i>	<i>Moy_TCPU</i>
<i>Optimale</i>	60 (100.0 %)	10126793	641.9
<i>Règle Front</i>	60 (100.0 %)	9480313	626.4
<i>Règle Flot</i>	60 (100.0 %)	7530154	454.6
<i>ECT</i>	54 (90.0 %)	1747416	2.5

TAB. VI.3 – Résultats de la règle d'affectation *ECT* et des règles de dominance pour le problème $\min \sum C_i$

60 Instances			
$n = 10, m = 3$	<i>Nb_Meilleure</i>	<i>Moy_nœuds</i>	<i>Moy_TCPU</i>
<i>Optimale</i>	60 (100.0%)	281896	5.6
<i>Règle Front</i>	60 (100.0%)	263474	7.9
<i>Règle Flot</i>	60 (100.0%)	219557	19.7
<i>ECT</i>	52 (86.7%)	69141	0.07

60 Instances			
$n = 15, m = 2$	<i>Nb_Meilleure</i>	<i>Moy_nœuds</i>	<i>Moy_TCPU</i>
<i>Optimale</i>	60 (100.0%)	11936385	884.8
<i>Règle Front</i>	60 (100.0%)	10503767	778.7
<i>Règle Flot</i>	60 (100.0%)	8945948	628.4
<i>ECT</i>	54 (90.0%)	4681104	7.27

TAB. VI.4 – Résultats de la règle d'affectation *ECT* et des règles de dominance pour le problème $\min L_{\max}$

Nous observons dans les tableaux VI.3 et VI.4 que la règle ECT est très efficace pour les deux problèmes. L'application de la règle permet en effet d'atteindre la solution optimale pour la plupart des instances (près de 90%) et permet aussi de réduire très sensiblement le temps moyen de calcul (Moy_TCPU). Pour les méthodes de recherche locale, nous allons considérer uniquement le problème de séquençement des tâches ; les tâches seront affectées aux machines selon la règle ECT. La règle de dominance basée sur la permutation des tâches du front est également efficace. En règle générale, cette règle permet de réduire le nombre de nœuds explorés ($Moy_nœuds$) et le temps moyen de calcul. Nous observons que la règle du calcul du flot maximum permet de réduire grandement le nombre de nœuds explorés et le temps de calcul, sauf pour les instances de très petite taille.

Les deux modes de comptage de divergences (*binnaire* et *non-binnaire*) ont aussi été évalués (uniquement pour la décision de séquençement des tâches). Nous avons testé, pour les instances de taille moyenne, le nombre de fois où le mode arrive à trouver la meilleure solution ($Nb_Meilleure$). Les modes ont été intégrés dans un schéma standard de LDS. Le temps de calcul est limité à 100 secondes. Les divergences ne sont pas autorisées pour la règle d'affectation des machines.

Nous avons enfin comparé les trois stratégies de branchement présentées dans le paragraphe VI.5.2 : DBDFS, LDS-haut et LDS-bas. Le tableau VI.5 montre que le mode binaire est plus performant que le mode non-binaire. Pour un ensemble de 120 instances (60 instances par critère), le mode binaire trouve une meilleure solution pour 75 % des instances, indépendamment de la stratégie de branchement choisie. Dans ce qui suit, nous retiendrons le mode de comptage binaire pour la structure de LDS pour la recherche locale.

120 Instances $n = 40, m \in [2, 4]$	$Nb_Meilleure$	
	mode <i>binnaire</i>	mode <i>non-binnaire</i>
<i>DBDFS</i>	90 (75.0 %)	48 (40.0 %)
<i>LDS-haut</i>	93 (77.5 %)	49 (40.8 %)
<i>LDS-bas</i>	98 (81.7 %)	31 (25.8 %)

TAB. VI.5 – Résultats de la comparaison entre modes de comptage de divergences

Le tableau VI.6 montre les résultats pour la comparaison entre les stratégies d'exploration. Dev_Moy indique la déviation moyenne par rapport à la meilleure solution trouvée. Le temps de calcul est limité à 100 secondes.

Mode binaire $n = 40, m \in [2, 4]$	$\min \sum C_i$ (60 instances)		$\min L_{\max}$ (60 instances)	
	<i>Nb_Meilleure</i>	<i>Dev_Moy</i>	<i>Nb_Meilleure</i>	<i>Dev_Moy</i>
<i>DBDFS</i>	43 (71.7 %)	0.91 %	47 (78.3 %)	1.86 %
<i>LDS-haut</i>	29 (48.3 %)	0.43 %	17 (28.3 %)	2.33 %
<i>LDS-bas</i>	50 (83.3 %)	0.71 %	59 (98.3 %)	0.75 %

TAB. VI.6 – Résultats pour la comparaison des stratégies de branchement

LDS-bas s'avère la stratégie plus efficace, car elle trouve la meilleure solution pour un plus grand nombre d'instances et l'écart moyen, lorsque la meilleure solution est trouvée par une autre des stratégies, est le plus petit. LDS-bas trouve la meilleure solution pour toutes les instances sauf pour une liée à la minimisation du retard algébrique et pour 50 instances sur 60 pour le critère de minimisation de la somme des dates de fin d'exécution des tâches. La stratégie LDS-bas est donc conservée pour le reste des tests à réaliser.

Les bornes inférieures, le raisonnement énergétique et la règle de dominance adaptée aux divergences sont comparés dans les tableaux VI.7 et VI.8. Une recherche LDS avec différentes configurations pour l'évaluation des nœuds a été lancée pour les instances de taille moyenne à grande. Dans la première configuration, nous considérons uniquement la borne inférieure calculée par propagation (*BIPC*). Dans la seconde configuration nous ajoutons le calcul de la borne inférieure (*BI_{NCY}*) proposée dans Nessah et al. (2005) pour le problème de $\min \sum C_i$ et le raisonnement énergétique (*NRJ*) pour le problème de $\min L_{\max}$. Finalement, dans la troisième configuration, nous ajoutons la règle de dominance adaptée aux divergences (*RDaD*). Nous comparons le nombre de fois où chaque configuration trouve la meilleure solution (*Nb_Meilleure*), le nombre de nœuds explorés en moyenne (*Moy-nœuds*), et le temps de calcul moyen pour atteindre la meilleure solution (*TMeilleure*), seulement pour les instances dont toutes les configurations ont trouvé la meilleure solution. Le temps de calcul est limité à 30 secondes.

60 Instances			
$n = 40, m \in [2, 4]$	<i>Nb_Meilleure</i>	<i>Moy_nœuds</i>	<i>TMeilleure</i>
<i>BIPC</i>	36 (60.0 %)	62007	4.52
<i>BI_{NCY}</i>	38 (63.3 %)	61742	4.47
<i>RDaD</i>	35 (58.3 %)	53373	1.69

60 Instances			
$n = 100, m \in [2, 4]$	<i>Nb_Meilleure</i>	<i>Moy_nœuds</i>	<i>TMeilleure</i>
<i>BIPC</i>	26 (43.3 %)	9259	17.55
<i>BI_{NCY}</i>	34 (56.7 %)	7813	15.63
<i>RDaD</i>	38 (63.3 %)	7606	8.71

TAB. VI.7 – Résultats sur l’efficacité de la borne inférieure et de la règle de dominance pour le problème de minimisation de $\sum C_i$

Les tableaux VI.7 et VI.8 montrent l’efficacité de la borne inférieure spécifique *BI_{NCY}* et du raisonnement énergétique avec le calcul de la consommation des temps de préparation. De plus, la règle de dominance adaptée aux divergences est très efficace, notamment pour les instances de grande taille et pour la minimisation de $\sum C_i$. En outre, le temps de calcul nécessaire pour atteindre la meilleure solution est réduit pour la plupart des instances lorsque la règle de dominance est utilisée.

