

Collection des rapports de recherche
de Télécom Bretagne

RR-2011-03-LUSSI



Synthèse du problème de routage de véhicules

Sahbi Ben Ismail (Télécom Bretagne, Labsticc)

François Legras (Deev Interaction)

Gilles Coppin (Télécom Bretagne, Labsticc)



Table des matières

Introduction	5
1 Les problèmes de transport	5
1.1 Le problème du voyageur de commerce (TSP)	6
1.2 Le problème du postier chinois (CPP)	6
2 Le problème de routage de véhicules (VRP)	7
2.1 Définition	7
2.2 Formulation mathématique	7
2.3 Paramètres	10
2.4 Variantes principales	12
3 Méthodes et techniques de résolution du VRP	13
3.1 Classification générale	14
3.1.1 Méthodes exactes	14
3.1.2 Méthodes approchées	14
3.2 Classification des heuristiques de résolution du VRP	16
3.2.1 Méthodes constructives	17
3.2.2 Méthodes à deux phases	17
3.2.3 Méthodes d'amélioration	20
3.2.4 Hybridation des heuristiques et des métaheuristiques	20
3.3 Algorithmes génétiques pour la résolution de VRP	22
3.3.1 Principe général	22
3.3.2 Application au VRP	25
3.4 Comparaison entre les différentes méthodes de résolution	31
4 Résolution interactive de problèmes de transport	33
4.1 Résolution humaine de VRP/TSP	33
4.1.1 Efficacité de la résolution humaine <i>manuelle</i>	33
4.1.2 Modèles des stratégies humaines de résolution	34
4.2 Taxonomie des niveaux d'implication d'un humain dans le processus de résolution	35

4.3	Approches de résolution interactive de VRP	37
4.3.1	Premières tentatives de résolution interactive	37
4.3.2	Approche <i>Human-Guided Search</i> (HuGS)	38
4.3.3	Approches évolutionnaires interactives	40
Conclusion		41
A Dates clés de l'histoire du VRP		42
B Problèmes références (Benchmarks)		44
B.1	Instances Tests de Solomon pour le DCVRPTW	44
B.2	Autres problèmes références	46

Table des figures

1	Un exemple de problème de VRP à $n = 20$ clients résolu avec $m = 4$ véhicules.	8
2	Variantes de base du VRP avec contraintes de capacité.	12
3	Classification des méthodes de résolution du VRP.	15
4	Illustration du principe de la méthode des économies (<i>savings</i>). . .	18
5	Illustration de la première phase de l'algorithme de balayage (<i>sweep</i>).	19
6	Opérateurs heuristiques d'amélioration utilisés par Tam and Ma [2004] pour résoudre un CVRPTW.	21
7	Classification des familles de méthodes de résolution du VRP selon Hao et al. [1999].	22
8	Exemple d'opérateur de mutation basique.	24
9	Exemple de deux opérateurs de croisement basiques.	24
10	Deux solutions utilisées comme parents pour illustrer l'opérateur de croisement de Baker and Ayechew [2003].	27
11	Enfant E2 généré par le croisement des parents de la figure 10 [Baker and Ayechew, 2003].	28
12	Résolution humaine de TSP : comment fait un humain pour résoudre un TSP ? (tirée de <i>The Journal of Problem Solving</i>). . . .	34
13	Exemple de résolution humaine manuelle d'un CVRP.	36
14	Niveaux d'implication d'un humain dans Wezel et al. [2011]. . . .	37
15	Approche de Krolak et al. [1970] pour la résolution interactive d'un TSP.	39
16	Dates clés du VRP : publications de référence et méthodes de résolution.	43
17	Visualisation de trois instances (c101, r101 et rc101) des problèmes de Solomon.	45

Introduction

Le problème de routage de véhicules (*Vehicle Routing Problem*, VRP dans la suite de ce document) est un problème d’optimisation combinatoire et de recherche opérationnelle. Il fait partie de la catégorie des problèmes de transport, tout comme le problème du voyageur de commerce (*Traveling Salesman Problem*, TSP) et le problème du postier chinois (*Chinese Postman Problem*, CPP).

Dans ces problèmes relevant du domaine de la logistique, un ou plusieurs véhicules doivent couvrir un réseau de transport pour livrer des marchandises à des clients ou couvrir les routes de ce réseau.

Le problème de routage de véhicules a été largement étudié durant la deuxième moitié du siècle dernier. Rego and Roucairol [1994] considèrent que les années 1980s constituent l’apogée de ces études et « l’un des plus beaux succès de la recherche opérationnelle ».

Les principales motivations de l’étude du VRP sont d’une part la difficulté de sa résolution et d’autre part ses nombreuses applications pratiques en logistique. Ce deuxième point concerne les retombées économiques et environnementales liées à la minimisation des coûts des systèmes de transport. En effet, selon Toth and Vigo [2001a], les frais de transport représentent généralement entre 10% et 20% des prix finaux des marchandises sur le marché, et les procédures informatisées basées sur des techniques d’optimisation permettent de faire des économies de l’ordre de 5% à 20% sur ces coûts de transport.

Nous commencerons tout d’abord par présenter les problèmes de TSP, VRP et CPP en mettant l’accent sur la modélisation mathématique du problème de routage de véhicules ainsi que sur ses variantes principales.

Nous présenterons ensuite une synthèse des différentes méthodes de résolution du VRP et de leurs classifications. Nous apporterons une attention particulière aux algorithmes génétiques et à leur application au VRP.

Nous nous intéresserons enfin aux approches interactives dans la résolution du problème de routage de véhicules et aux performances humaines dans ce domaine.

1 Les problèmes de transport

Les problèmes de transport, appelés aussi problèmes de routage, modélisent des problèmes réels liés au transport de marchandises ou de personnes. Afin d’introduire le problème de routage de véhicules, nous parlerons de deux autres problèmes de transport : le problème du voyageur du commerce et le problème du postier chinois.

1.1 Le problème du voyageur de commerce (TSP)

Selon Laporte and Osman [1995] et Dhaenens et al. [2002], le problème du voyageur de commerce serait le problème le plus célèbre et le plus étudié en optimisation combinatoire. Dans ce problème, un voyageur de commerce doit visiter plusieurs villes (ou clients) en passant une et une seule fois par chacune d'entre elles, et en minimisant la distance totale parcourue [Lawler et al., 1985].

Plus formellement, un TSP est modélisé sous forme d'un graphe où les sommets représentent les villes à visiter, et les arêtes les liaisons entre ces villes [Dhaenens et al., 2002]. La pondération ou le poids associé à chaque arête représente le coût de la liaison entre les deux villes et correspond généralement à la distance qui les sépare. L'objectif est de trouver un *cycle hamiltonien*, *c-à-d* un cycle passant une et une seule fois par tous les sommets du graphe, et de *longueur minimale*.

En tant que problème d'optimisation, le TSP est un problème *NP-difficile*. En effet, dans sa version symétrique, *c-à-d* dans le cas où le graphe associé n'est pas orienté, le nombre total de solutions possibles est $\frac{(n-1)!}{2}$ où n est le nombre de villes. Avec une telle complexité factorielle, une résolution efficace du TSP nécessite donc le recours à des heuristiques spécialisées voire même à des méta-heuristiques. En effet, les méthodes exactes restent limitées aux problèmes de petite taille.

Nous reviendrons sur ces différentes méthodes de résolution avec plus de détails dans la section 3.

1.2 Le problème du postier chinois (CPP)

Le problème du postier chinois fait partie aussi de la classe des problèmes de transport : un réseau de villes est représenté sous forme d'un graphe où les sommets et les arcs correspondent respectivement aux villes et aux liaisons. Mais contrairement au voyageur du commerce, le postier chinois doit effectuer un tour passant par toutes les arêtes au moins une fois, tout en minimisant la distance totale parcourue. Ce problème peut modéliser les tournées de ramassage d'ordures ménagères ou de distribution de courrier [Dhaenens et al., 2002].

Nous retiendrons que les problèmes de transport se modélisent par un graphe orienté ou non $G = (V, E)$ où $V = \{v_1, \dots, v_n\}$ est un ensemble de *sommets* modélisant les villes (ou les clients) et $E = \{(v_i, v_j) : i \leq j; v_i, v_j \in V\}$ est un ensemble d'*arêtes* ou d'arcs reliant les villes. Les arcs et les arêtes correspondent respectivement aux problèmes symétriques (ou non orientés) et aux problèmes asymétriques (ou orientés).

Pour plus de détails sur le TSP et le CPP, on pourra se référer à la bibliographie dressée par Laporte and Osman [1995]. Nous retiendrons que la différence principale entre le TSP et le CPP réside dans l'élément du graphe sur lequel

opère l'optimisation : le TSP impose que chaque sommet soit visité et le CPP impose que chaque arête (ou arc) soit visitée.

Plus généralement, les problèmes de routage peuvent être classés en deux catégories : les problèmes qui opèrent sur les villes (*vertex routing problems*) et les problèmes qui opèrent sur les liaisons (*arc routing problems*).

2 Le problème de routage de véhicules (VRP)

2.1 Définition

Le problème de routage de véhicules est une extension du problème du voyageur du commerce [Dhaenens et al., 2002]. Il a été introduit pour la première fois par Dantzig et al. [1954] sous le nom de « *Truck Dispatching Problem* » et a depuis fait l'objet d'études intensives pour le modéliser et le résoudre.

Dans sa version la plus basique dite *Capacitated VRP* (CVRP) ou VRP avec contraintes de capacité, une *flotte de véhicules* de *capacité finie*, basée dans un *dépôt*, doit assurer des tournées entre plusieurs *clients* (ou villes) ayant demandé chacun une certaine *quantité* de marchandises. L'ensemble des clients visités par un véhicule désigne la *tourné*e de celui-ci. Chaque client doit être desservi une et une seule fois et chaque tournée commence et se termine au dépôt.

L'objectif du CVRP est de minimiser le coût total, *c-à-d* la somme des distances ou des temps de parcours des tournées, tout en respectant la contrainte de capacité des véhicules : la quantité de marchandises livrées sur une tournée ne doit pas dépasser la capacité du véhicule qui l'assure.

La figure 1¹ représente un exemple de problème de VRP avec 20 clients, résolu avec 4 véhicules.

2.2 Formulation mathématique

La formulation du VRP que nous présentons ici correspond à la formulation mathématique utilisée en programmation linéaire en nombres entiers. Elle traduit la modélisation naturelle du problème par la définition d'une variable binaire x_{ijk} égale à 1 si le véhicule k parcourt l'arc (v_i, v_j) , noté plus simplement (i, j) . Cette formulation est la plus utilisée dans la littérature, et a ainsi été adoptée par Laporte [1992], Rego and Roucairol [1994], Toth and Vigo [2001a], Crainic and Semet [2006], etc.

Nous reprenons les mêmes notations définies dans la section 1 en ajoutant à l'ensemble des sommets du graphe $V = \{v_1, \dots, v_n\}$ un autre sommet v_0 , correspondant au dépôt. Comme c'est le cas généralement, nous supposons que le

1. exemple tiré de [Baker and Ayechew, 2003].

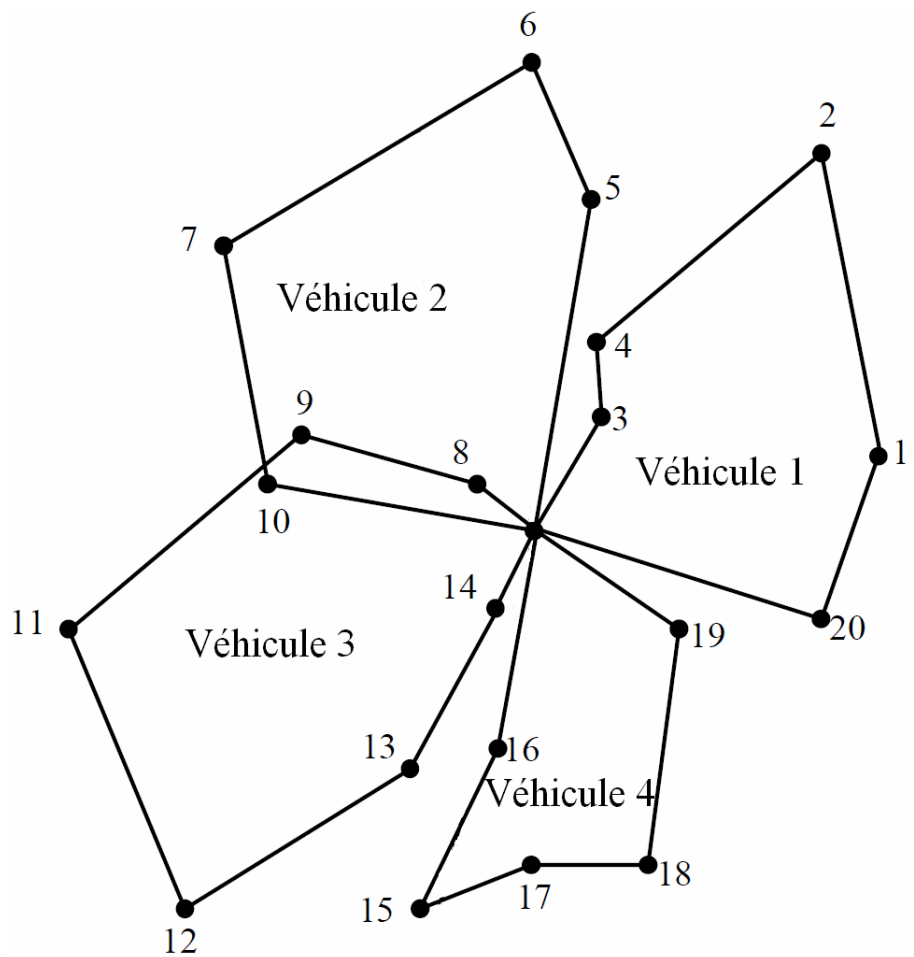


FIGURE 1 – Un exemple de problème de VRP à $n = 20$ clients résolu avec $m = 4$ véhicules.

graphe $G = (V, E)$ est complet², c-à-d que tous les sommets sont reliés entre eux. Cela signifie qu'une ville peut être visitée à partir de toute autre ville.

Les autres constantes du problème sont les suivantes :

n nombre de clients (ou sommets)

m nombre de véhicules

Q capacité des véhicules

q_i demande du client i

c_{ij} le coût de l'arête entre les sommets i et j (distance ou temps de parcours)

Les variables de décision du problème sont les x_{ijk} évoquées plus haut :

$$x_{ijk} = \begin{cases} 1 & \text{si } (i, j) \text{ est parcouru par le véhicule } k, \\ 0 & \text{sinon.} \end{cases}$$

Ainsi, en tant que problème d'optimisation, le CVRP s'écrit :

$$\text{Minimiser } \sum_{i=1}^n \sum_{j=1}^n c_{ij} \sum_{k=1}^m x_{ijk} \quad (1)$$

sujet aux contraintes suivantes :

$$\sum_{i=1}^n \sum_{k=1}^m x_{ijk} = 1 \quad \forall 1 \leq j \leq n \quad (2)$$

$$\sum_{j=1}^n \sum_{k=1}^m x_{ijk} = 1 \quad \forall 1 \leq i \leq n \quad (3)$$

$$\sum_{i=1}^n \sum_{l=1}^n x_{ilk} = \sum_{l=1}^n \sum_{j=1}^n x_{ljk} \quad (4)$$

$$\sum_{j=1}^n x_{0jk} = 1 \quad \forall 1 \leq k \leq m \quad (5)$$

$$\sum_{i=1}^n x_{i0k} = 1 \quad \forall 1 \leq k \leq m \quad (6)$$

$$\sum_{i=1}^n \sum_{j=1}^n x_{ijk} \leq Q \quad \forall 1 \leq k \leq m \quad (7)$$

2. Un graphe complet à n sommets possède $\frac{n(n-1)}{2}$ arêtes.

$$x_{ijk} \in \{0, 1\} \quad \forall 0 \leq i, j \leq n; 1 \leq k \leq m \quad (8)$$

Sous cette formulation, (1) signifie que l'objectif du problème d'optimisation est de minimiser la somme des coûts de toutes les tournées.

Les contraintes (2) et (3) imposent que chaque client soit desservi une et une seule fois et les contraintes (4) assurent la conservation de flot. Les contraintes (5) assurent que chaque tournée commence et se termine au dépôt. Finalement, les contraintes (6) sont les contraintes de capacité et les contraintes (7) sont des contraintes de binarité sur les variables de décision x_{ijk} .

Remarque : Notons que cette formulation mathématique du problème de routage de véhicules avec contraintes de capacité repose sur un modèle appelé *modèle de flot à trois indices (three-index vehicle flow formulation)* [Rego and Roucairol, 1994], [Toth and Vigo, 2001b], [Crainic and Semet, 2006]. Étant donné que dans (1) le coût de la solution est exprimé comme la somme des coûts des arêtes traversées, un modèle de flot à deux indices seulement (x_{ij}) peut également être utilisé de manière aussi efficace. Cependant, l'absence de l'information « quel véhicule k parcourt l'arête (i, j) ? » devient limitante pour d'autres variantes du VRP, notamment si le coût de la solution dépend du type des véhicules utilisés par exemple [Toth and Vigo, 2001b].

2.3 Paramètres

Outre la version basique du *Capacitated VRP*, présentée dans 2.1, le problème de routage de véhicules a plusieurs autres variantes plus ou moins étudiées dans la littérature.

En effet, la définition la plus générale du VRP est la suivante : il s'agit de la conception de *routes optimales* par une *flotte de véhicules*, basée en un ou plusieurs dépôts, pour desservir un *ensemble de clients (ou villes)* dispersés géographiquement et ayant des demandes connues.

Cette définition généraliste met en évidence l'ensemble de paramètres qui caractérisent une variante du VRP : le *réseau de transport*, la *clientèle* et la *flotte de véhicules*. Des *contraintes auxiliaires* peuvent éventuellement s'ajouter à ces trois paramètres principaux. Un dernier paramètre réside dans la *fonction objectif* à optimiser.

Le réseau Le réseau routier peut être *symétrique* ou *asymétrique* ; en conséquence, le graphe associé $G = (V, E)$ sera orienté ou non et les liaisons entre les sommets seront des arcs ou des arêtes.

Les tournées peuvent partir d'un seul dépôt ou de *plusieurs dépôts*.

La clientèle La principale caractéristique de la clientèle est sa *demande* en marchandise.

La livraison de celle-ci peut être contrainte à s'effectuer au cours de périodes de temps spécifiées, appelées *fenêtres temporelles* (*time windows*). Ces contraintes peuvent être dures ou souples. Dans le cas de contraintes dures, une arrivée avant la fenêtre temporelle impose une attente, et les retards sont interdits ; alors que les contraintes souples peuvent être violées et induisent ainsi des pénalités [Kallehauge et al., 2005].

Les clients peuvent être livrés en marchandises mais peuvent aussi en remettre aux véhicules. On parle alors de tournées de livraison/ramassage ou de service mixte. Les temps de livraison et de ramassage peuvent être non négligeables et sont ainsi pris en compte dans le calcul des durées de tournées.

Finalement, l'accès à un client peut être limité à un sous-ensemble de véhicules seulement.

La flotte de véhicules Le nombre de véhicules disponibles peut être fixe ou non. Notons que dans le cas d'un seul véhicule, le problème de tournées reste tout de même différent d'un problème de voyageur de commerce car la livraison des différents clients peut s'effectuer en plusieurs tournées.

Un véhicule peut être associé à un dépôt particulier ou pas.

Les véhicules peuvent avoir une capacité maximale en termes de marchandises transportées (volume, poids etc.). Dans le cas d'une flotte hétérogène, cette capacité peut différer selon le type de véhicules.

La fonction objectif Les objectifs les plus communs sont soit la minimisation du nombre de véhicules utilisés soit la minimisation de la distance totale parcourue par les véhicules. D'autres objectifs peuvent être considérés :

- la minimisation de la durée totale des tournées
- la minimisation du coût total des tournées (en prenant en compte les coûts des véhicules, des chauffeurs etc.)
- la minimisation des pénalités liées aux violations des contraintes, notamment dans le cas de fenêtres temporelles
- la maximisation des gains engendrés par les tournées
- etc.

Selon Bräysy [2001], les objectifs de minimisation du nombre de véhicules et de la distance (ou durée) totale des tournées sont conflictuels : la diminution du nombre de véhicules engendre le plus souvent une augmentation de la distance totale parcourue.

Notons que dans des approches multi-objectifs, une somme pondérée de ces objectifs peut être considérée [Jozefowicz, 2004]. Un exemple de cette pondération est donné dans la sous-section 3.3.2.

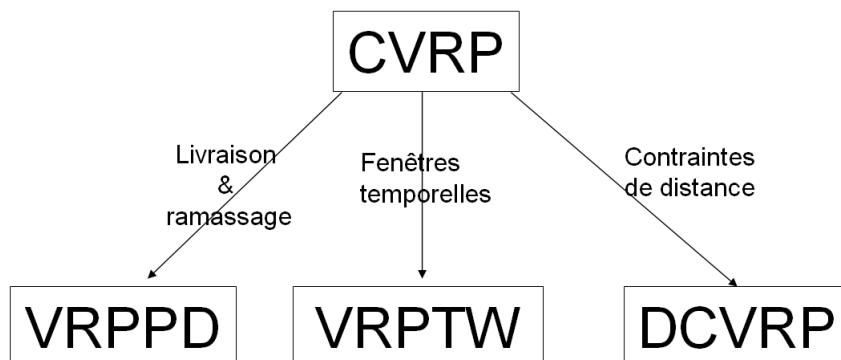


FIGURE 2 – Variantes de base du VRP avec contraintes de capacité.

Dans un état de l’art des différentes approches multi-objectifs, Talbi [2001] classe cette méthode d’*agrégation* parmi les approches transformant le problème multi-objectif en un problème mono-objectif. Il distingue aussi deux autres types d’approches : les approches *non-Pareto* –qui traitent les différents objectifs séparément– et les approches *Pareto* qui cherchent des solutions diversifiées sur la frontière de Pareto. Cette dernière approche est basée sur la notion de *Pareto-dominance* entre les solutions : une solution est dite Pareto-optimale (ou optimale au sens de Pareto) si elle est optimale sur au moins un objectif, et que toute amélioration d’un objectif entraînerait la dégradation d’un autre.

2.4 Variantes principales

Nous nous intéressons ici aux trois variantes principales du VRP basique :

- le VRP avec contraintes de capacité (CVRP pour *Capacitated VRP*) défini dans 2.1 et formulé dans 2.2,
- le VRP avec fenêtres temporelles (VRPTW pour *VRP with Time Windows*),
- et le VRP avec livraison/ramassage (VRPPD pour *VRP with Pick-up and Deliveries*).

Ces variantes ne sont pas indépendantes les unes des autres puisque le VRPTW et VRPPD constituent deux extensions du CVRP. Cette dépendance est illustrée dans la figure 2. Nous y avons fait figurer une autre variante du VRP, le CVRRP avec contraintes de distance (DCVRP pour *Distance-Constrained VRP*) où une contrainte de distance maximale parcourue est imposée aux véhicules.

Dans le VRPTW, chaque client i dispose d’une fenêtre temporelle $[a_i, b_i]$ durant laquelle il peut être livré. En poursuivant les notations de la formulation mathématique de 2.2, d’autres variables de décision sont définies : pour $0 \leq i \leq n$

et $1 \leq k \leq m$, s_{ik} désigne l'instant où le véhicule k commence à servir le client i . Il en résulte deux ensembles de contraintes supplémentaires :

$$x_{ijk}(s_{ik} + t_{ij} - s_{jk}) \leq 0 \quad (9)$$

et

$$a_i \leq s_{ik} \leq b_i \quad (10)$$

Les contraintes (9) permettent de prendre en compte la durée de trajet entre deux clients consécutifs i et j (t_{ij}) et les contraintes (10) correspondent tout simplement à la définition des fenêtres temporelles des clients.

Comme son nom l'indique, dans le VRPPD les véhicules effectuent un double service : la livraison (*Delivery*) des clients ainsi que le ramassage (*Pick-up*) de marchandises de ces derniers. Le ramassage induit une difficulté supplémentaire en rapport avec la contrainte de capacité des véhicules. Les quantités de marchandises à ramasser sont notées p_i , $1 \leq i \leq n$ et par convention, lorsqu'un véhicule dessert un client il commence par la livraison puis effectue le ramassage. Le dépôt est par défaut le lieu de départ des marchandises à livrer et le lieu des marchandises ramassées, sinon, il faut préciser pour chaque client v_i , $1 \leq i \leq n$ le client O_i , l'origine de la marchandise demandée, et le client D_i , la destination des marchandises ramassées. Il faut aussi que O_i et D_i soient desservis sur la même tournée que le client v_i , respectivement avant et après v_i , c-à-d dans l'ordre $\rightarrow O_i \rightarrow \dots \rightarrow v_i \rightarrow \dots \rightarrow D_i \rightarrow$.

Au delà de ces trois problèmes importants, le VRP a plusieurs autres variantes qui résultent de la combinaison de plusieurs contraintes. Par exemple, le problème prenant en compte les contraintes de capacité, les contraintes de fenêtres temporelles et le double service livraison/ramassage est appelé VRPPDTW pour *VRP with Pick-up and Deliveries and Time Windows*.

Pour une revue détaillée de ces variantes, on pourra se référer à Bodin and Golden [1981] [Rego and Roucairol, 1994]. Outre les ouvrages et les publications, plusieurs sites web ont été dédiés à l'étude du VRP. Nous en citons par exemple *The VRP Web*³ où sont décrites plusieurs variantes du VRP (dépôts multiples (*Multiple Depot VRP*), livraisons périodiques (*Periodic VRP*), livraison d'un client par plusieurs véhicules (*Split Delivery VRP*), livraisons stochastiques (*Stochastic VRP*), etc.

3 Méthodes et techniques de résolution du VRP

Dans cette section nous proposons une vue d'ensemble sur les méthodes de résolution du problème de routage de véhicules. Notre but n'est pas de détailler

3. <http://neo.lcc.uma.es/radi-aeb/WebVRP/index.html>

www.telecom-bretagne.eu

Campus de Brest

Technopôle Brest-Iroise

CS 83818

29238 Brest Cedex 3

France

Tél. : + 33 (0)2 29 00 11 11

Fax : + 33 (0)2 29 00 10 00

Campus de Rennes

2, rue de la Châtaigneraie

CS 17607

35576 Cesson Sévigné Cedex

France

Tél. : + 33 (0)2 99 12 70 00

Fax : + 33 (0)2 99 12 70 19

Campus de Toulouse

10, avenue Edouard Belin

BP 44004

31028 Toulouse Cedex 04

France

Tél. : +33 (0)5 61 33 83 65

Fax : +33 (0)5 61 33 83 75

© Télécom Bretagne, 2011

Imprimé à Télécom Bretagne

Dépôt légal : janvier 2011

ISSN : 1255-2275



le fonctionnement de ces méthodes, mais plutôt d’avoir un aperçu sur leur principe de base et leurs classifications afin d’identifier les méthodes adaptées à une résolution interactive.

3.1 Classification générale

Comme les autres problèmes d’optimisation combinatoire, le problème de routage de véhicules a été étudié et résolu par des méthodes exactes, des heuristiques spécifiques ainsi que par des métaheuristiques. Ces trois familles correspondent à la classification générale des méthodes de résolution ([Hao et al., 1999], [Gendreau and Potvin, 2005]) et sont illustrées dans la figure 3.

3.1.1 Méthodes exactes

Les méthodes exactes, appelées aussi méthodes *complètes*, permettent de trouver la solution optimale d’un problème d’optimisation en explorant exhaustivement l’ensemble des solutions (ou configurations) possibles [Solnon, 2010]. L’exploration énumérative (toutes les solutions sont évaluées une à une) est la technique la plus basique mais elle reste inappropriée aux problèmes combinatoires. Voilà pourquoi des méthodes exactes comme l’algorithme de « Séparation & Évaluation » (*Branch & Bound*⁴) explorent l’ensemble des solutions (ou configurations) possibles et éliminent des sous-ensembles de mauvaises solutions à l’aide de techniques d’élagage. L’avantage d’une telle approche est que les solutions éliminées ne sont pas évaluées « à la main » mais de façon globale.

3.1.2 Méthodes approchées

Contrairement aux méthodes exactes, les méthodes approchées sont *incomplètes* : elles permettent de trouver des bonnes solutions mais ne garantissent en aucun cas l’optimalité de celles-ci⁵.

Les méthodes approchées sont composées de heuristiques et de métaheuristiques.

Les heuristiques Par définition, une heuristique⁶ est un moyen de guider le choix que doit faire un algorithme pour réduire sa complexité. Une heuristique est spécifique à un problème et ne peut pas être généralisée.

4. L’algorithme *Branch & Bound* est une généralisation de l’algorithme *backtrack* utilisé pour la résolution des problèmes de satisfaction de contraintes [Bensana and Verfaillie., 1995].

5. Notons que par abus de langage, nous confondons ici les deux notions de *complétude* d’une méthode d’optimisation (sa faculté à trouver une solution s’il y en a) et de son *optimalité* (sa faculté à trouver la (ou les) meilleure(s) solution(s)) [Russell et al., 1995]

6. Étymologiquement, le mot heuristique signifie « trouver, découvrir » en grec ancien.