60 Instances			
$n = 40, m \in [2, 4]$	<i>Nb_Meilleure</i>	<i>Moy_nœuds</i>	<i>TMeilleure</i>
<i>BIPC</i>	47 (78.3 %)	93737	4.81
<i>NRJ</i>	48 (80.0 %)	99856	4.24
<i>RDaD</i>	44 (73.3 %)	71737	4.59

60 Instances			
$n = 100, m \in [2, 4]$	<i>Nb_Meilleure</i>	<i>Moy_nœuds</i>	<i>TMeilleure</i>
<i>BIPC</i>	44 (3.3 %)	11474	4.29
<i>NRJ</i>	48 (80.0 %)	12961	3.58
<i>RDaD</i>	55 (91.7 %)	9462	3.17

TAB. VI.8 – Résultats sur l’efficacité de la borne inférieure, du raisonnement énergétique et de la règle de dominance pour le problème de minimisation de L_{\max}

Nous comparons la méthode *CDS* et la méthode *HD-CDDS* avec d’autres méthodes de recherche arborescente présentées dans Néron et al. (2008). Dans Néron et al. (2008), les auteurs comparent deux schémas de branchement différents, l’un basé sur les fenêtres de temps (*tw*) et un schéma chronologique (*chr*), et plusieurs techniques de branchement (*branch-and-bound* tronqué,

LDS, *beam search* et *branch-and-greed*) pour le problème $Pm|r_i, q_i|C_{\max}$. Nous avons adapté les méthodes proposées à ce problème et nous utilisons la même heuristique pour déterminer la solution initiale et la même borne supérieure que dans leur travail. Dans le tableau VI.9, nous comparons les résultats de LDS (où z est le nombre de divergences autorisées) et de la Beam Search (BS , où ω est le nombre de nœuds fils explorés), qui est la méthode obtenant les meilleurs résultats dans leur travail, avec les résultats obtenus par CDS et HD-CDDS. Nous avons évalué, pour un ensemble de 50 instances ($n = 100$ et $m = 10$), le nombre de fois où la meilleure solution est trouvée par chaque méthode ($Nb_Meilleure$) et le nombre de fois où la méthode est la seule méthode à trouver cette meilleure solution ($Nb_Meilleure_Stricte$). Le temps de calcul est limité à 30 secondes comme dans Néron et al. (2008).

50 instances	$Nb_Meilleure$	$Nb_Meilleure_Stricte$
$LDS_{z=1}^{tw}$	1 (2.0 %)	0
$LDS_{z=2}^{chr}$	7 (14.0 %)	0
$BS_{\omega=3}^{tw}$	25 (50.0 %)	3
$BS_{\omega=4}^{chr}$	22 (44.0 %)	0
<i>CDS</i>	35 (70.0 %)	6
<i>HD-CDDS</i>	38 (76.0 %)	9

TAB. VI.9 – Résultats pour la comparaison avec d'autres méthodes de recherche arborescente tronquée

Les résultats des méthodes que nous avons proposées sont très bons, même si les contraintes de précedence et les temps de préparation ne son pas considérées dans le problème. En effet, pour un ensemble de 50 instances, CDS et HD-CDDS trouvent la meilleure solution pour le plus grand nombre d'instances et une nouvelle meilleure solution est trouvée pour 6 et 9 instances, respectivement. Les résultats illustrent ainsi l'efficacité des méthodes de recherche locale à grands voisinages basées sur LDS pour ce problème.

Finalement, nous comparons les résultats obtenus par les méthodes de recherche locale avec les résultats obtenus par le logiciel IBM ILOG CP Optimizer 6.0 pour les problèmes $Pm|r_i, prec, s_{ij}|L_{\max}$ et $Pm|r_i, prec, s_{ij}|\sum C_i$.

Les quatre variantes de recherche locale hybrides (*CDS*, *CDDS*, *HD-CDDS*, *MC-CDS*) ont été implémentées avec la stratégie d'exploration *LDS-bas*, la règle de dominance adaptée aux divergences et un comptage binaire des divergences (sauf pour *MC-CDS* qui utilise un comptage mixte des divergences). Les divergences ne sont pas autorisées pour l'affectation de la tâche à

la machine, la règle d'affectation est la règle ECT (Earliest Completion Time). Les instances à grande taille ($n = 100, m \in [2, 4]$) ont été résolues. Nous avons choisi de fixer deux temps de résolution limites à 30 et 300 secondes. Nous comparons le nombre de fois où la meilleure solution a été trouvée par chaque méthode et la déviation moyenne par rapport à la meilleure solution trouvée.

30 instances	$TCPU = 30s$		$TCPU = 300s$	
$p \sim U[1, 5], s_{ij} \sim U[1, 10]$	<i>Nb_Meilleure</i>	<i>Dev_Moy</i>	<i>Nb_Meilleure</i>	<i>Dev_Moy</i>
<i>CDS</i>	17 (56.6 %)	0.64 %	7 (23.3 %)	0.51 %
<i>CDDS</i>	7 (23.3 %)	0.75 %	7 (23.3 %)	0.82 %
<i>HD-CDDS</i>	16 (53.3 %)	0.60 %	14 (46.7 %)	0.43 %
<i>MC-CDS</i>	17 (56.6 %)	0.64 %	10 (33.3 %)	0.45 %
<i>CP Optimizer</i>	4 (13.3 %)	1.51 %	2 (6.7 %)	1.47 %

30 instances	$TCPU = 30s$		$TCPU = 300s$	
$p \sim U[1, 5], s_{ij} \sim U[20, 40]$	<i>Nb_Meilleure</i>	<i>Dev_Moy</i>	<i>Nb_Meilleure</i>	<i>Dev_Moy</i>
<i>CDS</i>	9 (30.0 %)	0.23 %	6 (20.0 %)	0.18 %
<i>CDDS</i>	7 (23.3 %)	0.35 %	6 (20.0 %)	0.38 %
<i>HD-CDDS</i>	12 (40.0 %)	0.26 %	11 (36.6 %)	0.17 %
<i>MC-CDS</i>	11 (36.7 %)	0.25 %	13 (43.3 %)	0.26 %
<i>CP Optimizer</i>	10 (33.3 %)	0.70 %	5 (16.6 %)	0.63 %

TAB. VI.10 – Résultats pour la comparaison des variantes hybrides pour le problème de minimisation de $\sum C_i$

Dans le tableau VI.10, nous observons que les méthodes hybrides proposées améliorent la meilleure solution trouvée par CP Optimizer. Toutes les méthodes, sauf CDDS, trouvent la meilleure solution pour un nombre plus élevé d'instances et une déviation moyenne plus petite par rapport à la meilleure solution obtenue par CP Optimizer. Nous observons que le calcul d'une borne supérieure augmente grandement l'efficacité de la recherche arborescente tronquée.

Le tableau VI.11 montre les résultats pour le problème de minimisation du retard algébrique maximal. Dans ce cas, CP Optimizer obtient les meilleurs résultats. Cependant, les méthodes proposées restent toujours compétitives. La déviation moyenne est acceptable et la meilleure solution est trouvée par les méthodes hybrides pour 50 % et 37 % des instances, pour un temps de calcul de 30 et 300 secondes respectivement.