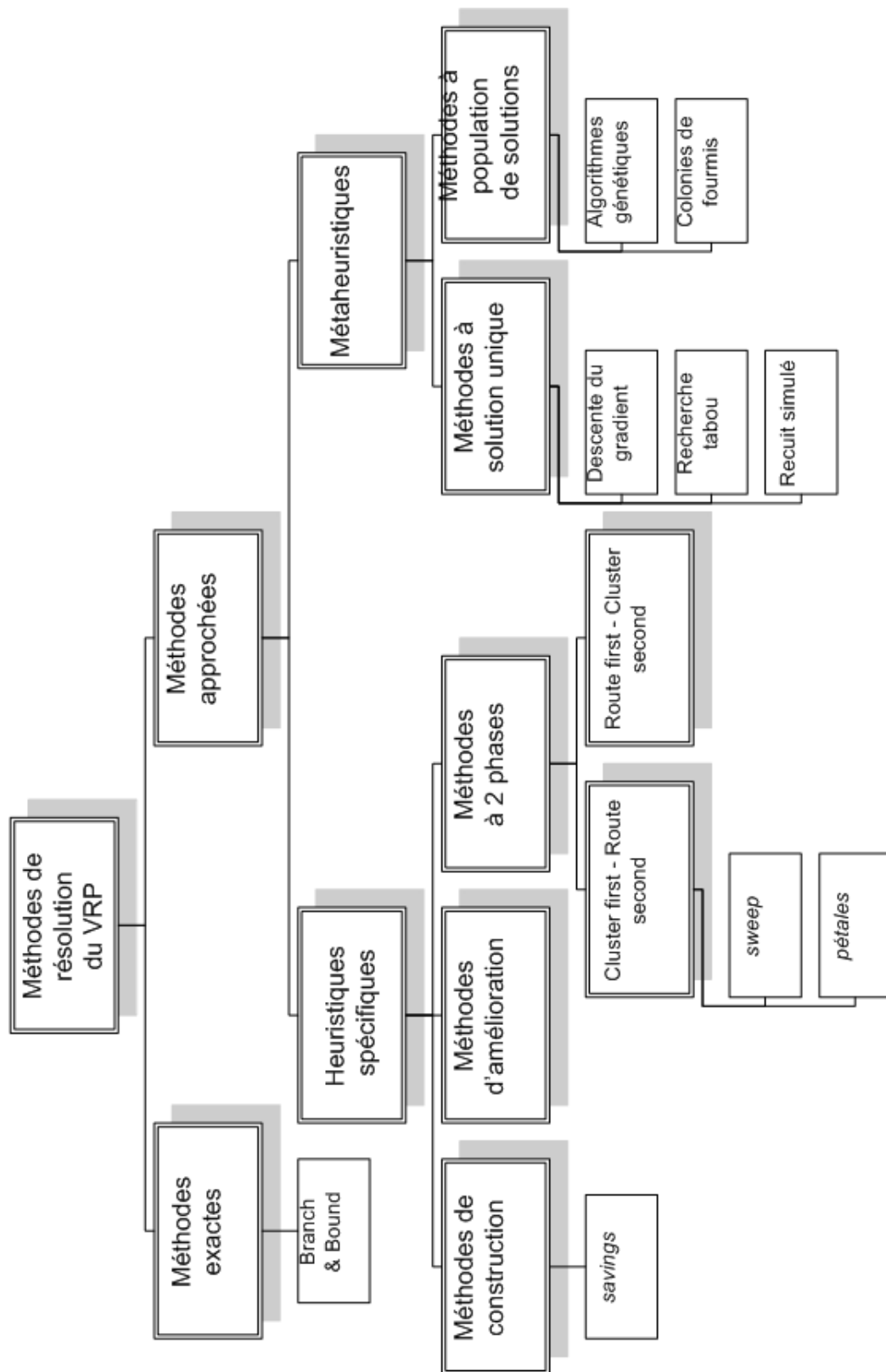


FIGURE 3 – Classification des méthodes de résolution du VRP.

Les métaheuristiques Les métaheuristiques peuvent être vues comme des heuristiques « puissantes et évoluées » dans la mesure où elles sont généralisables à plusieurs problèmes d’optimisation. Les métaheuristiques sont habituellement classées en fonction du nombre de solutions qu’elles manipulent : les métaheuristiques à solution unique telles que la recherche tabou et le recuit simulé ; et les métaheuristiques à population de solutions telles que les algorithmes génétiques et les colonies de fourmis [Talbi, 2009].

Dilemme intensification versus diversification Notons que la mise au point des méthodes approchées doit trouver un équilibre entre deux tendances opposées lors de la recherche dans l’espace des solutions : l’*intensification* (ou *exploitation*) et la *diversification* (ou *exploration*) [Solnon, 2010]. L’intensification de la recherche signifie que celle-ci se concentre autour des meilleures solutions rencontrées, considérées comme prometteuses ; alors que la diversification incite la recherche à explorer des nouvelles zones de l’espace de recherche en vue d’y trouver des bonnes solutions.

Les métaheuristiques à solution unique ont plus tendance à l’exploitation du voisinage de la solution en question, et les approches à base de population de solutions ont plutôt tendance à l’exploration [Hao et al., 1999].

Dans la sous-section suivante 3.2 nous présentons les heuristiques principales dédiées au problème de routage de véhicules.

La sous-section 3.3, elle, sera consacrée aux grandes lignes de l’application des algorithmes génétiques à la résolution du VRP. Pour une revue détaillée de l’application des métaheuristiques au VRP, on pourra se référer à Bräysy and Gendreau [2005b], Gendreau et al. [2003], Bräysy and Gendreau [2001a], Laporte [1992], Gendreau et al. [2001] et Gendreau et al. [2008].

3.2 Classification des heuristiques de résolution du VRP

Les heuristiques de résolution du problème du voyageur du commerce (TSP), introduit dans 1.1, sont essentiellement classées en deux catégories : les *procédures de construction de tours* et les *procédures d’amélioration de tours* [Gendreau et al., 1992] [Yeun et al., 2008]. Dans la première classe, une solution est construite progressivement en ajoutant un sommet (ville) à chaque étape. Dans la deuxième classe plusieurs opérateurs d’échange de sommets sont appliqués à une solution en vue de l’améliorer.

Le problème de routage de véhicules étant une extension du TSP, les deux approches de construction et d’amélioration trouvent naturellement leur place dans les heuristiques qui lui sont dédiées. Une troisième classe de méthodes dites « à deux phases » vient s’ajouter à ces deux classes et reflète la décomposition du VRP en deux sous-problèmes : l’affectation des clients aux véhicules et l’optimisation des tournées de ces derniers.

Cette classification des heuristiques du VRP en trois catégories est la plus répandue dans la littérature ([Dhaenens et al., 2002], [Crainic and Semet, 2006], [Gendreau et al., 2003] et [Rego and Roucairol, 1994]), nous donnerons ensuite un aperçu de chacune d’entre elles.

3.2.1 Méthodes constructives

Les méthodes constructives sont des méthodes itératives où, à chaque itération, une solution partielle est complétée [Dhaenens et al., 2002].

L’heuristique constructive la plus connue est la méthode des économies (*savings*) de Clarke and Wright [1964] dont voici le principe : elle part d’une solution initiale où chaque client est desservi par un véhicule, et tente de fusionner progressivement les tournées. Concrètement, la méthode calcule pour chaque paire de sommets $(v_i, v_j)_{1 \leq i, j \leq n}$ l’économie e_{ij} réalisée en allant directement de v_i à v_j plutôt que de passer par le dépôt.

La figure 4 illustre un exemple basique où 0 est le dépôt et i et j sont deux clients. En retirant les liaisons $(i, 0)$ et $(0, j)$ et en ajoutant la liaison (i, j) le coût passe de $c_{0i} + c_{i0} + c_{0j} + c_{j0}$ à $c_{0i} + c_{ij} + c_{j0}$. Ainsi l’économie (*saving*) engendrée par cette liaison directe est $e_{ij} = c_{i0} + c_{0j} - c_{ij}$ [Rego and Roucairol, 1994]. Notant que cette économie n’a pas de sens si la liaison directe entre les clients i et j induit une violation d’une des contraintes du problème, telle que la contrainte de capacité des véhicules dans le CVRP ou la contrainte de distance dans le DCVRP.

Les économies calculées sont ensuite ordonnées, et à chaque itération, les clients correspondants à la plus grande économie sont regroupés sur une même tournée. Cette version de la méthode des *savings* est qualifiée de *parallèle* car elle permet d’étendre toutes les tournées en même temps. Il existe aussi une autre version dite *séquentielle* où une seule tournée est étendue [Crainic and Semet, 2006].

Dhaenens et al. [2002] proposent une classification intéressante des *schémas de construction* de solutions, et l’illustrent par des exemples de heuristiques pour le TSP. Nous avons résumé cette classification dans le tableau 1.

En général, les méthodes constructives sont des algorithmes gloutons (*c-à-d* qu’elles ne remettent pas en question les choix effectués lors des itérations précédentes) et ne sont pas efficaces sur les instances tests de la littérature [Crainic and Semet, 2006]. Elles permettent cependant de construire rapidement de bonnes solutions, ce qui justifie leur utilisation en amont des méthodes d’amélioration et leur hybridation avec les métaheuristiques.

3.2.2 Méthodes à deux phases

Les méthodes à deux phases sont basées sur la décomposition du VRP en deux sous-problèmes : le sous-problème de partitionnement des clients en sous-groupes

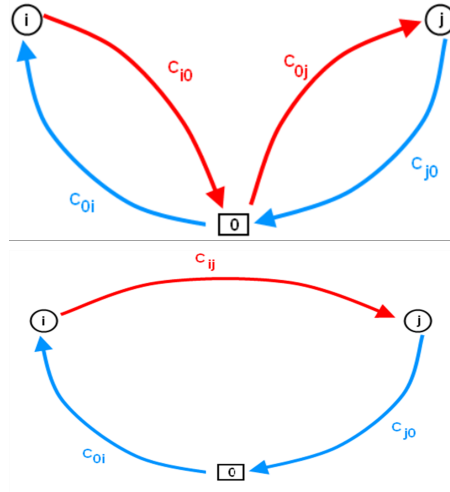


FIGURE 4 – Illustration du principe de la méthode des économies (*savings*).

Schéma de construction	Principe	Exemple de heuristique pour le TSP
algorithmes de liste	<ol style="list-style-type: none"> 1. une liste de priorités est calculée 2. construction de la solution en fonction de la liste 	construction par abscisse croissante (les villes sont ordonnées par abscisses croissantes puis sont reliées dans cet ordre)
règles de priorité	à chaque itération de la construction, choisir un élément en fonction d'une règle de priorité	heuristique du plus proche voisin (à chaque itération, la ville voisine la plus proche est visitée)
méthodes de décomposition	<ol style="list-style-type: none"> 1. décomposition du problème en sous-problèmes plus petits 2. résolution des sous-problèmes 3. assemblage des solutions trouvées pour obtenir une solution au problème global 	partitionnement géographique (les villes sont regroupées en sous-ensembles constituant des sous-problèmes)

TABLE 1 – Schémas de construction de solutions [Dhaenens et al., 2002].

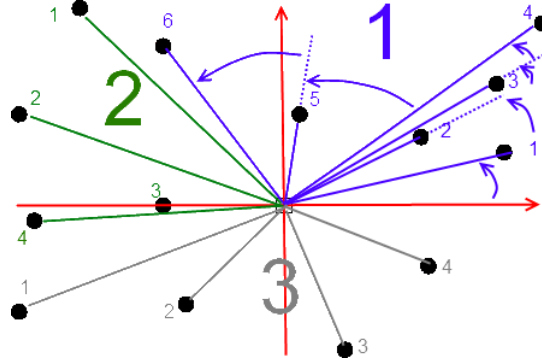


FIGURE 5 – Illustration de la première phase de l'algorithme de balayage (*sweep*).

Le dépôt est l'origine du repère, et le premier groupe créé se compose des 6 clients en bleu ($\{1-2-3-4-5-6\}$). Deux autres groupes ont été formé ensuite. Le passage d'un groupe à un autre est dû à une violation d'une contrainte du VRP.

et le sous-problème de détermination des routes pour chaque sous-groupe. Elles se décomposent naturellement en deux classes suivant l'ordre dans lequel les deux phases (ou sous-problèmes) sont traitées.

Cluster first - Route second Les deux algorithmes les plus connus de cette méthode *Partition d'abord - Parcours ensuite* sont l'algorithme de balayage (*sweep*) et l'algorithme de pétales [Crainic and Semet, 2006].

L'algorithme *sweep* commence d'abord par attribuer aux clients des coordonnées angulaires relatives au dépôt, puis effectue un balayage dans le sens des coordonnées croissantes. Les clients sont ainsi ajoutés successivement, et dès qu'une des contraintes du problème est violée, un nouveau groupe est créé. La figure 5 illustre le fonctionnement de la première phase de l'algorithme de balayage sur l'exemple de VRP de la figure 1. Une fois les groupes formés, l'algorithme procède à la résolution d'un TSP sur chacun d'entre eux.

Route first - Cluster second Ces méthodes de type *Parcours d'abord - Partition ensuite* commencent par résoudre un TSP global avec tous les clients. La tournée trouvée est ensuite divisée en plusieurs tournées respectant les contraintes du VRP.

Notons que les méthodes de type *Cluster first - Route second* sont en général plus efficaces que celles de type *Route first - Cluster second* [Gendreau et al., 2003].

3.2.3 Méthodes d'amélioration

Les méthodes d'amélioration tentent d'améliorer une solution du VRP en lui appliquant des opérateurs *heuristiques d'échange* de clients au sein des tournées. Ces échanges peuvent opérer sur des clients d'une même tournée ou sur les clients de plusieurs tournées différentes. Ces deux modes expliquent la classification des méthodes d'amélioration en deux classes selon le type d'amélioration des tournées (individuellement ou collectivement) [Crainic and Semet, 2006].

Dans la première classe, les tournées sont améliorées individuellement en utilisant des méthodes d'amélioration conçues pour le TSP. Les algorithmes d'échange de type *r-opt*, proposés initialement par Lin [1965], sont très employés et généralisables à tout problème dont une solution consiste en une permutation des éléments de celui-ci. Le principe de ces algorithmes est de considérer r éléments (les clients dans le cas du TSP) et de les échanger entre eux en vue d'améliorer la solution. Le terme *r-opt* (opt pour *optimal*) indique qu'à la fin de la procédure, aucune autre amélioration n'est possible en échangeant r éléments [Dhaenens et al., 2002]. Pour limiter le temps de calcul, la valeur de r est généralement limitée à 2 ou 3.

Dans la deuxième classe, l'amélioration de la solution du VRP s'effectue en opérant sur plusieurs tournées simultanément. Breedmam [1996] a proposé trois types d'opérateurs pour le CVRP et le DCVRP. Le premier, nommé *relocalisation de chaîne*, consiste à déplacer au plus k clients consécutifs d'une tournée vers une autre. Le deuxième, appelé *échange de chaînes*, consiste à échanger k clients consécutifs d'une tournée avec l clients consécutifs d'une autre tournée. Enfin, le troisième type, nommé *échange mixte de chaînes*, combine les deux types précédents en choisissant l'opération qui améliore le plus la solution courante [Gendreau et al., 2003] [Crainic and Semet, 2006].

3.2.4 Hybridation des heuristiques et des métaheuristiques

Il est important de souligner que les heuristiques –de type amélioration notamment– sont très utilisées dans les métaheuristiques. À titre d'exemple, la figure 6 illustre trois opérateurs utilisés par Tam and Ma [2004] pour améliorer les solutions d'un CVRPTW dans une recherche tabou et un recuit simulé. Le premier opérateur (*2-opt*) améliore une seule tournée alors que les deux autres améliorent deux tournées à la fois et sont respectivement de type *relocalisation de chaîne* et *échange de chaînes*⁷ vus dans la sous-section précédente 3.2.3.

Soulignons aussi que plusieurs métaheuristiques à base de solution unique fonctionnent aussi selon le principe d'améliorations successives, notamment les méthodes de recherche locale (la descente du gradient typiquement), la recherche

7. Les auteurs utilisent la taxonomie *intra-route* et *inter-routes* pour distinguer ces deux types d'opérateurs d'amélioration.

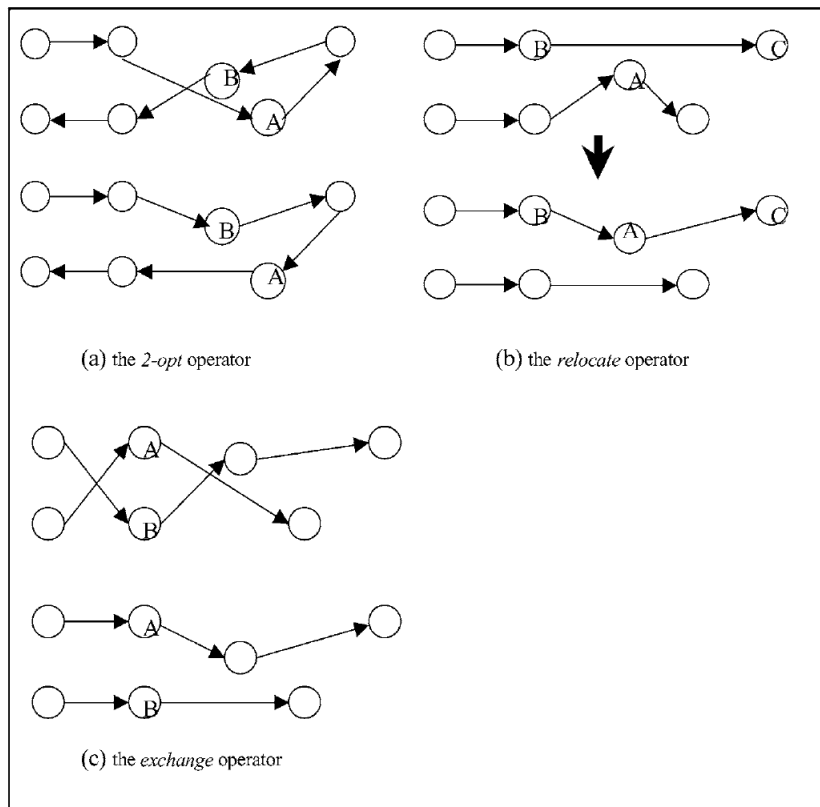


FIGURE 6 – Opérateurs heuristiques d'amélioration utilisés par Tam and Ma [2004] pour résoudre un CVRPTW.

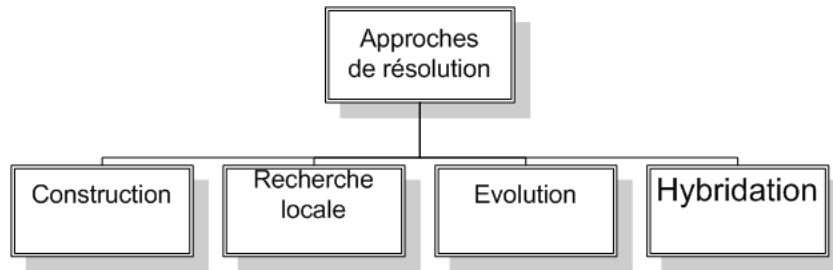


FIGURE 7 – Classification des familles de méthodes de résolution du VRP selon Hao et al. [1999].

tabou et le recuit simulé [Rego and Roucairol, 1994]. En effet, ces méthodes sont basées sur une *recherche locale* de meilleures solutions dans le voisinage de la solution courante.

Cette remarque nous conduit à revoir la classification que nous avons dressée dans la figure 3 et l’adapter en fonction des approches utilisées dans les méthodes de résolution des problèmes d’optimisation. Dans ce cadre, Hao et al. [1999] et Defaix [2010] distinguent quatre classes : la *construction*, la *recherche locale* (dite aussi *recherche à base de voisinage*), l’*évolution* et l’*hybridation*. Dans un livre récent, Zäpfel et al. [2010] ont fait le tour des principales classifications des métaheuristiques élaborées dans la littérature. Ces classifications sont basées sur plusieurs critères, tels que l’utilisation ou non de la mémoire, l’inspiration de la nature, la fonction objectif etc. Nous retiendrons leur classification des méthodes de recherche en méthodes basées sur une *construction* répétée ou une *modification* répétée ou une *recombinaison* répétée de solution (*Repeated Solution Construction*, *Repeated Solution Modification* et *Repeated Solution Recombination*).

3.3 Algorithmes génétiques pour la résolution de VRP

3.3.1 Principe général

Nous développons ici l’une des métaheuristiques les plus utilisées pour la résolution du problème de routage de véhicules : les algorithmes génétiques. Il s’agit d’une méthode bio-inspirée introduite par Holland [1975] dans le cadre d’une analogie avec la sélection naturelle des espèces. Elle a été formalisée ensuite par Goldberg [1989] pour être appliquée à la résolution de problèmes d’optimisation.

Un algorithme génétique entretient une *population* de solutions au problème à optimiser. La qualité de ces solutions (appelées *individus*) est donnée par une *fonction d’évaluation* (dite *fitness*) f . L’algorithme applique des opérateurs génétiques de *croisement* et de *mutation* à la population en vue d’en générer une autre. À partir de deux individus *parents*, l’opérateur de croisement génère

un ou deux individus *enfants* héritant chacun une partie de la structure de ses parents. L'opérateur de mutation agit quant à lui sur un seul individu en modifiant une partie de sa structure.

L'idée sous-jacente à ces deux opérateurs correspond à deux principes de la sélection naturelle des espèces : les meilleurs éléments se reproduisent entre eux et assurent l'amélioration de la population en transférant leur information génétique; et de temps à autre quelques éléments mutent et maintiennent une diversité génétique au sein de la population.

En pratique, pour mettre en place un tel algorithme, on doit disposer des « ingrédients » suivants :

1. une représentation d'une solution du problème (modélisation ou *codage* d'un individu sous forme d'un *chromosome*)
2. une manière de générer une population initiale d'individus
3. une manière de qualifier ou évaluer un individu donné (calcul de la *fitness*)
4. une manière de sélection des meilleurs individus (principe de sélection)
5. une manière de croiser deux individus (opérateur de croisement ou *recombinaison*) et de muter un individu (opérateur de mutation)
6. des paramètres de réglage de l'algorithme, principalement la taille de la population de solutions, les probabilités d'application des opérateurs de croisement et de mutation, ainsi que le critère d'arrêt de l'algorithme.

Les phases de modélisation des individus et de définition des opérateurs génétiques sont primordiales pour le bon fonctionnement de la méthode. Le codage peut être *direct* si le chromosome contient toute l'information de la solution, ou *indirect* si une étape de transformation est nécessaire pour passer du chromosome à la solution [Housroum, 2005]. Cette étape de « décodage » n'est pas forcément un inconvénient car même si les codages directs sont naturels et intuitifs, ils peuvent générer des solutions qui violent les contraintes du problème. L'application d'une procédure de réparation –éventuellement coûteuse en temps de calcul– serait alors nécessaire. Un autre avantage des codages indirects réside dans leur compatibilité avec les opérateurs génétiques.

Un codage basique et très classique consiste à représenter une solution sous forme d'une chaîne de bits pouvant prendre deux valeurs possibles 0 ou 1. Ce codage s'appelle *codage binaire* et fut historiquement l'un des premiers encodages employés. Il permet en effet de définir facilement des opérateurs basiques de mutation et de croisement. La figure 8 illustre un opérateur de mutation modifiant un bit sélectionné aléatoirement. La figure 9 illustre deux opérateurs de croisement qui génèrent des enfants en échangeant des sous-structures des parents. L'échange peut s'effectuer en un seul ou en plusieurs points de croisement.

Le principe de sélection peut s'effectuer en choisissant les p premiers meilleurs individus ou en attribuant à chaque individu i une probabilité de sélection proportionnelle à sa fitness f_i ou à sa fitness relative $\frac{f_i}{\sum f_j}$.

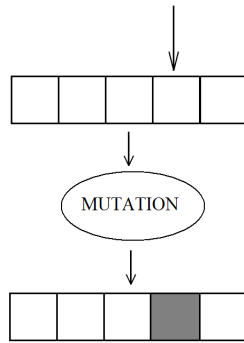


FIGURE 8 – Exemple d'opérateur de mutation basique.

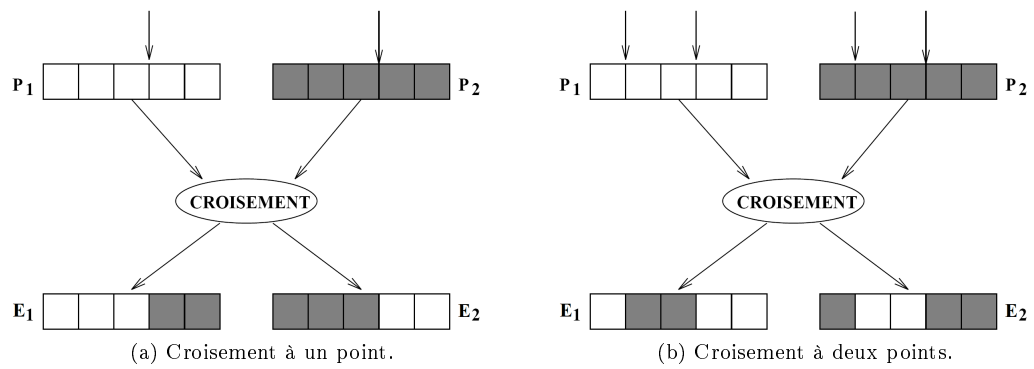


FIGURE 9 – Exemple de deux opérateurs de croisement basiques.

Le critère d'arrêt est exprimé généralement sous forme d'un nombre maximal de générations ou d'une durée maximale d'exécution.

3.3.2 Application au VRP

Les algorithmes génétiques ont été appliqués avec succès à la résolution du problème du voyageur de commerce, et ont donc naturellement été adaptés aussi à la résolution du problème de routage de véhicules dans ses variantes courantes. Ces algorithmes sont tout particulièrement performants sur le VRPTW, et ont produit certaines des meilleures solutions connues sur les problèmes tests de Solomon⁸ [Gendreau et al., 2003].

Les adaptations des algorithmes génétiques au VRP sont tellement nombreuses que l'état de l'art dressé par Haj-Rachid et al. [2010] dénombre 5 types de codage, 2 méthodes de création de la population initiale, 3 méthodes d'évaluation des individus, 5 types de sélection des parents, 22 types de croisements, 6 types de mutations, 4 stratégies de construction de la génération suivante, et 3 types de critères d'arrêt.

Notre but n'étant pas de détailler exhaustivement toutes les adaptations possibles, nous nous contentons de deux parmi les meilleures d'entre elles [Crainic and Semet, 2006] : il s'agit des adaptations de Baker and Ayechew [2003] et de Prins [2004] appliquées au VRP avec contraintes de capacités (CVRP). Pour les illustrer, nous utiliserons l'exemple de la figure 1 (VRP ayant $n = 20$ clients).

Codage Un exemple de codage direct d'une solution de VRP consiste à représenter celle-ci sous forme de plusieurs séquences de clients dont chacune correspond à une tournée. Des délimiteurs, appelés *séparateurs de tournées*, peuvent être insérés entre les séquences en question pour séparer les tournées. À titre d'exemple, la solution de la figure 1 peut être représentée par le chromosome suivant :

0	20	1	2	4	3	0	5	6	7	10	0	8	9	11	12	13	14	0	16	15	17	18	19	0
---	----	---	---	---	---	---	---	---	---	----	---	---	---	----	----	----	----	---	----	----	----	----	----	---

Le séparateur de tournée utilisé est le symbole **0** (le dépôt) et permet de distinguer le début et la fin de chaque tournée. L'ordre des clients est important car il détermine le sens dans lequel sont effectuées les tournées.

En supprimant le dépôt de ce chromosome, on obtient un deuxième chromosome correspondant à un codage indirect de la même solution :

20	1	2	4	3	5	6	7	10	8	9	11	12	13	14	16	15	17	18	19
----	---	---	---	---	---	---	---	----	---	---	----	----	----	----	----	----	----	----	----

En effet, cette représentation peut être vue comme une seule méta-tournée d'un voyageur du commerce. Elle indique certes l'ordre des clients dans les tournées, mais ne permet pas de connaître les débuts et les fins de celles-ci. Une procédure

⁸. présentés dans l'annexe B.1.

supplémentaire de découpage est donc utilisée. Elle se base généralement sur les contraintes du problème telles que la capacité des véhicules ou les fenêtres temporelles des clients.

Prins [2004] utilise ce deuxième type codage car les codages avec séparateurs de tournées ne se sont pas avérés très efficaces à cause des procédures de réparation auxquelles elles doivent recourir. La procédure de découpage (appelée SPLIT) appliquée à un chromosome consiste à résoudre un problème de plus court chemin [Haj-Rachid et al., 2010].

Ces deux premiers codages sous forme de cycles correspondent à une approche de type *route first-cluster second*. Une approche de type *cluster first-route second* donnerait lieu à un codage de type *affectation* [Crainic and Semet, 2006] : le chromosome indique pour chaque client le n° du véhicule qui le dessert. Il s'agit ainsi d'un codage indirect puisqu'il ne donne pas d'information sur l'ordre des tournées au sein des partitions de clients.

Baker and Ayechev [2003] utilisent ce codage, et représentent la solution de la figure 1 avec le chromosome suivant :

1	1	1	1	2	2	2	3	3	2	3	3	3	3	4	4	4	4	4	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Opérateurs génétiques

Opérateur de croisement Pour illustrer les opérateurs de croisement, nous utiliserons la solution de la figure 1 comme premier parent (P1), et une autre solution comme deuxième parent P2 (problème à $n = 20$ clients et résolu avec $m = 4$ véhicules). Les opérateurs s'effectuent à deux points, entre les positions 6-7 et 15-16 du chromosome.

Baker and Ayechev [2003] utilisent un opérateur classique de croisement à deux points, identique à celui de la figure 9b. Le croisement entre les deux solutions de la figure 10 s'effectue de cette manière :

Les chromosomes des deux solutions enfants héritent chacun d'une partie de ceux des parents. Cet héritage s'effectue comme suit : pour l'enfant E1 (respectivement E2), les éléments situés entre les points de croisement sont hérités du parent P1 (respectivement P2) et les éléments restants sont hérités du parent

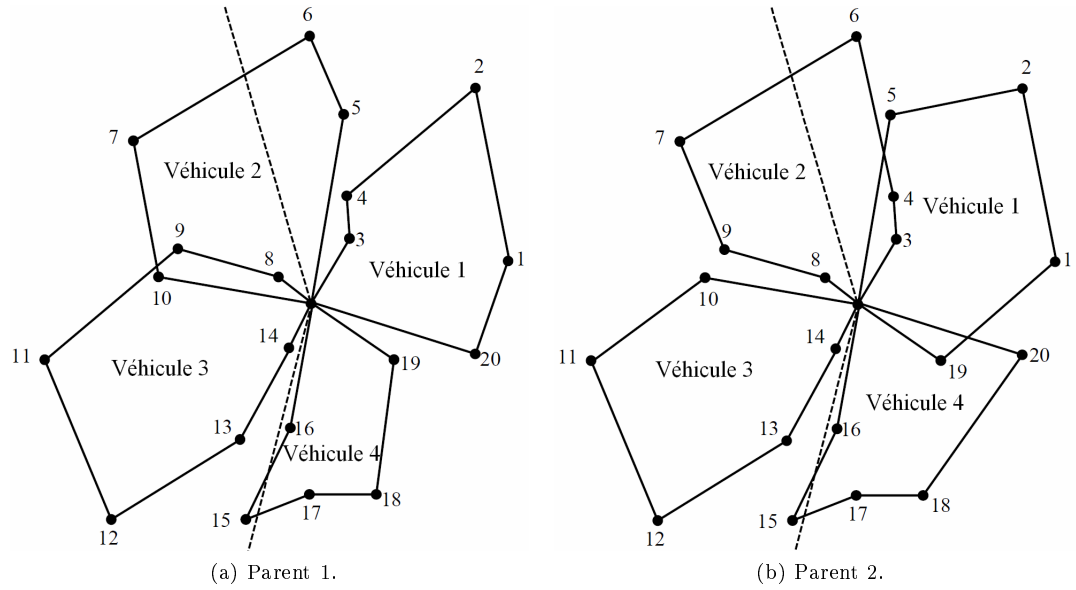


FIGURE 10 – Deux solutions utilisées comme parents pour illustrer l’opérateur de croisement de Baker and Ayechew [2003].