30 instances	$TCPU = 30s$		$TCPU = 300s$	
$p \sim U[1, 5], s_{ij} \sim U[1, 10]$	$Nb_Meilleure$	Dev_Moy	$Nb_Meilleure$	Dev_Moy
<i>CDS</i>	10 (33.3 %)	2.75 %	7 (23.3 %)	3.06 %
<i>CDDS</i>	9 (30.0 %)	2.65 %	8 (26.7 %)	3.28 %
<i>HD-CDDS</i>	13 (43.3 %)	1.92 %	10 (33.3 %)	2.56 %
<i>MC-CDS</i>	13 (43.3 %)	1.75 %	11 (30.0 %)	2.29 %
<i>CP Optimizer</i>	15 (50.0 %)	2.07 %	18 (60.0 %)	1.55 %

30 instances	$TCPU = 30s$		$TCPU = 300s$	
$p \sim U[1, 5], s_{ij} \sim U[20, 40]$	$Nb_Meilleure$	Dev_Moy	$Nb_Meilleure$	Dev_Moy
<i>CDS</i>	3 (10.0 %)	2.76 %	2 (6.0 %)	2.89 %
<i>CDDS</i>	3 (10.0 %)	2.71 %	2 (6.0 %)	2.88 %
<i>HD-CDDS</i>	13 (43.3 %)	2.12 %	7 (23.3 %)	1.55 %
<i>MC-CDS</i>	12 (40.0 %)	2.08 %	8 (26.7 %)	1.83 %
<i>CP Optimizer</i>	15 (50.0 %)	0.91 %	19 (63.3 %)	0.90 %

TAB. VI.11 – Résultats pour la comparaison des variantes hybrides pour le problème de minimisation du L_{\max}

VI.7 Conclusion

Dans ce chapitre, nous avons proposé une méthode de recherche arborescente basée sur les divergences pour résoudre le problème à machines parallèles avec contraintes de précédence et temps de préparation. Nous avons testé et comparé différentes modalités pour les éléments caractéristiques de la méthode LDS. Plusieurs modes de comptage de divergences et plusieurs schémas de branchement ont été testés.

Des nouvelles méthodes de recherche locale basées sur la méthode LDS ont été proposées et comparées avec des méthodes semblables. Les résultats expérimentaux montrent que les méthodes sont efficaces pour la résolution des problèmes à machines parallèles en général et prouvent aussi, au moins pour le problème étudié, l'intérêt d'incorporer la méthode LDS dans un schéma de recherche locale à grands voisinages. Les résultats obtenus dans Néron et al. (2008) pour le problème standard à machines parallèles ont été améliorés. Nous avons aussi obtenu de bons résultats, notamment pour le critère de minimisation de la somme des dates de fin d'exécution des tâches, lors de comparaisons entre nos propositions et la recherche locale à grands voisinages implémentée dans le solveur IBM-ILOG CP Optimizer 6.0 (Laborie and Godard, 2007).

Enfin, d'un point de vue plus théorique, nous avons proposé une extension du raisonnement énergétique pour intégrer les temps de préparation dans le raisonnement. De nouvelles règles de dominance adaptées pour être intégrées dans un schéma de recherche locale ont également été

proposées. Les résultats expérimentaux ont montré que ces propositions permettent de réduire de manière très sensible le nombre de nœuds explorés, ainsi que le temps d'exécution.

Conclusion et perspectives

Dans ce manuscrit, nous avons présenté le travail réalisé pendant cette thèse. Deux sujets clairement différenciés ont été traités. Dans la première, et plus importante, partie du manuscrit, la problématique liée à l'ordonnancement des transports a été abordée d'un point de vue interdisciplinaire. Nous nous sommes focalisé sur la conception d'un système d'aide à la décision dédié à la résolution de ce type de problèmes. Dans l'approche que nous avons proposée, des méthodes issues de l'ingénierie cognitive comme l'analyse du domaine de travail et la conception d'interfaces écologiques ont été combinées à des mécanismes de résolution. La deuxième partie consiste en une analyse focalisée sur la résolution de problèmes théoriques d'ordonnancement à machines parallèles avec contraintes de précédence et temps de préparation des machines entre opérations.

En ce qui concerne la conception du système d'aide à la décision, nous avons d'abord présenté une analyse du domaine de travail pour l'ordonnancement des transports. Pour cela, nous avons suivi une démarche fondée sur l'énumération des différentes variantes du problème (académiques ou réelles) décrites dans la littérature. Ceci nous a permis d'identifier les objets, les contraintes et les objectifs qui interviennent dans la résolution de ce type de problèmes. L'avantage d'une telle approche est que l'analyse du domaine proposée reste indépendante vis-à-vis des moyens et des événements. Cette analyse a été réalisée à travers une hiérarchie d'abstraction. Elle permet de décrire le système sous différents angles selon les besoins de l'opérateur. De plus, les liens entre les différents niveaux de la hiérarchie ont été également établis. Ces deux caractéristiques permettent aux interfaces conçues à partir d'une telle analyse d'être pertinentes pour l'opérateur d'un point psychologique.

L'analyse du domaine réalisée est la base d'une architecture générique d'un système d'aide à la décision de résolution de problèmes d'ordonnancement des transports. Dans ce contexte, nous

avons proposé un mécanisme de résolution fondé sur des méthodes d'optimisation issues de la recherche opérationnelle et un ensemble d'interfaces homme-machine. Les deux éléments partagent une vision commune du problème, ce qui fait que le mécanisme de résolution et l'opérateur humain interagissant au moyen des interfaces manipulent les mêmes objets physiques et les mêmes contraintes. Une analyse des tâches à réaliser pour résoudre le problème a permis de définir la structure de ce mécanisme de résolution. La démarche proposée permet de concevoir un système homme-machine capable d'aider l'opérateur humain dans la réalisation de chacune des tâches qui doivent ou peuvent être réalisées. L'opérateur humain dispose, d'un côté, d'algorithmes performants capables de résoudre automatiquement tous les aspects bien formalisables et prévisibles du problème et, de l'autre côté, d'outils qui facilitent la coopération homme-machine en autorisant la prise en compte d'aspects peu ou pas formalisés ou non prévus à l'avance. Enfin, dans cette même logique d'intégration de l'opérateur humain dans le processus de résolution, une autre originalité du système proposé est qu'il permet à l'opérateur de choisir entre différents modes de contrôle. En effet, ces problèmes sont classiquement résolus soit avec une approche complètement manuelle, soit avec une approche entièrement informatisée. L'intégration d'un mode de contrôle hybride ou interactif permet à l'opérateur humain de prendre les décisions liés aux aspects non-formalisables ou non-prévus pour après laisser la main aux algorithmes d'optimisation pour le reste des décisions à prendre.

Des algorithmes spécifiques pour chaque sous-tâche ont été proposés. L'efficacité de ces algorithmes a été testée sur des instances de la littérature. Ces propositions peuvent cependant être améliorées. Dans ce contexte, deux pistes intéressantes à court terme sont, en premier lieu, de travailler sur de nouveaux algorithmes mieux adaptés à chaque sous-tâche du problème. Nous pensons que l'intégration d'algorithmes plus complexes peut en effet aider à améliorer la qualité des solutions. Le deuxième aspect qui mérite aussi d'être développé concerne les mécanismes de propagation de contraintes intégrés dans le système. Les mécanismes actuels sont très légers, ce qui ne permet pas une détection rapide des inconsistances lors de la résolution du problème. Nous croyons donc que la performance du mécanisme de résolution peut être encore fortement améliorée.