P2 (respectivement P1) :

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1 :	1	1	1	1	2	2	2	3	3	2	3	3	3	3	4	4	4	4	4	1
P2 :	1	1	2	2	1	2	2	2	2	3	3	3	3	3	4	4	4	4	1	4

⇓

E1 :							2	3	3	2	3	3	3	3	4					
E2 :							2	2	2	3	3	3	3	3	4					

⇓

E1 :	1	1	2	2	1	2	2	3	3	2	3	3	3	3	4	4	4	4	1	4
E2 :	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	1

D’après le chromosome de la solution E2, on sait que le véhicule 1 (respectivement 2, 3, 4) assure la tournée entre les clients {1, 2, 3, 4, 20} (respectivement {5, 6, 7, 8, 9}, {10, 11, 12, 13, 14}, {15, 16, 17, 18, 19}). L’information manquante est l’ordre dans lequel s’effectuent ces tournées : un TSP est alors résolu pour chaque tournée. Les tournées résultantes sont dessinées dans la figure 11.

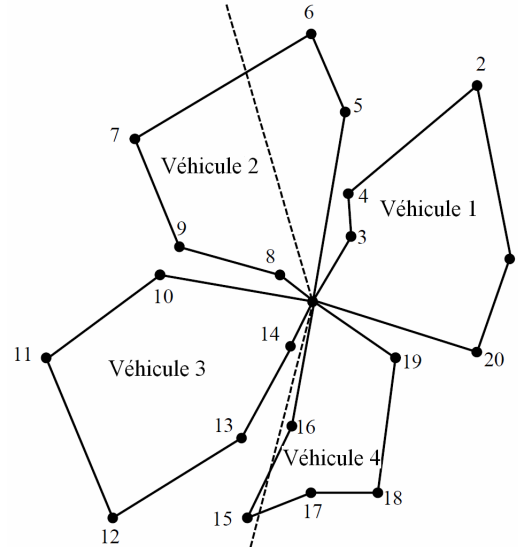


FIGURE 11 – Enfant E2 généré par le croisement des parents de la figure 10 [Baker and Ayechew, 2003].

Nous utiliserons une autre solution parent P2 pour illustrer l'opérateur de croisement de Prins ; son chromosome est le suivant :

17	18	19	20	1	3	4	2	6	5	8	7	9	10	11	14	13	12	15	16
----	----	----	----	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Le codage de Prins étant proche de celui du codage d'un TSP, alors l'opérateur de croisement utilisé est l'opérateur *Order Crossover (OX)*, un opérateur classique pour le TSP, dont le principe est le suivant :


1. Pour l'enfant E1 (respectivement E2), les éléments situés entre les points de croisement sont hérités du parent P1 (respectivement P2),
2. Les éléments restants de l'enfant E1 (respectivement E2) sont hérités aussi du parent P1 (respectivement P2) mais ne sont pas recopiés tels quels (sinon E1 (resp. E2) serait identique à P1 (resp. P2)). Ils sont recopiés selon leur ordre d'apparition dans l'autre parent P2 (resp. P1), à partir de la position située juste après le deuxième point de croisement.

Dans l'exemple, les éléments entre les points de croisement dans E2 sont mis en évidence en gris clair. Les éléments restants sont $\{20, 1, 2, 4, 3, 5, 16, 15, 17, 18, 19\}$ et sont mis en évidence en gris foncé. Ces éléments sont soulignés dans le deuxième parent P2. À partir de la 16ème position (position située juste après le deuxième point de croisement), ces éléments apparaissent dans P2 selon l'ordre $\{15, 16, 17, 18, 19, 20, 1, 3, 4, 2, 5\}$ et sont rajoutés à E1 dans ce même ordre d'apparition :

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1 :	20	1	2	4	3	5	6	7	10	8	9	11	12	13	14	16	15	17	18	19
P2 :	17	18	19	20	1	3	4	2	6	5	8	7	9	10	11	14	13	12	15	16

↓

E1 :							6	7	10	8	9	11	12	13	14					
P2 :	<u>17</u>	<u>18</u>	<u>19</u>	<u>20</u>	<u>1</u>	<u>3</u>	<u>4</u>	<u>2</u>	6	<u>5</u>	8	7	9	10	11	14	13	12	<u>15</u>	<u>16</u>



↓

E1 :	<u>20</u>	<u>1</u>	<u>3</u>	<u>4</u>	<u>2</u>	<u>5</u>	6	7	10	8	9	11	12	13	14	<u>15</u>	<u>16</u>	<u>17</u>	<u>18</u>	<u>19</u>
------	-----------	----------	----------	----------	----------	----------	---	---	----	---	---	----	----	----	----	-----------	-----------	-----------	-----------	-----------

De manière similaire, les éléments rajoutés à l'enfant E2 sont {15, 16, 17, 18, 19, 20, 1, 3, 4, 2, 5}, le chromosome obtenu est alors :

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P2 :	17	18	19	20	1	3	4	2	6	5	8	7	9	10	11	14	13	12	15	16
E2 :	20	1	3	4	2	5	4	2	6	5	8	7	9	10	11	15	16	17	18	19


Notons qu'un autre opérateur classique, dit *Linear Order Crossover (LOX)*, opère de manière similaire à *OX* : après la recopie des éléments entre les deux points de croisement, la recopie des éléments restants commence à la 16ème position (juste après le deuxième point de croisement) s'effectue selon leur ordre d'apparition dans l'autre parent. Cependant, dans *LOX*, cet ordre est calculé à partir de la première position, et non pas la 16ème.

L'application de *LOX* aux deux parents traités précédemment donne que l'enfant E1, les éléments restants sont recopiés dans l'ordre {17, 18, 19, 20, 1, 3, 4, 2, 5, 15, 16}.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1 :	20	1	2	4	3	5	6	7	10	8	9	11	12	13	14	16	15	17	18	19
P2 :	17	18	19	20	1	3	4	2	6	5	8	7	9	10	11	14	13	12	15	16

↓

E1 :							6	7	10	8	9	11	12	13	14					
P2 :	<u>17</u>	<u>18</u>	<u>19</u>	<u>20</u>	<u>1</u>	<u>3</u>	<u>4</u>	<u>2</u>	6	<u>5</u>	8	7	9	10	11	14	13	12	<u>15</u>	<u>16</u>



↓

E1 :	<u>3</u>	<u>4</u>	<u>2</u>	<u>5</u>	<u>15</u>	<u>16</u>	6	7	10	8	9	11	12	13	14	<u>17</u>	<u>18</u>	<u>19</u>	<u>20</u>	<u>1</u>
------	----------	----------	----------	----------	-----------	-----------	---	---	----	---	---	----	----	----	----	-----------	-----------	-----------	-----------	----------

En dépit de leur principe simple, les deux opérateurs de croisement *OX* et *LOX* sont les plus utilisés dans la littérature [Haj-Rachid, 2010]. Prins [2004] a choisi *OX* car il s’est avéré plus efficace que *LOX* lors des expérimentations.

Opérateur de mutation Prins n’applique pas un vrai opérateur de croisement mais le remplace par une procédure d’amélioration basée sur une recherche locale de type *2-opt* (comme vu dans la sous-section 3.2.3). Voilà pourquoi son algorithme génétique est qualifié d’*hybride*⁹.

Baker and Ayechew utilise un opérateur de mutation classique basé sur l’échange aléatoire de deux clients entre deux tournées.

Population initiale Les deux approches utilisent des heuristiques simples pour générer la population initiale; telles que la méthode des économies et la méthode de balayage vues dans la sous-section 3.2. Haj-Rachid et al. [2010] identifient deux autres moyens d’initialiser les algorithmes génétiques : générer la population de façon totalement aléatoire, ou combiner la génération aléatoire et le recours aux heuristiques.

Fitness Le plus souvent, la fonction d’évaluation (fitness) est calculée à partir d’une somme pondérée de plusieurs objectifs (méthode d’agrégation).

Berger et al. [1998] ont par exemple proposé la formule

$$fitness = R - R_{min} + \frac{\min(D, 2D_{min})}{D_{min}}$$

où :

- R est le nombre de tournées dans la solution à évaluer,
- R_{min} est le nombre de tournées dans la meilleure solution dans la population courante,
- D est la distance totale parcourue dans la solution à évaluer
- et D_{min} est la distance totale parcourue dans la meilleure solution de la population courante.

Cette formulation de la fonction objectif implique que les meilleures solutions recherchées ont peu de tournées et des distances totales parcourues petites.

La formulation de la fitness proposée par Sushil and Rilun [1999] permet de prendre en compte les fenêtres temporelles

$$fitness = w_1 * D + w_2 * T + w_3 * R_1 + w_4 * R_2 + w_5 * O$$

où :

- D est la distance totale parcourue par les véhicules,
- T est la durée totale des différentes tournées,

9. un tel algorithme est appelé aussi *algorithme mémétique*.

- R_1 est le retard total des véhicules chez les clients,
 - R_2 est le retard total des véhicules au dépôt,
 - O est la surcharge totale de capacité des véhicules
 - et les w_i sont les poids¹⁰ (coefficients) de pondération de ces objectifs.
- Housroum [2005], qui a repris cette formulation générale, a comparé les formulations de 17 publications différentes traitant du VRPTW. Il en conclue que les objectifs les plus utilisés sont le nombre de véhicules utilisés et la distance totale parcourue.

3.4 Comparaison entre les différentes méthodes de résolution

Avant de clore cette section 3, nous proposons de donner un aperçu sur l'efficacité des méthodes et des techniques qui y ont été mentionnées, ainsi qu'une comparaison de ces efficacités.

Précisons tout d'abord que les méthodes exactes (telles que *Branch & Bound*) ne sont pas efficaces sur des problèmes ayant plus que $n = 25$ clients [Rego and Roucairol, 1994], et qu'elles ne sont pas utilisées à partir de $n = 50$ clients [Laporte, 1992]. Nous discuterons donc plutôt de l'efficacité des méthodes approchées.

Étant donné la diversité des variantes du VRP et des problèmes tests de la littérature, il est difficile d'effectuer une comparaison globale des différentes méthodes approchées. Bräysy and Gendreau [2005a] et Bräysy and Gendreau [2005b] ont cependant procédé à une telle comparaison entre les méthodes locales et les heuristiques de construction d'une part, et entre les métaheuristiques d'autre part. La référence de la comparaison est la performance de la résolution des 56 instances tests du VRP avec fenêtres temporelles. Il s'agit des problèmes de Solomon [1987] présentés avec plus de détails dans l'annexe B.1.

La méthodologie de comparaison est basée sur l'évaluation, pour chaque méthode, des critères de performance suivants :

1. le temps CPU d'exécution de la méthode¹¹,
2. les moyennes des nombres de véhicules utilisés ainsi que celles des distances totales parcourues, et ce pour chacune des 6 classes de problèmes,
3. et deux autres indicateurs globaux *CNV/CTD* (*Cumulative Number of Vehicles/Cumulative Total Distance*) représentant le nombre cumulé des véhicules utilisés ainsi que la distance totale cumulée sur l'ensemble des 56 problèmes.

10. les valeurs numériques des poids utilisés par les auteurs sont $w_1 = 5, w_2 = 156, w_3 = 50, w_4 = 25, w_5 = 50$.

11. le temps est normalisé selon la procédure de normalisation de Dongarra [1998].

En dépit des difficultés de comparaison dues notamment à la non mention de quelques auteurs des durées d'exécution de leurs méthodes ou à la prise en compte de différents objectifs d'optimisation, nous pouvons tout de même tirer quelques conclusions générales de la procédure de comparaison.

Heuristiques Les approches heuristiques de *construction* sont bien plus rapides que les méthodes d'amélioration par recherche locale mais produisent des solutions de moins bonne qualité car elles sont moins sophistiquées.

Métaheuristiques Les métaheuristiques comparées sont essentiellement les algorithmes génétiques et la recherche tabou. Les paramètres qui caractérisent ces méthodes sont mentionnés car ils peuvent avoir des conséquences importantes sur leur efficacité. Il s'agit principalement de la procédure d'élaboration de la solution (ou population) initiale ainsi que des opérateurs appliqués (opérateurs de voisinage pour une recherche tabou et opérateurs de croisement/mutation pour un algorithme génétique).

Bräysy and Gendreau [2005b] comparent les algorithmes génétiques et les recherches tabous et Bräysy and Gendreau [2001b] dressent une comparaison plus détaillée entre différents algorithmes génétiques appliqués à la résolution des instances de Solomon.

Ces deux comparaisons montrent que même si les recherches tabous peuvent être efficaces sur le VRPTW, elles restent en général dominées par les algorithmes génétiques qui ont des performances bien meilleures. Il est important de souligner que la tendance est inversée lorsqu'il s'agit de VRP sans fenêtres temporelles : le premier algorithme génétique qui arrive à rivaliser avec les recherches tabous fut celui de Prins [2004] que nous avons présenté dans la sous-section 3.3.2.

De manière générale, les méthodes heuristiques les plus coûteuses en temps n'excèdent pas les 30 minutes pour résoudre les problèmes de Solomon alors que certaines métaheuristiques nécessitent 10 heures d'exécution. Cependant, les métaheuristiques les plus efficaces nécessitent au moins 1h30 de temps d'exécution.

Conclusion Nous retiendrons qu'en termes de qualité des solutions obtenues, les métaheuristiques sont généralement meilleures que les heuristiques; et qu'en termes de temps d'exécution la comparaison est inversée. Ce résultat est prévisible car plusieurs métaheuristiques appliquent elles-mêmes des heuristiques d'amélioration sur leurs solutions (hybridation des méthodes).

Nous retiendrons aussi que la recherche tabou et les algorithmes génétiques sont les deux métaheuristiques les plus efficaces sur les principales variantes du VRP. Les algorithmes génétiques sont plus performants que la recherche tabou sur le VRPTW, et la recherche tabou était plus efficace sur le VRP (sans les

fenêtres temporelles) mais a été rattrapée par les algorithmes génétiques depuis les approches de Baker and Ayechew [2003] et Prins [2004], présentées dans la sous-section précédente 3.3.2.

4 Résolution interactive de problèmes de transport

Dans cette section nous proposons l'étude des humains en résolution de problèmes de transport tels que le TSP et le VRP. Nous nous intéressons plus particulièrement à la résolution totalement manuelle et à la résolution « interactive » où l'humain interagit avec une machine pour résoudre les problèmes en question. Les principales contributions historiques dans ces deux approches sont reportées dans l'annexe A.