Les interfaces homme-machine du système d'aide à la décision ont été conçues à partir des principes définis par la méthode de conception d'interfaces écologiques. Nous avons proposé des interfaces pour faciliter la coopération homme-machine pour chacune des tâches à réaliser. Les in-

terfaces proposées affichent l'information nécessaire pour assister l'opérateur dans le processus de prise de décision. Cette information est affichée de manière à stimuler une résolution du problème basé sur le couplage perception-action. La performance et l'efficacité des outils d'interaction homme-machine d'une des interfaces proposées ont été évaluées au travers d'une expérience. Les résultats de celle-ci nous ont permis d'extraire des conclusions intéressantes pour la conception des interfaces. En premier lieu, nous avons pu corroborer un incrément de l'efficacité (réduction de la charge mentale, accélération du processus de prise de décision et incrément de l'efficacité) de la coopération homme-machine lorsque les interfaces favorisent une résolution fondée sur la perception plutôt que sur l'analyse. Ceci est en accord avec les avantages anticipés par les auteurs de la méthode de conception. La deuxième conclusion issue des résultats de l'expérience est qu'il faut trier l'information à afficher sur l'interface pour chaque tâche à réaliser. Nous avons pu vérifier que l'excès d'information peut pénaliser la coopération homme-machine. Pour arriver à une conception d'un système efficace, il est alors nécessaire de mener une phase d'expériences ou d'entretiens avec les opérateurs, de manière à déterminer les modes d'affichage les plus efficaces et l'information nécessaire à afficher pour favoriser le processus de résolution. Dans ce contexte, nous observons qu'une validation des autres interfaces et des mécanismes d'interaction proposés dans le système s'impose.

Une autre originalité apportée dans ce travail réside en la proposition d'une nouvelle approche pour assister l'opérateur dans la tâche de relaxation de contraintes lorsque le problème est ou devient non-réalisable. Nous avons défini la structure du mécanisme pour la conception de cet outil d'aide à l'opérateur. L'approche est fondée sur le principe d'inversion de modèle. Le mécanisme a été appliqué pour un cas simple de problème non-réalisable. Nous avons évalué l'hybridation de l'inversion de modèle avec les méthodes de classification issues de l'analyse de données. Bien que les résultats obtenus soient assez prometteurs, il est nécessaire de mieux évaluer cette approche, d'une part sur des instances plus réalistes et, d'autre part, dans un contexte de coopération homme-machine. Ce nouveau cadre de travail peut être, en effet, une piste très intéressante à suivre dans l'avenir.

Enfin, dans la deuxième partie du document nous avons proposé de nouvelles méthodes de résolution pour un problème d'ordonnancement à machines parallèles avec considération de contraintes de précédence et temps de préparation. Les techniques proposées sont des variantes des algorithmes hybrides de recherche locale et de recherche arborescente décrits dans la

littérature. Ces algorithmes ont été évalués sur des instances de la littérature. Les performances obtenues sont bonnes, ce qui nous a encouragé à utiliser ces algorithmes pour la résolution des problèmes de tournées de véhicules. Une perspective de recherche à court terme est d'étendre les mécanismes proposés pour résoudre des problèmes plus complexes comme le flow-shop hybride ou le problème d'ordonnancement de projet à moyens limités (RCPSP).

Bibliographie

- C. Artigues, P. Lopez, and P-D. Ayache. Schedule generation schemes and priority rules for the job-shop problem with sequence-dependent setup times: dominance properties and computational analysis. *Annals of Operations Research*, 138(1):21–52, 2005.
- B. De Backer, V. Furnon, P. Kilby, P. Prosser, and P. Shaw. Solving Vehicle Routing Problems using Constraint Programming and Metaheuristics. *Journal of Heuristics*, 6:501–523, 2000.
- I. D. Baev, W. M. Meleis, and A. Eichenberger. An experimental study of algorithms for weighted completion time scheduling. *Algorithmica*, 22:34–51, 2002.
- K. Baker. *Introduction to sequencing and scheduling*. John Wiley & Sons, New York (USA), 1974.
- M. Balinski and R. Quandt. On an integer program for a delivery problem. *Operations Research*, 12:300–304, 1964.
- C. Barnhart and G. Laporte, editors. *Handbooks in operational research and management science: Transportation*, volume 14. North-Holland, 2007.
- C. Basnet, L. Foulds, and M. Igbaria. FleetManager: a microcomputer-based decision support system for vehicle routing. *Decision Support systems*, 16(3):195–207, 1996.
- J. C. Beck and L. Perron. Discrepancy-Bounded Depth First Search. In *Second International Workshop on Integration of AI and OR Technologies for Combinatorial Optimization Problems (CP-AI-OR'00)*, Paderborn (Germany), 2000.
- M. Bellmore and S. Hong. Transformation of multisalesman problem to the standard traveling salesman problem. *Journal of the ACM*, 21(3):500–504, 1974.
- E. Beltrami and L. Bodin. Networks and vehicle routing for municipal waste collection. *Networks*, pages 568–581, 1974.
- C. Blanche. *Contribution à l'Ingénierie Cognitive pour l'évaluation et la conception de Systèmes d'Information: Application au livre électronique et à une base de données d'archive*. PhD thesis, Université de Bretagne Sud, 2010.

- L. Bodin, B. Golden, A. Assad, and M. Ball. Routing and Scheduling of Vehicles and Crews - The State of the Art. volume 10, pages 63–212. *Computers & Operations Research*, 1983.
- C. M. Burns. Navigation strategies with ecological displays. *International Journal of Human-Computer Studies*, 1(52):111–129, 2000.
- C. M. Burns and J. R. Hajdukiewicz. *Ecological Interface Design*. Taylor and Francis Group, CRC Press, 2004.
- C. M. Burns, G. Skraaning, G. A. Jamieson, N. Lau, J. Kwok, R. Welch, and G. Andresen. Evaluation of Ecological Interface Design for Nuclear Process Control: Situation Awareness Effects. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 50(17): 663–679, 2008.
- R. G. Busacker and P. J. Gowen. A procedure for determining minimal-cost network flow patterns. Technical report, ORO Technical Rep. 15, Operational Research Office, John Hopkins University, Baltimore, 1961.
- Y. Caseau, F. Laburthe, and G. Silverstein. A meta-heuristic factory for vehicle routing problems (meta-programming for meta-heuristics). In J. Jaffar, editor, *Lecture Notes in Computer Science*, volume 1713, pages 144–158. Springer, 1999. Principles and Practice of Constraint Programming (CP'99).
- J. Cegarra. A cognitive typology of scheduling situations: A contribution to laboratory and field studies. *Theoretical Issues in Ergonomics Science*, 9(3):201–222, 2008.
- J. Cegarra and J.-M. Hoc. The role of algorithm and result comprehensibility of automated scheduling on complacency. *Human Factors and Ergonomics in Manufacturing*, 28(6):603–620, 2008.
- J. Cegarra and W. van Wezel. A comparison of task analysis methods for planning and scheduling. In Jan C. Fransoo, Toni Waefer, and John R. Wilson, editors, *Behavioral Operations in Planning and Scheduling*. Springer, 2010.
- J. Cegarra, B. Gacias, and P. Lopez. Implications of technological changes in vehicle routing systems for planners' constraint processing. *Submitted to Human Factors and Ergonomics in Manufacturing & Service Industries*, 2010.

- C. Chauvin and J.-M. Hoc. In P. Millot, editor, *Ergonomie des systèmes homme machine*, chapter “Intégration de l’ergonomie dans la conception des systèmes homme-machine”. Paris: Hermès, sous presse, 2010.
- Z.-L. Chen and W. B. Powell. Solving Parallel Machine Scheduling Problems by Column Generation. *INFORMS Journal on Computing*, 11(1):78–94, 1999.
- T. Cheng and C. Sin. A State-of-the-Art Review of Parallel-Machine Scheduling Research. *European Journal of Operational Research*, 47:271–292, 1990.
- G. Clarke and J. V. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581, 1964.
- J. Cohen. Statistics a power primer. *Psychology Bulletin*, pages 155–159, 1992.
- J-F Cordeau, M. Gendreau, A. Hertz, G. Laporte, and J-S. Sormany. In *Logistics Systems: Design and Optimization*, chapter “New Heuristics for the Vehicle Routing Problem”, pages 279–297. Springer US, 2005.
- G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.
- F. D. Davis and J. E. Kottemann. Determinants of decision rule use in a production planning task. *Organizational Behavior and Human Decision Processes*, 63(2):145–157, 1995.
- C. de Snoo, M. Hoogenraad, and J.C. Wortmann. Opportunities for collaborative planning in freight transport planning. In *15th International Annual EurOMA Conference (EurOMA’08)*, pages 15–18, Groningen (The Netherlands), 2008.
- R. Dechter. *Constraint Processing*. Morgan Kaufmann, San Francisco, USA, 2003.
- E. L. Demeulemeester and W. S. Herroelen. New benchmark results for the resource-constrained project scheduling problem. *Management Science*, 43(11):1485–1492, 1997.
- E. L. Demeulemeester, M. Vanhoucke, and W. Herroelen. Rangen: A random network generator for activity-on-the-node networks. *Journal of Scheduling*, (6):17–38, 2003.

- M. Desrochers, J. K. Lenstra, M. W. P. Savelsbergh, and F. Soumis. In B. L. Golden and A. A. Assad, editors, *Vehicle Routing: Methods and Studies*, chapter “Vehicle routing with time windows: Optimization and approximation”, pages 65–84. North-Holland, Amsterdam, 1988.
- M. Desrochers, J. K. Lenstra, and M. W. P. Savelsbergh. A classifications scheme for vehicle routing and scheduling problems. *European Journal of Operations research*, 46:322–332, 1990.
- M. Desrochers, C. V. Jones, J. K. Lenstra, M. W. P. Savelsbergh, and L. Stougie. Towards a model and algorithm management system for vehicle routing and scheduling problems. *Decision Support Systems*, 25(2):109–133, 1999.
- D. Diaper. In D. Diaper and N. Stanton, editors, *The Handbook of Task Analysis for Human-Computer*, chapter “Undertanding Task Analysis for Human-Computer Interaction, pages 5–47. Mahwah, NJ: Lawrence Erlbaum Associates, 2004.
- E. Diday. Une nouvelle méthode en classification automatique et reconnaissance des formes : la méthode des nuées dynamiques. *Revue de Statistique Appliquée*, 19(2):19–33, 1971.
- C. Dimopoulos and A. Ioannides. Designing support for the human scheduler: a case study and a systematic evaluation of the conventional production research perspective. sous presse, 2010.
- M. L. Fisher and R. Jaikumar. A generalized assignment heuristic for the vehicle routing problem. *Networks*, 11:109–124, 1981.
- E. W. Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, (3):768, 1965.
- B. Gacias, J. Cegarra, and P. Lopez. An interdisciplinary method for a generic vehicle routing problem decision support system. In *International Conference on Industrial Engineering and Systems Management (IESM 2009)*, Montreal (Canada), 2009.
- B. Gacias, C. Artigues, and P. Lopez. Parallel machine scheduling with precedence constraints and setup times. *Computers & Operations Research*, 37(12):2141–2151, 2010a.
- B. Gacias, J. Cegarra, and P. Lopez. Work domain analysis and ecological interface for the vehicle routing problem. In *11th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Human-Machine System*, Valenciennes (France), 2010b.

-
- B. Gacias, P. Lopez, and J. Cegarra. A decision support system for vehicle routing based on model inversion and data analysis. In *International Conference of Modeling and Simulation (MOSIM 2010)*, Hammamet (Tunisie), 2010c.
- H. Gehring and J. Homberger. A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. In K. Miettinen, M. Mäkelä, and J. Toivanen, editors, *Proceedings of EUROGEN'99*, Finland, 1999.
- E. Gelle and B. Faltings. Solving mixed and conditional constraint satisfaction problems. *Constraints*, 8(2):107–141, 2003.
- M. Gendreau and C. D. Tarantilis. Solving large-scale vehicle routing problems with time windows: The state-of-the-art. CIRRELT, CIRRELT-2010-04, 2010.
- J. J. Gibson. *The ecological approach to visual perception*. Boston: Houghton Mifflin, 1979.
- B. E. Gillet and L. R. Miller. A heuristic algorithm for the vehicle dispatch problem. *Operations Research*, 22(2):340–349, 1974.
- M. Goetschalckx and C. Jacobs-Blecha. The vehicle routing problem with backhauls. *European Journal of Operational Research*, 42:39–51, 1989.
- B. Golden, S. Raghavan, and E. Wasil, editors. *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43. Springer, 2008.
- B. L. Golden, J. DeArmon, and E. K Baker. Computational experiments with algorithms for a class of routing problems. *Computers & Operations Research*, 10(1):47–59, 1983.
- B. L. Golden, E. Baker, J. Alfaro, and J. Schaffer. The vehicle routing problem with backhauling: Two approaches. In *Proceedings of the XXI Annual Meeting of S.E TIMS*, pages 90–92, Myrtle Beach (SC), 1985.
- B. L. Golden, A. A. Assad, and E. A. Wasil. Routing vehicles in the real world: Applications in the solid waste, beverage, food, dairly, and newspaper industries. In P. Toth and D. Vigo, editors, *The Vehicle Rouring Problem. SIAM Monographs on Discrete Mathematics and Applications*, pages 245–286, 2002.

- R.L. Graham, E.L. Lawler, J.K. Lenstra, and A. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, pages 287–326, 1979.
- G. Gutin and A. P. Punnen. *The Traveling Salesman Problem and Its Variations*. Kluwer Academic Publishers, 2002.
- J. R. Hajdukiewicz, C. M. Burns, K. J. Vicente, and R. G. Eggleston. Work domain analysis for intentional systems. In *43rd Annual Meeting Human Factors and Ergonomics Society*, pages 333–337, 1999.
- P. Hansen and N. Mladenovic. Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, (130):449–467, 2001.
- S. G. Hart and L. E. Staveland. In P. A. Hancock and N. Meshkati, editors, *Human Mental Workload*, chapter Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research, pages 139–184. Elsevier Science Publishers, North-Holland, 1988.
- W. D. Harvey and M. L. Ginsberg. Limited discrepancy search. In *14th International Joint Conference on Artificial Intelligence (IJCAI'96)*, 1996.
- P. G. Higgins. In *Human performance in planning and scheduling: fieldwork studies, methodologies and research issues*, chapter “Architecture and Interface Aspects of Scheduling Decision Support”, pages 245–279. Taylor and Francis, London, 2001.
- P. G. Higgins. Interaction in hybrid intelligent scheduling. *International Journal of Human Factors in Manufacturing*, 6:185–203, 1996.
- P. G. Higgins. *Job Shop Scheduling: Hybrid Intelligent Human-Computer Paradigm*. PhD thesis, University of Melbourne, Australia, 1999.
- F. L. Hitchcock. The distribution of a product from several sources to numerous localities. *Journal of Mathematics and Physics*, 20:224–230, 1941.
- A. Hmida, M. J. Huguet, P. Lopez, and M. Haouari. Climbing depth-bounded discrepancy search for solving hybrid flow shop scheduling problems. *European Journal of Industrial Engineering*, 1(2):223–243, 2007.