4.1 Résolution humaine de VRP/TSP

4.1.1 Efficacité de la résolution humaine *manuelle*

Comme le souligne Michie et al. [1968], avant le développement des heuristiques et des capacités de calcul des machines, les problèmes de transport étaient résolus par des humains : le problème est dessiné sur un support papier, et les humains tracent les tournées à la main, tout comme dans la figure 13 tirée de Ormerod [2005]. La figure 16 montre que cette résolution manuelle a laissé la place à une résolution totalement automatique par les heuristiques (à partir des années 1965) puis par les métaheuristiques (à partir de la fin des années 1980).

Au cours de la dernière décennie, la résolution humaine des problèmes de transport, notamment du TSP, a été largement étudiée par des chercheurs en psychologie cognitive. Des évaluations et des modélisations des performances humaines ont pu être effectuées à travers des approches empiriques et des études expérimentales. La majorité de ces travaux sont répertoriées dans *The Journal of Problem Solving*¹², journal créé en 2006, dont le premier numéro était entièrement dédié au TSP. La figure 12, emblème de ce journal, montre un humain (le penseur de Rodin) à face à un problème de TSP et évoque les motivations de l'étude des performances humaines de résolution.

MacGregor and Ormerod [1996] ont étudié les performances humaines sur des TSP avec 10 et 20 villes. Les résultats obtenus avec respectivement 58 et 29 sujets de tests montrent que les solutions humaines sont en général de bonne qualité et que certaines solutions parmi elles sont très proches de l'optimalité.

Dans le même esprit, Best [2004] a conduit une étude empirique de la performance humaine en résolution de TSP en vue de la modéliser. Ses observations

12. <http://docs.lib.purdue.edu/jps/>.

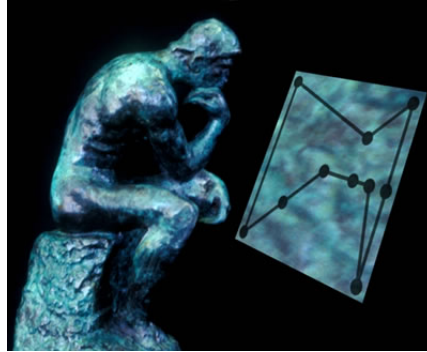


FIGURE 12 – Résolution humaine de TSP : comment fait un humain pour résoudre un TSP ? (tirée de *The Journal of Problem Solving*).

rejoignent celles de MacGregor and Ormerod [1996], mais les conclusions principales qu'elles permettent de déduire sont plus détaillées [Best, 2005] :

1. le temps de la résolution humaine est pratiquement linéaire avec la taille du problème,
2. la qualité des solutions humaines est généralement 5% en dessous de celle des solutions optimales (en termes de distance parcourue),
3. la résolution des problèmes est précédée de 2-3 secondes pendant lesquelles le problème est analysé et une solution approximative est développée,
4. l'effet de l'apprentissage est minime.

Dans une approche originale, Chronicle et al. [2006a] ont proposé à des sujets de tests deux tâches sur un TSP : une tâche ordinaire d'optimisation (minimisation de la distance totale parcourue) et une tâche de *pessimisation* ou de « dégradation » (*pessimization*) où les sujets doivent trouver le chemin le plus long possible. Les résultats obtenus montrent que les performances d'optimisation (6% à 8% de l'optimalité) sont nettement meilleures que celles de la *pessimisation* (16% et 49% de l'optimalité). Cette observation soutient l'hypothèse que dans le cadre de l'optimisation, les humains peuvent facilement appliquer des heuristiques de recherche.

Nous allons présenter maintenant les principaux modèles qui ont été conçus pour ces heuristiques humaines.

4.1.2 Modèles des stratégies humaines de résolution

Partant de la constatation que le temps de résolution humaine est pratiquement linéaire avec la taille du problème, Graham et al. [2000] ont établi un modèle de la performance humaine basé sur le mélange de deux stratégies : un groupement visuel (*perceptual grouping*) et une recherche locale itérée.

Se basant sur des expériences conduites par Chronicle et al. [2006b], MacGregor et al. [2006] ont quand à eux comparé les performances humaines sur la résolution de problèmes de TSP (12 problèmes à 12 clients et 6 problèmes à 15 clients, 25 et 24 sujets de test) à celles de trois heuristiques : l'heuristique de l'enveloppe convexe (*convex hull*), celle du plus proche voisin (*nearest neighbor*) et celle de l'évitement de croisement (*crossing avoidance*). Plus précisément, les solutions trouvées sont comparées en termes de la longueur de leurs chemins, de leur chevauchement avec les autres solutions et de leur nombre de croisements entre chemins. Les résultats expérimentaux obtenus montrent que d'une part les sujets de tests arrivent à trouver des bonnes solutions sans trop de difficultés, et que d'autre part leurs stratégies de résolution se rapprochent le plus de l'heuristique de l'enveloppe convexe.

Van Rooij et al. [2003] ont étudié l'effet de l'âge sur la stratégie de résolution humaine. Pour ce faire, ils ont comparé les performances de sujets de tests de trois catégories d'âges différentes (7 ans, 12 ans et des adultes). Les auteurs expliquent les bonnes performances des trois catégories par le *traitement perceptuel* qu'elles mettent en place. Cependant, les performances exceptionnelles de certains sujets adultes laissent penser que ces derniers emploient leurs capacités cognitives d'analyse pour renforcer leurs capacités visuelles.

Nous mentionnons finalement que les études de la résolution humaine de problèmes de routage de véhicules ne sont pas aussi nombreuses que celles dédiées au TSP. Ormerod [2005] ont par exemple considéré le VRP avec contraintes de capacités. La figure 13 montre un exemple de problème à 45 clients à résoudre avec 6 véhicules de capacité 100 unités et un exemple de solution humaine manuelle. Ormerod [2005] ont identifié des stratégies efficaces chez les participants, notamment la stratégie de regroupement des clients et la stratégie « *Draw 'by eye' then refine by counting* » (dessiner puis compter) qui propage les contraintes de capacité des véhicules après une construction visuelle des routes.

4.2 Taxonomie des niveaux d'implication d'un humain dans le processus de résolution

Fisher, 1986 a proposé de coupler les capacités visuelles humaines avec la vitesse de calcul des machines en produisant un style d'optimisation hybride qu'il a baptisé « optimisation interactive ». Dans son approche, l'humain dispose d'un système de visualisation de la solution et peut être amené à intervenir pendant l'exécution de l'algorithme (ou la méthode) d'optimisation en proposant des idées [Jones, 1996].

Une telle approche s'inscrit dans le cadre de la conception de systèmes d'aide à la décision. Rappelons à ce titre qu'un tel système est défini comme « *un outil qui permet aux décideurs de combiner jugement personnel et soutien informatique,*

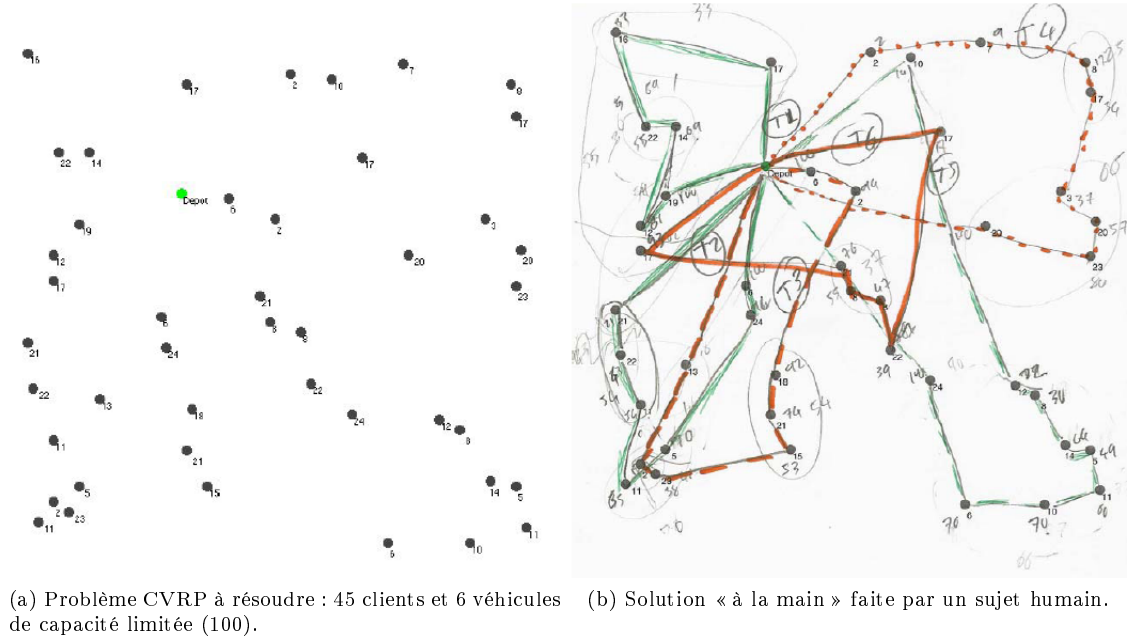


FIGURE 13 – Exemple de résolution humaine manuelle d'un CVRP.

dans une interface machine pour produire des informations utiles au processus décisionnel ».

Par abus de langage, le terme « résolution interactive » peut s'utiliser par simple opposition à « résolution manuelle », comme on l'a vu dans la sous-section 4.1, ou à « résolution automatique ». Il peut aussi s'utiliser pour qualifier le processus de résolution dans un système d'aide à la décision ou dans tout autre système disposant d'une interface homme-machine. Dans les approches multi-critères (multi-objectifs), quand le décideur exprime ses préférences avant (respectivement après) la résolution du problème, celle-ci est dite *a priori* (respectivement *a posteriori*); et quand le décideur exprime ses préférences au cours de la résolution (en répondant à des questions par exemple), celle-ci est dite interactive [Collette and Siarry, 2003, Branke et al., 2008].

Wezel et al. [2011] proposent une terminologie plus rigoureuse et plus détaillée englobant les applications « interactives » et détaillant les niveaux d'implication de l'humain et d'un algorithme dans la réalisation d'une tâche d'optimisation. La figure 14 résume cette terminologie et montre cinq modes différents :

1. le mode *manuel* où l'humain réalise toutes les activités de la tâche,
 2. le mode de *contrôle mutuel* où l'humain et l'algorithme participent tous les deux à la réalisation de la tâche.
- Deux sous-modes sont distingués :

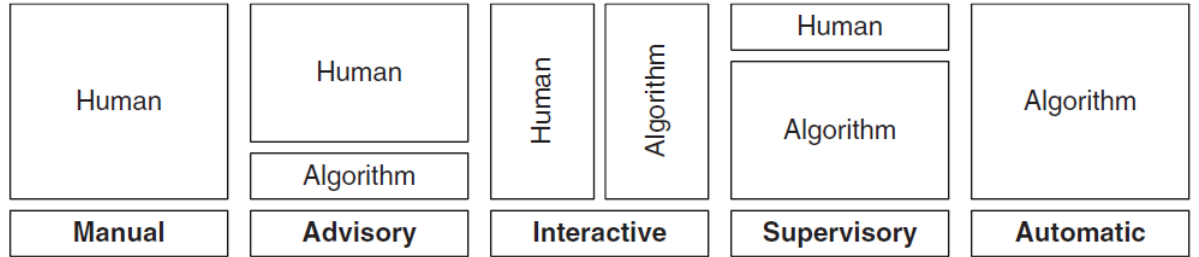


FIGURE 14 – Niveaux d’implication d’un humain dans Wezel et al. [2011].

- (a) le mode *consultatif* (*advisory*) où l’humain propose des solutions (complètes ou partielles) et l’algorithme les vérifie et les évalue,
- (b) le mode de *supervision* (*supervisory*) où l’algorithme propose des solutions (complètes ou partielles) et l’humain les vérifie et les évalue.
3. le mode *interactif* où l’humain et l’algorithme peuvent proposer des solutions, et chaque solution proposée est vérifiée et évaluée par chacun d’entre eux,
4. le mode *automatique* où l’algorithme effectue toutes les activités de la tâche.

Cette terminologie a été adoptée par Gacias [2010] pour la mise en place d’un système d’aide à la décision interactive pour le problème de routage de véhicules. Dans ce système, le mécanisme de résolution du VRP s’effectue selon l’approche partition d’abord - routes ensuite et trois modes de contrôle (consultatif, de supervision et interactif) sont possibles pour les deux phases du processus de résolution : l’affectation des clients aux véhicules et l’élaboration des tournées.

4.3 Approches de résolution interactive de VRP

4.3.1 Premières tentatives de résolution interactive

Les travaux de Michie et al. [1968] font partie des premières études abordant l’interaction au cours de la résolution d’un problème d’optimisation. Ce problème est un TSP à 50 villes résolu de trois manières différentes : manuellement par un humain disposant d’une feuille et d’un stylo, automatiquement par une heuristique basique de recherche locale avec backtrack, et « interactivement » par un humain disposant d’un système d’affichage de la solution, d’un stylo optique et pouvant être aidé par l’heuristique mentionnée. Les résultats obtenus montrent la supériorité de l’association homme-machine sur les deux autres approches.

Krolak et al. [1970] et Krolak et al. [1972] proposent une autre approche de résolution interactive de TSP, se déroulant en quatre étapes illustrées dans la figure 15 :

1. la machine produit des groupements de villes (dits *clusters* de villes) ainsi que quelques segments de routes en utilisant des heuristiques,
2. l'humain relie les villes au sein des groupes puis relie les groupes entre eux,
3. la machine tente d'améliorer cette solution par des heuristiques,
4. la solution obtenue est affichée à l'utilisateur ; s'il est satisfait la procédure s'arrête.

Cette approche a été appliquée à des instances de TSP à 200 clients et s'est avérée très coûteuse en ressources machine et surtout en ressources humaines (monopolisation de l'humain) tout en étant sous-optimale [Lin and Kernighan, 1973].

Les approches de Waters [1984] et Fisher [1986] s'intègrent dans le cadre de systèmes d'aide à la décision pour le problème de routage de véhicules. L'humain dispose d'un affichage graphique de la solution au problème d'optimisation et d'un ensemble d'outils de manipulation de celle-ci. Il peut par exemple afficher son coût, lui appliquer des procédures d'amélioration etc.

4.3.2 Approche *Human-Guided Search* (HuGS)

Le projet *Human-Guided Search* (HuGS) [Klau et al., 2009] s'inscrit dans le cadre des systèmes d'optimisation interactive et s'intéresse directement à la question suivante : comment les humains peuvent-ils interagir de manière efficace avec les systèmes d'optimisation ?

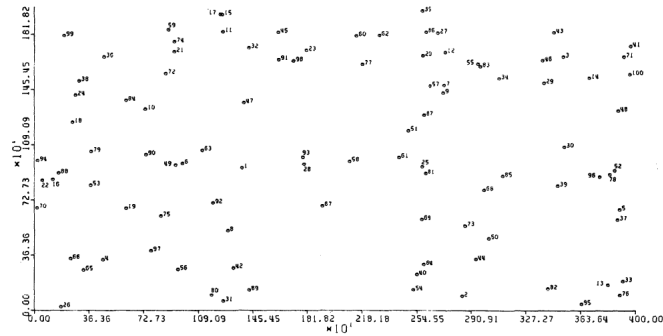
La réponse apportée dans HuGS fut de diviser le processus d'optimisation clairement entre l'humain et un algorithme de recherche de la manière suivante :

- la machine trouve les optima locaux,
- l'humain guide l'algorithme à sortir de ces optima et améliorer la solution : il annote visuellement les éléments du problème pour contraindre l'algorithme.

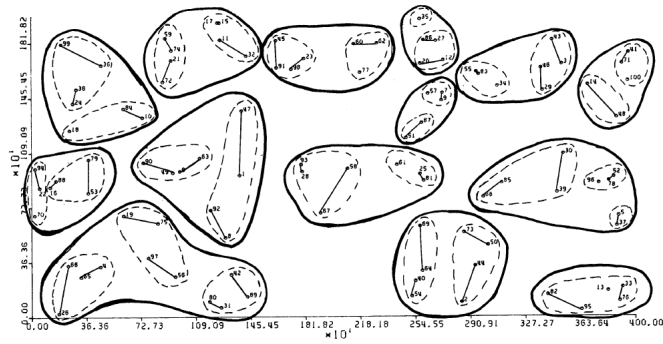
Outre le guidage de la recherche, les utilisateurs peuvent modifier une solution à la main ou revenir à une solution précédente. Cette approche a été appliquée à huit problèmes d'optimisation dont le VRP avec contraintes de capacité et fenêtres temporelles (CVRPTW) ainsi qu'à une variante du TSP. Les résultats expérimentaux montrent que 10 minutes d'une recherche tabou guidée équivalent en moyenne à 70 minutes de recherche tabou non guidée.

Scott et al. [2002] ont quant à eux étudié de manière détaillée la capacité des humains à guider la recherche. Ils ont demandé à 3 sujets de test d'effectuer 3 types de tâches différentes sur une variante du TSP :

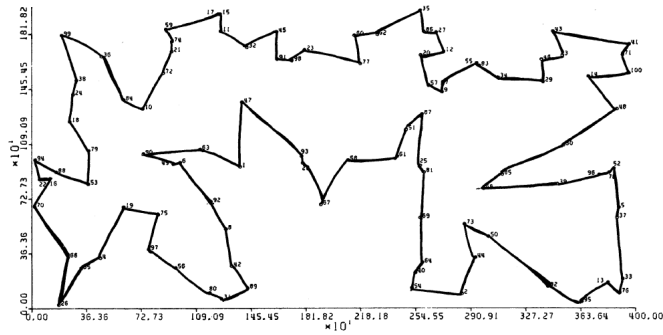
1. une tâche de « *focus* » où ils doivent mieux guider l'algorithme,
2. une tâche de recherche de clients cibles où ils doivent sélectionner le maximum de « *bons* » clients,
3. et une tâche de « *stopping* » où ils doivent arrêter un algorithme de recherche au moment où ils estiment qu'il a effectué la meilleure amélioration.



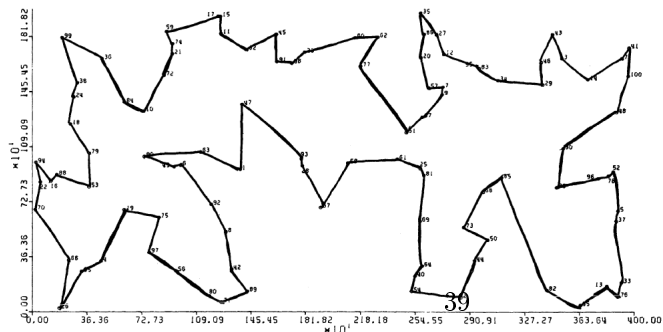
(a) Problème à résoudre (100 villes).



(b) Étape 1 : la machine fait des groupements et dessine quelques segments de routes.



(c) Étape 2 : résolution humaine (routes intra et inter-groupes).



(d) Étape 3 : amélioration de la solution humaine par la machine (heuristiques).

FIGURE 15 – Approche de Krolak et al. [1970] pour la résolution interactive d'un TSP.

L'analyse des résultats des expérimentations montre que les utilisateurs sont plus performants pour évaluer « l'effort » nécessaire à un algorithme pour s'améliorer (tâche 3) et pour focaliser des courtes recherches (tâche 1) que pour identifier visuellement les meilleurs éléments (tâche 2).

4.3.3 Approches évolutionnaires interactives

Les méthodes évolutionnaires interactives (ou IEC pour *Interactive Evolutionary Computation*) font partie des méthodes d'optimisation où l'humain est impliqué fortement dans le processus d'optimisation car il effectue l'évaluation des solutions trouvées. Historiquement, l'IEC a été largement appliquée aux problèmes où la fonction objectif à optimiser n'est pas formulable mathématiquement, tels que qu'un problème de mélange de café [Takagi, 2001]. Tout comme les approches interactives d'optimisation multi-objectif, l'IEC monopolise l'humain et est confronté au problème de fatigue de celui-ci [Geiger and Wenger, 2007].

Les algorithmes génétiques ont été utilisés de manière interactive par Sushil and Rilun [1999] pour la résolution d'un TSP en 3 phases :

1. l'utilisateur partitionne tout d'abord visuellement le TSP en sous-problèmes,
2. ensuite l'algorithme génétique résout ces sous-problèmes séparément,
3. et enfin, l'utilisateur recombine visuellement les sous-tours trouvés en un seul tour global.

Par rapport à l'IEC, l'approche « *Human-Based Genetic Algorithm* » (HBGA) proposée par Kosorukoff [2001], étend davantage le rôle de l'humain dans son interaction avec un algorithme génétique. En effet, l'humain ne se contente plus d'évaluer les solutions uniquement, mais peut aussi jouer le rôle des opérateurs de croisement et de mutation comme le montre le tableau suivant :

	sélection	croisement/mutation
Algorithme génétique classique	Machine	Machine
IEC	Humain	Machine
HGBA	Humain	Humain

Conclusion

Nous avons dressé dans ce document un état de l'art du problème de routage de véhicules en termes de variantes et de techniques de résolution. Nous avons présenté une classification de ces techniques en fonction de leur nature et de leur fonctionnement.

Nous avons particulièrement développé la résolution du problème de routage de véhicules par les algorithmes génétiques en l'illustrant par des exemples précis.

Nous avons finalement abordé la problématique de la résolution interactive des problèmes de routage de véhicules en rappelant les bonnes performances humaines sur ces problèmes. Nous avons étudié à ce titre les principales approches dans la littérature tout en expliquant leurs réponses à la question principale en résolution interactive : « comment diviser le travail entre l'humain et la méthode d'optimisation utilisée ? ».

A Dates clés de l’histoire du VRP

Nous dressons dans cette annexe un aperçu historique du problème de routage de véhicules depuis sa formulation en 1959. Les dates-clés que nous avons retenues correspondent d’une part aux dates des publications significatives et d’autre part aux différentes méthodes de résolution. Elles sont résumées dans la figure 16.

Les publications que nous avons sélectionnées sont les suivantes :

- Les papiers :
 - les deux premiers papiers séminaux du VRP : la première formulation de Dantzig and Ramser [1959] et la première résolution efficace (heuristique gloutonne) de Clarke and Wright [1964],
 - les premières bibliographies annotées : la bibliographie complète de Laporte and Osman [1995], la bibliographie de Laporte [1997] spécifique aux papiers entre 1985 et 1996 et la bibliographie récente de Marinakis and Migdalas [2007].
- Les 5 livres dédiés au VRP :
 - le premier livre parlant du VRP [Eilon et al., 1971],
 - le premier livre édité entièrement consacré au VRP : *Vehicle Routing : Methods and Studies* de Golden and Assad [1988]. Il référence la plupart des premiers travaux sur plusieurs variantes,
 - le livre de Crainic and Laporte [1998],
 - le livre *The vehicle routing problem* de Toth and Vigo [2001a], une référence principale du domaine.
 - et le dernier livre paru, *The vehicle routing problem : latest advances and new challenges* de Golden et al. [2008]

La figure 16 met en évidence aussi l’évolution des méthodes de résolution du VRP : après les méthodes totalement manuelles jusqu’aux années 1965, les heuristiques ont connu un essor jusqu’aux années 1970 avec le développement des méthodes informatisées. La décennie des années 1980 a connu le plus d’avancées algorithmiques, notamment avec l’essor des métaheuristiques (algorithmes génétiques et recherche tabou).

En ce qui concerne les approches interactives de résolution, nous remarquons que mis à part quelques travaux anciens (Michie et al. [1968] et Krolak et al. [1970]) et un peu moins anciens (Waters [1984] et Fisher [1986]), ces approches sont de retour, notamment avec Anderson et al. [1999] (et l’optimisation évolutionnaire interactive (*Interactive Evolutionary Computation*, [Takagi, 2001]) pour d’autres problèmes)).

Des études récentes sur les performances humaines en résolution de TSP montrent l’importance que peuvent avoir les humains au cœur des systèmes d’optimisation et laissent penser qu’entre une résolution totalement manuelle et une résolution totalement automatisée, une interaction homme-machine serait plus bénéfique.

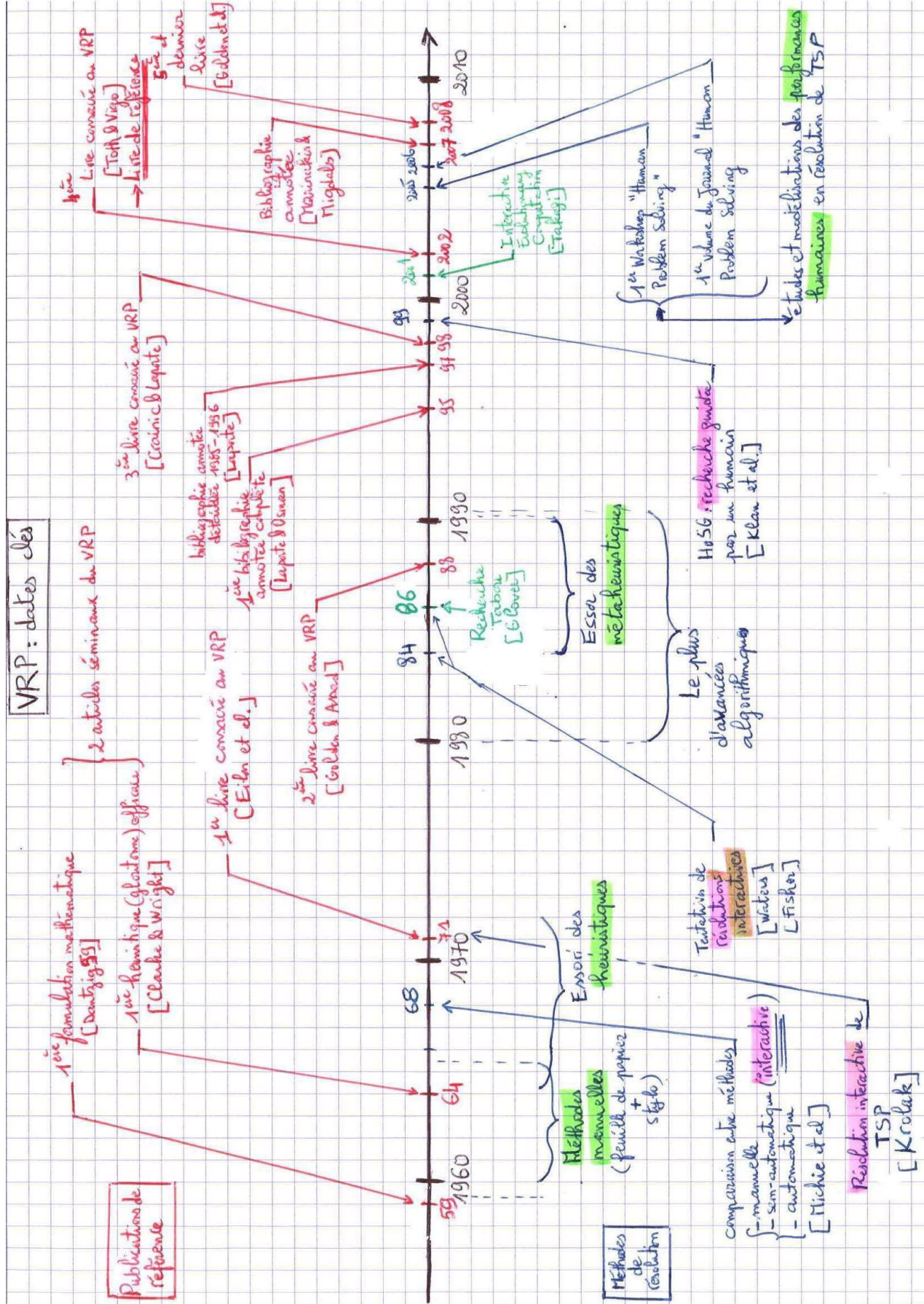


FIGURE 16 – Dates clés du VRP : publications de référence et méthodes de résolution.

Type	Nbre d'instances	Capacité des véhicules		Répartition spatiale des clients		durée du service	
		petite	large	clusterisée	aléatoire	longue	courte
C1	9	x		x			x
C2	8		x	x		x	
R1	12	x			x		x
R2	11		x		x	x	
RC1	8	x		x	x		x
RC2	8		x	x	x	x	

TABLE 2 – Description des 6 classes des problèmes de Solomon.

B Problèmes références (Benchmarks)

B.1 Instances Tests de Solomon pour le DCVRPTW

Les instances de Solomon [1987] sont les principaux problèmes de référence en termes de VRP avec fenêtres temporelles et sont donc un moyen très efficace pour comparer les méthodes de résolution [Bräysy and Gendreau, 2005a].

Ces problèmes, au nombre de 56, ont tous un dépôt central et 100 clients avec des coordonnées comprises dans $[0, 100]$. Ils imposent des contraintes de capacité, de distance totale maximale et de fenêtres temporelles. Les distances et les durées correspondent aux distances euclidiennes entre les points géométriques.

Les problèmes de Solomon sont répartis en six classes ayant chacune entre 8 et 12 instances. Les classes C1 et C2 ont des clients répartis en groupes (C pour *Cluster*) alors que les classes R1 et R2 ont des clients uniformément répartis. Les classes RC1 et RC2 ont des clients mélangés en termes de répartition géographique. Ces trois types de répartitions sont illustrés dans la figure 17 qui visualise les instances c101, r101 et rc101.

Les classes C1-R1-RC1 ont un horizon temporel court et des routes courtes (de 5 à 10 clients). Elles nécessitent entre 9 et 19 véhicules de petite capacité.

Les classes C2-R2-RC2 ont, elles, un horizon temporel long, des routes plus longues (certaines routes pouvant atteindre 30 clients) et nécessitent peu de véhicules (entre 2 et 4). Ce sont donc des tournées « long-courriers » relativement à la catégorie précédente.

Nous avons récapitulé toutes ces propriétés dans le tableau 2.