- J.-M. Hoc. In P. Millot, editor, *Ergonomie des systèmes homme machine*, chapter “Coopération homme-machine. Une approche fonctionnelle”. Paris: Hermès, sous presse, 2010.
- J.-M. Hoc, N. Mebarki, and J. Cegarra. L’assistance à l’opérateur humain pour l’ordonnancement dans les ateliers manufacturiers. *Le travail humain*, 67:181–208, 2004.
- J. Hurink and S. Knust. List scheduling in a parallel machine environment with precedence constraints and setup times. *Operations Research Letters*, (29):231–239, 2001.
- E. L Hutchins, J. D. Hollan, and D. A. Norman. In D. A. Norman and S. W. Draper, editors, *User Centered System Design: New Perspectives on Human Computer Interaction*, chapter “Direct manipulation interfaces”, pages 87–124. NJ: Erlbaun, 1986.
- J. Itoh, A. Sakuma, and K. Monta. An ecological interface for supervisory control of BWR nuclear power plants. *Control Engineering Practice*, 3:231–239, 1995.
- F. J. Jüngen and W. Kowalczyk. An intelligent interactive project management support system. *European Journal of Operational Research*, pages 60–81, 1995.
- N. Jussien. e-constraints: explanation-based constraint programming. In *Workshop on User-Interaction in Constraint Satisfaction (CP’01)*, Paphos (Cyprus), 2001.
- W. Karoui, M.-J. Huguet, P. Lopez, and W. Naanaa. YIELDS: A Yet Improved Limited Discrepancy Search for CSPs. In *4th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR’07)*, LNCS 4510, pages 99–111, Brussels (Belgium), 2007. Springer.
- P. Kilby, P. Prosser, and P. Shaw. A comparison of traditional and constraint-based heuristic methods on vehicle routing problems with side constraints. *Constraints*, 5(4):389–414, 2000.
- B. Kim, S. Kim, and S. Sahoo. Waste collection vehicle routing problem with time windows. *Computers & Operations Research*, 33(12):3624–3642, 2006.
- A. M. Kinsley, L. Sharit, and K. J. Vicente. In P. T. Kidd & W. Karwowski, editor, *Advances in Agile Manufacturing*, chapter “Abstraction hierarchy representation of manufacturing: towards ecological interfaces for advanced manufacturing systems”, pages 297–300. IOS Press, Amsterdam (The Netherlands), 1994.

- C. Kolski and H. Ezzedine. Conception et évaluation des IHM de supervision: éléments méthodologiques. *Revue Génie Logiciel*, 65:2–11, 2003.
- R. Korf. Improved limited discrepancy search. In *13th AAAI*, 1996.
- S. P. Krosner, C. M. Mitchell, and T. Govindaraj. Design of an FMS operator workstation using the Rasmussen abstraction hierarchy. In *IEEE: International Conference on Systems, Man, and Cybernetics*, pages 959–964, New York (USA), 1989.
- P. Laborie. IBM ILOG CP Optimizer for Detailed Scheduling Illustrated on Three Problems. In Willem Jan van Hoeve and John N. Hooker, editors, *6th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR'09)*. *Lecture Notes in Computer Science*, number 5547, pages 148–162, Pittsburgh (USA), 2009.
- P. Laborie and D. Godard. Self-adapting large neighborhood search: Application to single-mode scheduling problems. In P. Baptiste, G. Kendall, A. Munier-Kordon, and F. Sourd, editors, *3rd Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA'07)*, pages 276–284, Paris (France), 2007.
- P. Lacomme, C. Prins, and M. Sevaux. *Algorithmes de graphes*. EYROLLES, 2003.
- G. Laporte and Y. Nobert. Exact algorithms for the vehicle routing problem. *Annals of Discrete Mathematics*, 31:147–184, 1987.
- G. Laporte, Y. Nobert, and M. Desrochers. Optimal routing under capacity and distance restrictions. *Operations Research*, 33:1050–1073, 1985.
- E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Khan, and D. B. Shmoys. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. John Wiley & Sons, 1985.
- J. K. Lenstra and A. H. G. Rinnooy Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 1981.
- M.-L. Levy. *Methodes par decomposition temporelle et problèmes d'ordonnancement*. PhD thesis, Institut National Polytechnique de Toulouse, 1996.

- F.-H. F. Liu and S.-Y. Shen. A route-neighborhood-based metaheuristic for vehicle routing problem with time windows. *European Journal of Operational Research*, 118(3):485–504, 1999.
- S. P. Lloyd. Least square quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
- P. Lopez. Approche par contraintes des problèmes d’ordonnancement et d’affectation: structures temporelles et mécanismes de propagation. *HDR, INP, Toulouse*, 2003.
- P. Lopez and P. Esquirol. Consistency enforcing in scheduling: A general formulation based on energetic reasoning. In *5th International Workshop on Project Management and Scheduling (PMS’96)*, pages 155–158, Poznan (Poland), 1996.
- N. F. Matsatsinis. Towards a decision support system for the ready concrete distribution system: A case of a Greek company. *European Journal of Operational Research*, 152(2):487–499, 2004.
- J. E. Mendoza, A. L. Medaglia, and N. Velasco. An evolutionary-based decision support system for vehicle routing: The case of a public utility. *Decision Support Systems*, 46(3):730–742, 2009.
- M. D. Mesarovic, D. Macko, and Y. Takahara. *Theory of hierarchical, multilevel, systems*. Academic Press, New York, 1970.
- D. M. Mietus. *Understanding planning for effective decision support*. PhD thesis, University of Groningen, 1994.
- M. Milano and A. Roli. On the relation between complete and incomplete search: an informal discussion. In *Forth International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR’02)*, Le Croisic (France), 2002.
- C. E. Miller, A.W. Tucker, and R.A. Zemlin. Integer programming formulations and traveling salesman problems. *Journal of the ACM*, 7:326–329, 1960.
- S. Mittal and B. Falkenhainer. Dynamic constraint satisfaction problems. In *Proceedings of AAAI-90, Boston, MA*, pages 25–32, 1990.

- N. Moray. The cognitive psychology and cognitive engineering of industrial systems. In F.T. Durso, R.S. Nickerson, R.W. Schvaneveldt, S.T. Dumais, D.S. Lindsay, and M.T.H. Chi, editors, *Handbook of Applied Cognition*, pages 209–245. New York: John Wiley & Sons Ltd, 1999.
- N. Moray, M. I. Dessouky, and R. Adapathya. Strategic behavior, workload, and performance in task scheduling. *Human Factors*, 33:607–629, 1991.
- N. Moray, D. Hiskes, J. Lee, and B. Muir. In J. M. Hoc, P. C. Cacciabue, and E. Hollnagel, editors, *Expertise and Technology: Cognition and Human-Computer Interaction*, chapter “Trust and human intervention in automated systems”. NJ: Lawrence Erlbaum, 1995.
- T. Morineau, C. Blanche, L. Tobin, and N. Guéguen. The emergence of the contextual role of the e-book in cognitive processes through an ecological and functional analysis. *International Journal of Human-Computer Studies*, 62(3):329–348, 2005.
- Y. Nagata, O. Bräysy, and W. Dullaert. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, 37(4):724–737, 2010.
- G. L. Nemhauser and L. A. Wosley. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1999.
- E. Néron, F. Tercinet, and F. Sourd. Search tree based approaches for parallel machine scheduling. *Computers & Operations Research*, 35(4):1127–1137, 2008.
- R. Nessah, Ch. Chu, and F. Yalaoui. An exact method for $Pm/sds, r_i / \sum C_i$ problem. *Computers & Operations Research*, (34):2840–2848, 2005.
- K. Neumann, C. Schwindt, and J. Zimmermann. *Project Scheduling with Time Windows and Scarce Resources*. Springer, 2002.
- I. Or. *Traveling salesman-type combinatorial problems and their relation to the logistics of the regional blood banking*. PhD thesis, Northwestern university, Evanston IL (USA), 1976.
- C. Orloff. Routing a fleet of m vehicles to/from a central facility. *Networks*, 4(2):147–162, 1974.

- T. C. Ormerod and E. P. Chronicle. Global perceptual processing in problem solving: The case of the travelling salesperson. *Perception & Psychophysics*, 61(6):1227–1238, 1996.
- I. Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41(4):421–451, 1993.
- I. M. Ovacik and R. Uzsoy. Rolling horizon procedures for dynamic parallel machine scheduling with sequence-dependent setup times. *International journal of production research*, 33(11):3173–3192, 1995.
- W. L. Pearn, S. H. Chung, and C .M. Lai. Scheduling Integrated Circuit Assembly Operations on Die Bonder. *IEEE Transactions on electronics packaging manufacturing*, 30(2), 2007.
- G. Pesant and M. Gendreau. A constraint programming framework for local search methods. *Journal of Heuristics*, 5:255–279, 1999.
- E. Prescott-Gagnon, G. Desaulniers, and L.-M. Rousseau. A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows. *Networks*, 54(4):190–204, 2009.
- C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985–2002, 2004.
- J. Rasmussen. *Information processing and human-machine interaction*. North-Holland: Elsevier, 1986a.
- J. Rasmussen. In r. flin and e. salas and m. strub and l. martin (eds.) merging Paradigms: Decision Making, Management, and Cognitive Control, 1997.
- J. Rasmussen. The role of hierarchical knowledge representation in decision making and system management. In *IEEE: International Conference on Systems, Man, and Cybernetics*, number SMC-15, pages 234–243, 1985.
- J. Rasmussen. Skills, rules, knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics*, 13:257–266, 1983.

- J. Rasmussen. *Information processing and human-machine interaction : an approach to cognitive engineering*. New York : North-Holland, 1986b.
- J. Rasmussen and L. P. Goodstein. Decision support in supervisory control of high-risk industrial systems. *Automatica*, 23(5):663–671, 1987.
- J. Rasmussen and K. J. Vicente. Coping with human errors through system design: Implications for ecological interface design. *International Journal of Man-Machine Studies*, 31:517–534, 1989.
- J. Rasmussen, A. M. Pejtersen, and L. P. Goodstein. *Cognitive Systems Engineering*. Wiley, New York, 1994.
- J. J. Ray. A web-based spatial decision support system optimizes routes oversize/overweight vehicles in Delaware. *Decision Support Systems*, 43:1171–1185, 2007.
- F. Rossi, P. van Beek, and T. Walsh. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier Science Inc., New York, NY, USA, 2006.
- E. M. Roth, E. S. Patterson, and R. J. Mumaw. In J. J. Marciniak, editor, *Encyclopedia of Software Engineering, 2^o Edition*, chapter “Cognitive Engineering: Issues in User-Centered System Design”, pages 110–123. John Wiley & Sons, New York: Wiley-Interscience, 1994.
- R. Ruiz, C. Maroto, and J. Alcaraz. A Decision Support System for a Real Vehicle Routing Problem. *European Journal of Operational research*, 153(3):593–606, 2004.
- A. Salem, G. C. Anagnostopoulos, and G. Rabadi. A branch-and-bound algorithm for parallel machine scheduling problems. *Society for Computer Simulation International (SCS)*, pages 88–93, 2000.
- L. Santos, J. Coutinho-Rodrigues, and J. R Current. Implementing a multi-vehicle multi-route spatial decision support system for efficient trash collection in Portugal. *Transportation Research*, 42 (Part A):922–934, 2008.
- G. Saporta. *Probabilité, analyse des données et statistique*. Technip, Paris, 1990.
- M. W. P. Savelsbergh. The Vehicle Routing Problem with Time Windows: Minimizing Route Duration. *INFORMS JOURNAL ON COMPUTING*, 4(2):146–154, 1992.

-
- G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt, and G. Dueck. Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, 159:139–171, 2000.
- J. M. J. Schutten. List scheduling revisited. *Operations Research Letters*, (18):167–170, 1994.
- J. M. J. Schutten and R. A. M. Leussink. Parallel machine scheduling with release dates, due dates and family setup times. *International journal of production economics*, 46-47(1):119–125, 1996.
- P. Shaw. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In *Principles and Practice of Constraint Programming (CP'98)*, 1998.
- B. Shneiderman. Direct manipulation: a step beyond programming languages. *IEEE Computer*, 16(8):57–69, 1983.
- M. M. Solomon. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research*, 35:254–265, 1987.
- M. M. Solomon. *Vehicle Routing and Scheduling with Time Window Constraints: Models and Algorithms*. PhD thesis, University of Pennsylvania, USA, 1983.
- F. Sourd. Earliness-tardiness scheduling with setup considerations. *Computers & Operations Research*, 32(7):1849–1865, 2005.
- A. Sprecher, R. Kolisch, and A. Drexl. Semi-active, active, and non-delay schedules for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 80(1):94–102, 1995.
- J. Sveska and V. Huckfeldt. Computational experience with an m-salesman traveling salesman algorithm. *Management Science*, 19(7):790–799, 1973.
- M. M. Syslo, N. Deo, and J. S. Kowalik, editors. *Discrete Optimization Algorithms*. Prentice-Hall, Englewood Cliffs, NJ (USA), 1983.
- E. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J.-Y. Potvin. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, 31(2):170–186, 1997.

- C. D. Tarantilis and C. T. Kiranoudis. Using a spatial decision support system for solving the vehicle routing problem. *Information & Management*, 39(5):359–375, 2002.
- P. Terrier and J. M. Cellier. Depth of processing and design-assessment of ecological interfaces: task analysis. *International Journal of Human-Computer Studies*, pages 287–307, 1999.
- P. Toth and D. Vigo. An exact algorithm for the vehicle routing problem with backhauls. *Transportation Science*, 31:372–385, 1997.
- P. Toth and D. Vigo, editors. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, Philadelphia (USA), 2001.
- R. Uzsoy and J. D. Velasquez. Heuristics for minimizing maximum lateness on a single machine with family-dependent set-up times. *Computers & Operations Research*, (35):2018–2033, 2008.
- M. van den Akker, J. Hoogeveen, and J. Kempen. Parallel machine scheduling through column generation: minimax objective functions, release dates, deadlines and/or generalized precedence constraints. Technical report, Utrecht university, 2005.
- R. van Paassen. New visualisation techniques for industrial process control. In *6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems*, pages 457–462, Cambridge, MA (USA), 1995. Pergamon Press, Oxford.
- W. van Wezel, J. Cegarra, and J.-M. Hoc. Allocating function to humans and algorithms in scheduling. In Jan C. Fransoo, Toni Waefer, and John R. Wilson, editors, *Behavioral Operations in Planning and Scheduling*. Springer, 2010.
- K. J. Vicente. Ecological Interface Design: Supporting operator adaptation, continuous learning, distributed, collaborative work. pages 93–97, 1999a.
- K. J. Vicente. *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ (USA), 1999b.
- K. J. Vicente. Wanted: psychologically relevant, device- and event-independent work analysis techniques. *Interacting with Computers*, 11:237–254, 1999c.
- K. J. Vicente and J. Rasmussen. Ecological Interface Design: Theoretical Foundations. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(4):589–606, 1992.

- T. Walsh. Depth-bounded Discrepancy Search. *APES Group, Department of Computer Science*, 1997.
- O. Watanabe, K. Takaura, Y. Fujita, and Y. Hayashi. Evaluation of ecological interface design. In Y. Anzai, K. Ogawa, and H. Mori, editors, *Symbiosis of Human and Artifact - Future Computing and Design for Human-Computer Interaction*, volume 20, Part 1 of *Advances in Human Factors/Ergonomics*, pages 977–982. Elsevier, 1995. Proceedings of the Sixth International Conference on Human-Computer Interaction (HCI'95).
- W. B. L. Wong, P. J. Sallis, and D. O'Hare. The Ecological Approach to Interface Design: Applying the Abstraction Hierarchy to Intentional Domains. In P. Calder & B. Thomas, editor, *8th Australian Conference on Computer-Human Interaction (OzCHI'98)*, pages 144–151, Adelaide (Australia), 1998. IEEE Computer Society Press.
- X. Yu and K. Liu. A spatial decision support system for large scale vehicle routing. In *Proceedings of the 2009 International Conference on Measuring Technology and Mechatronics Automation (ICMTMA'09)*, pages 444–449, Washington, DC, USA, 2009. IEEE Computer Society.

Table des matières

Introduction	1
I Systèmes d'aide à la décision pour l'ordonnancement des transports	5
I.1 Problèmes de décision	5
I.2 L'ordonnancement des transports	6
I.2.1 Définition du problème de transport	7
I.2.2 Définition du problème de tournées de véhicules (Vehicle Routing Problem - VRP)	9
I.2.3 Variantes du problème de tournées de véhicules	10
I.2.4 Systèmes d'aide à la décision pour les problèmes de tournées de véhicules	15
I.3 Conception de systèmes interactifs	17
I.3.1 Le système homme-machine	17
I.3.2 Les approches existantes pour la conception de systèmes	18
I.4 Une approche interdisciplinaire	22
I.5 Conclusion	25
II Le domaine de travail	27
II.1 Introduction	27
II.2 Analyse du domaine de travail	27
II.2.1 Analyse du domaine de travail dans les problèmes d'ordonnancement . . .	31
II.3 Approche proposée pour l'analyse du domaine	34
II.3.1 Hiérarchie d'abstraction	35
II.3.2 Analyse du domaine de validité de la décomposition	42

II.4 Conclusion	47
III Architecture décisionnelle pour l'ordonnancement des transports	49
III.1 Introduction	49
III.2 Synoptique de l'architecture	50
III.3 Analyse des tâches pour la résolution du problème	51
III.3.1 Identification des tâches pour la résolution du problème	51
III.3.2 Répartition des fonctions entre l'homme et la machine	55
III.4 Mécanisme de résolution	59
III.4.1 Programmation par contraintes	59
III.4.2 Algorithmes spécifiques pour la résolution du problème	60
III.4.3 Définition formelle du problème	61
III.4.4 Algorithmes pour la sélection de véhicules	63
III.4.5 Algorithmes pour l'affectation des clients aux véhicules	68
III.4.6 Algorithmes pour la sélection des itinéraires	71
III.5 Résultats expérimentaux	77
III.6 Conclusion	80
IV Les interfaces homme-machine	83
IV.1 Introduction	83
IV.2 Interfaces écologiques	83
IV.3 Les interfaces du système	86
IV.3.1 Interfaces pour la modélisation du problème	86
IV.3.2 Interfaces pour la résolution du problème	89
IV.3.3 Interface pour la sélection de la solution finale	95
IV.3.4 Interface pour la poursuite en temps-réel des tournées	98
IV.4 Évaluation de l'interface pour la sélection des solutions	99
IV.4.1 Description du problème	99
IV.4.2 Résolution du problème	100
IV.4.3 Présentation de l'expérience	101
IV.4.4 Résultats expérimentaux	103
IV.5 Conclusion	109

V Assistance à l'opérateur dans la phase de résolution	111
V.1 Introduction	111
V.2 Problèmes non-réalisables	111
V.3 Inversion de modèle	114
V.3.1 Méthodes de classification de données	120
V.4 Résultats expérimentaux	127
V.5 Conclusion	130
VI Résolution d'un problème d'ordonnement	131
VI.1 Introduction	131
VI.2 Problèmes d'ordonnement	131
VI.2.1 Définitions	132
VI.3 Problème à machines parallèles	134
VI.3.1 Relation avec les problèmes d'ordonnement des transports	135
VI.3.2 Définition du problème	136
VI.3.3 Analyse des solutions	138
VI.4 Recherche arborescente	142
VI.4.1 Structure arborescente	142
VI.4.2 Evaluation d'un nœud	143
VI.4.3 Règles de dominance	145
VI.5 Recherche arborescente basée sur les divergences	147
VI.5.1 Recherche arborescente à divergences limitées	147
VI.5.2 Stratégie d'exploration	148
VI.5.3 Recherche locale à grands voisinages basée sur les divergences	151
VI.5.4 Règles de dominance adaptées aux divergences	153
VI.6 Résultats expérimentaux	155
VI.6.1 Générateur d'instances	155
VI.6.2 Tests et résultats	156
VI.7 Conclusion	163
Conclusion et perspectives	165

Bibliographie

169

Table des figures

I.1	Tournées de véhicules	9
II.1	Analyse du domaine de travail à travers une hiérarchie d'abstraction pour le problème d'ordonnement des transports	42
III.1	Architecture du système d'aide à la décision	50
III.2	Schéma des tâches à réaliser pour la résolution du problème	55
III.3	Différents modes de contrôle proposés par van Wezel et al. (2010)	56
III.4	Programmation par contraintes	59
III.5	Comparaison des voisinages explorés par CDS-V1 et CDS-V2	74
IV.1	Ensemble d'interfaces pour la modélisation des objets physiques du problème	87
IV.2	Interface de support pour la modélisation des contraintes entre les objets physiques	88
IV.3	Interface de support à l'opérateur humain pour la sélection des véhicules	91
IV.4	Interface de support à l'opérateur humain pour l'affectation des clients	93
IV.5	Interfaces de support à l'opérateur humain pour la construction des itinéraires	94
IV.6	Interface dédiée au support à l'opérateur pour la sélection de la solution finale	97
IV.7	Graphe équivalent au problème de transbordement de l'expérience	101
IV.8	Comparaison entre l'interface analytique et l'interface perceptive – Résultats du temps (secondes) nécessaire pour la sélection d'une solution	104
IV.9	Comparaison entre l'interface analytique et l'interface perceptive – Résultats de la charge mentale (NASA-TLX) requise pour la sélection d'une solution pour la	105
IV.10	Comparaison entre l'interface analytique et l'interface perceptive – Résultats de performance (pourcentage de réussite) pour la sélection d'une solution	106

IV.11 Comparaison entre l'interface intégrale et l'interface abrégée – Résultats du temps (secondes) nécessaire pour la sélection d'une solution	107
IV.12 Comparaison entre l'interface intégrale et l'interface abrégée – Résultats de la charge mentale (NASA-TLX) requise pour la sélection d'une solution	108
IV.13 Comparaison entre l'interface intégrale et l'interface abrégée – Résultats de per- formance (pourcentage de réussite) pour la sélection d'une solution	108
V.1 Exemple de différents types de problèmes	112
V.2 Analyse des options proposées dans Higgins (1996) pour surmonter un problème non-réalisable	113
V.3 Localisation géographique des clients à servir	117
V.4 Illustration du fonctionnement de l'algorithme des centres mobiles pour la classi- fication géographique des clients	122
V.5 Principales relations entre deux fenêtres de temps	123
V.6 Distribution normale standard ou distribution z	127
VI.1 Exemple d'un ordonnancement réalisable	138
VI.2 Exemple d'affectation de tâches	139
VI.3 Graphe pour déterminer la réalisabilité d'un ordonnancement	140
VI.4 Energie minimale requise pour un ordonnancement partiel	145
VI.5 Ordonnancement partiel du nœud évalué	147
VI.6 Recherche arborescente basée sur les divergences pour un arbre binaire	148
VI.7 Exemple des modes de comptage de divergences pour la sélection de la tâche à ordonnancer	150
VI.8 Ordre d'exploration des feuilles pour différent schémas de branchement	150
VI.9 Exemple d'ordonnancement dominant	155