Comme le souligne Tam and Ma [2004], les problèmes de Solomon ont été tellement étudiés et « matraqués » par les chercheurs depuis des années que la

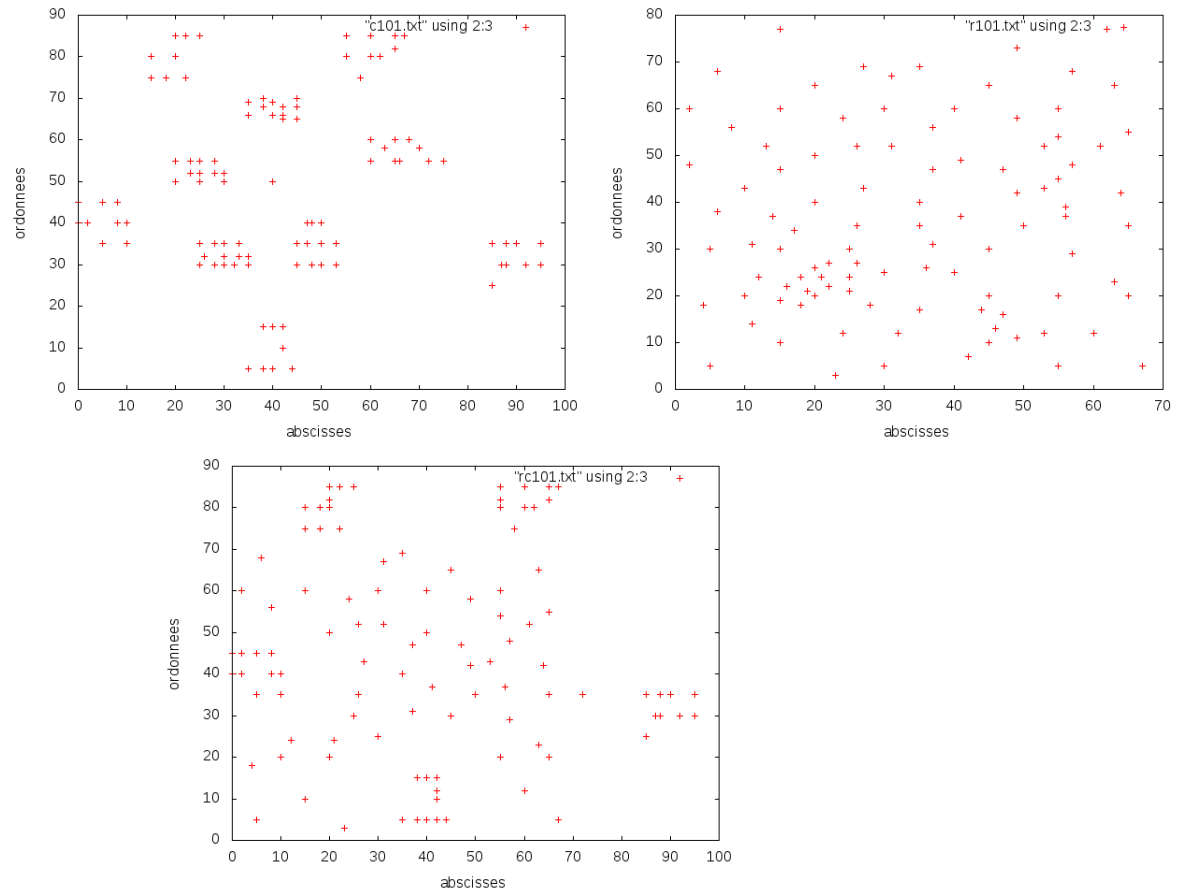


FIGURE 17 – Visualisation de trois instances (c101, r101 et rc101) des problèmes de Solomon.

moindre amélioration des solutions (ne serait ce que de l'ordre de 1% ou même de 0.1%) peut être considéré comme assez significative.

À titre indicatif, les instances de Solomon peuvent être trouvées sur le site « The VRP Web » (<http://neo.lcc.uma.es/radi-aeb/WebVRP/>) ou sur cette page <http://www.idsia.ch/~luca/macsvrptw/problems/welcome.htm>.

B.2 Autres problèmes références

Si les instances de Solomon sont la référence principale pour le VRPTW, les autres variantes du VRP ont aussi des benchmarks dédiés tels que les 14 problèmes de Christofides et al. [1979] pour le CVRP par exemple.

La prolifération des variantes du VRP et des problèmes de tests correspondants ont conduit Toth and Vigo [2001b] à établir un schéma de nomenclature des instances de 15 problèmes différents de CVRP proposés dans la littérature. Les noms donnés aux instances permettent de retrouver rapidement les caractéristiques principales de celles-ci : notamment le type du problème (symétrique/asymétrique, euclidien/non euclidien), le nombre de sommets qu'il contient, le nombre de véhicules, ainsi que sa référence bibliographique.

D'autres instances du CVRP peuvent être trouvées dans les librairies TSPLIB95¹³, une librairie de problèmes pour le TSP qui contenant aussi 16 instances de CVRP, ou VRPLIB¹⁴, une librairie spécifique au CVRP.

La librairie OR-Library¹⁵ est une librairie de plusieurs problèmes en recherche opérationnelle, y compris le problème de routage de véhicules dans la majorité de ses variantes.

Finalement, les deux sites web <http://www.branchandcut.org/> et « The VRP Web » (<http://neo.lcc.uma.es/radi-aeb/WebVRP/>) contiennent de nombreux jeux de données pour la plupart des variantes courantes du VRP.

13. consultable sur <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>

14. consultable sur http://www.or.deis.unibo.it/research_pages/ORinstances/VRPLIB/VRPLIB.html

15. consultable sur <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>

Références

- D. Anderson, E. Anderson, N. Lesh, J. Marks, K. Perlin, D. Ratajczak, and K. Ryall. Human-guided simple search : combining information visualization and heuristic search. In *Proceedings of the 1999 workshop on New Paradigms in Information Visualization and Manipulation (NPIVM '99)*, pages 21–25, New York, NY, USA, 1999. ACM. 42
- M.B. Baker and M.A. Ayechew. A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, 30(5) :787 – 800, 2003. 4, 7, 25, 26, 27, 28, 30, 33
- E. Bensana and G. Verfaillie. Optimisation combinatoire : quelques lecons de la théorie et de l’expérience. In *Ateliers du CST, Techniques et technologies des segments sols informatiques*, 1995. 14
- J. Berger, M. Salois, and R. Begin. A hybrid genetic algorithm for the vehicle routing problem with time windows. In Robert Mercer and Eric Neufeld, editors, *Advances in Artificial Intelligence*, volume 1418 of *Lecture Notes in Computer Science*, pages 114–127. Springer Berlin / Heidelberg, 1998. 30
- B. J. Best. *Modeling Human Performance on the Traveling Salesperson Problem : Empirical Studies and Computational Simulations*. PhD thesis, Department of Psychology, Carnegie Mellon University, Pittsburgh, PA, 2004. 33
- B.J. Best. A model of fast human performance on a computationally hard problem. In *Proceedings of the 27th annual conference of the cognitive science society*, pages 256–262, 2005. 34
- L. Bodin and B. Golden. Classification in vehicle routing and scheduling. *Networks*, 11(2) :97–108, 1981. 13
- Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Slowinski, editors. *Multiobjective Optimization : Interactive and Evolutionary Approaches*. Springer-Verlag, Berlin, Heidelberg, 2008. 36
- A. Van Breedmam. An analysis of the effect of Local Improvement Operators en Genetic Algorithms and Simulated Annealing for the Vehicle Routing Problem. 1996. 20
- O. Bräysy. *Local search and variable neighborhood search algorithms for the vehicle routing problem with time windows*. PhD thesis, University of Vaasa, Finland, 2001. 11
- O. Bräysy and M. Gendreau. Metaheuristics for the vehicle routing problem with time windows. *Report STF42 A*, 1025, 2001a. 16
- O. Bräysy and M. Gendreau. Genetic algorithms for the vehicle routing problem with time windows. Technical report, SINTEF Applied Mathematics, Department of Optimisation, Norway, 2001b. 32

- O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, Part I : Route construction and local search algorithms. *Transportation Science*, 39(1) :104–118, 2005a. 31, 44
- O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, Part II : Metaheuristics. *Transportation Science*, 39(1) :119–139, 2005b. 16, 31, 32
- N. Christofides, A. Mingozzi, and P. Toth. The vehicle routing problem. In *Combinatorial Optimization*, chapter 11. Wiley, 1979. 46
- E. Chronicle, J. MacGregor, and T. Ormerod. Optimizing and "pessimizing" : human performance with instructional variants of the traveling salesperson problem. *The Journal of Problem Solving*, 1(1) :7, 2006a. 34
- E.P. Chronicle, J.N. MacGregor, T.C. Ormerod, and A. Burr. It looks easy! Heuristics for combinatorial optimization problems. *Quarterly Journal of Experimental Psychology*, 59(4) :783, 2006b. 35
- G. Clarke and JW Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4) :568–581, 1964. 17, 42
- Y. Collette and P. Siarry. *Multiobjective optimization : principles and case studies*. Springer Verlag, 2003. 36
- T.G. Crainic and G. Laporte. *Fleet management and logistics*. Kluwer Academic Pub, 1998. 42
- T.G. Crainic and F. Semet. Recherche opérationnelle et transport de marchandises. In *Optimisation combinatoire. 3, Applications*. Hermès Science : Lavoisier, 2006. 7, 10, 17, 19, 20, 25, 26
- G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4) :pp. 393–410, 1954. 7
- G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1) :pp. 80–91, 1959. 42
- Thierry Defaix. Résolution des problèmes difficiles d'optimisation combinatoire, métaheuristiques. Notes de cours, 2010. URL <http://www.ensta.fr/~diam/ocro/>. 22
- C. Dhaenens, M.L. Espinouse, and Bernard Penz. Problèmes combinatoires classiques. In *Recherche opérationnelle et réseaux : méthodes d'analyse spatiale*. Hermès Science Publications, 2002. 6, 7, 17, 18, 20
- J.J. Dongarra. Performance of various computers using standard linear equations software. Technical Report CS-89-85, Department of Computer Science, University of Tennessee, U.S.A., 1998. 31

- S. Eilon, C.D.T. Watson-Gandy, and N. Christofides. *Distribution management*. Griffin London, 1971. 42
- M.L. Fisher. Interactive optimization. *Annals of Operations Research*, 5(1) : 541–556, 1986. 35, 38, 42
- B. Gacias. *Une approche interdisciplinaire pour l’ordonnancement des transports*. PhD thesis, LAAS - Laboratoire d’analyse et d’architecture des systèmes, 2010. 37
- M.J. Geiger and W. Wenger. On the interactive resolution of multi-objective vehicle routing problems. In *Evolutionary Multi-Criterion Optimization : 4th International Conference, EMO 2007*, volume 4403 of *LNCS*, pages 687–699. Springer Verlag, 2007. 40
- M. Gendreau and J.Y. Potvin. Metaheuristics in combinatorial optimization. *Annals of Operations Research*, 140 :189–213, 2005. 14
- M. Gendreau, A. Hertz, and G. Laporte. New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, 40(6) : 1086–1094, 1992. 16
- M. Gendreau, G. Laporte, and J.-Y. Potvin. Metaheuristics for the capacitated vrp. In *The vehicle routing problem*, chapter 6, pages 129–154. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001. 16
- M. Gendreau, G. Laporte, and J.-Y. Potvin. Métaheuristiques pour le problème des tournées de véhicules. In *Résolution de problèmes de RO par les métaheuristiques (Traité IC2, série Informatique et systèmes d’information)*. Hermès, 2003. 16, 17, 19, 20, 25
- M. Gendreau, J.Y. Potvin, O. Bräysy, G. Hasle, and A. Løkketangen. Metaheuristics for the vehicle routing problem and its extensions : A categorized bibliography. In *The Vehicle Routing Problem : Latest Advances and New Challenges*, volume 43, pages 143–169. Springer, 2008. 16
- D. Goldberg. *Genetic algorithms in search optimization and machine learning*. Addison Wesley, 1989. 22
- B. Golden and A. Assad. *Vehicle routing : methods and studies*, volume 39. Palgrave Macmillan, 1988. 42
- B.L. Golden, S. Raghavan, and E.A. Wasil. *The vehicle routing problem : latest advances and new challenges*. Springer, 2008. 42
- S.M. Graham, A. Joshi, and Z. Pizlo. The traveling salesman problem : a hierarchical model. *Memory & Cognition*, 28(7) :1191, 2000. 34
- M. Haj-Rachid. *Les problèmes de tournées de véhicules en planification industrielle : classification et comparaison d’opérateurs évolutionnaires*. PhD thesis, LIFC, Laboratoire Informatique de l’Université de Franche-Comté, 2010. 30

- M. Haj-Rachid, C. Bloch, W. Ramdane-Cherif, and P. Chatonnay. Différents opérateurs évolutionnaires de permutation : sélections, croisements et mutations. Research Report RR2010-07, LIFC - Laboratoire d'Informatique de l'Université de Franche Comté, July 2010. 25, 26, 30
- J.K. Hao, P. Galinier, and M. Habib. Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes. *Revue d'Intelligence Artificielle*, 13(2) :283–324, 1999. 4, 14, 16, 22
- J.H. Holland. Adaptation in natural and artificial systems. *Ann Arbor MI : University of Michigan Press*, 1975. 22
- H. Housroum. *Une approche génétique pour la résolution du problème VRPTW dynamique*. PhD thesis, Université d'Artois, 2005. 23, 31
- C.V. Jones. *Visualization and optimization*. Springer, 1996. 35
- N. Jozefowicz. *Modélisation et résolution approchées de problèmes de tournées multi-objectif*. PhD thesis, Laboratoire d'Informatique Fondamentale de Lille, Université des Sciences et Technologies de Lille, Villeneuve d'Ascq, France, December 2004. 11
- B. Kallehauge, J. Larsen, O.B.G. Madsen, and M.M. Solomon. Vehicle routing problem with time windows. *Column Generation*, pages 67–98, 2005. 11
- G.W. Klau, N. Lesh, J. Marks, and M. Mitzenmacher. Human-guided search. *Journal of Heuristics*, pages 1–22, 2009. 38
- A. Kosorukoff. Human based genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, SMC 2001*, 5 :3464–3469, 2001. 40
- P. Krolak, W. Felts, and G. Marble. A man-machine approach toward solving the traveling salesman problem. In *Proceedings of the 7th Design Automation Workshop (DAC '70)*, pages 250–264, New York, NY, USA, 1970. ACM. 4, 37, 39, 42
- P. Krolak, W. Felts, and J. Nelson. A Man-Machine Approach Toward Solving the Generalized Truck-Dispatching Problem. *TRANSPORTATION SCIENCE*, 6(2) :149–170, 1972. 37
- G. Laporte. The vehicle routing problem : An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3) :345 – 358, 1992. 7, 16, 31
- G. Laporte. Vehicle routing. In S. Martello (Eds.) M.D. Amico, F. Maffioli, editor, *Annotated Bibliographies in Combinatorial Optimization*, pages 223–240. Wiley, Chichester, 1997. 42
- G. Laporte and I.H. Osman. Routing problems : A bibliography. *Annals of Operations Research*, 61(1) :227–262, 1995. 6, 42

- E.L. Lawler, J.K. Lenstra, A.H.G.R. Kan, and D.B. Shmoys. *The traveling salesman problem : a guided tour of combinatorial optimization*, volume 3. Wiley New York, 1985. 6
- S. Lin. Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44 :22245–2269, 1965. 20
- S. Lin and B.W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2) :498–516, 1973. 38
- J.N. MacGregor and T. Ormerod. Human performance on the traveling salesman problem. *Perception and Psychophysics*, 58(4) :527–539, 1996. 33, 34
- J.N. MacGregor, E.P. Chronicle, and T.C. Ormerod. A Comparison of Heuristic and Human Performance on Open Versions of the Traveling Salesperson Problem. *Journal of Problem Solving*, 1(1) :33–43, 2006. 35
- Y. Marinakis and A. Migdalas. Annotated bibliography in vehicle routing. *Operational Research*, 7 :27–46, 2007. 42
- D. Michie, JG Fleming, and JV Oldfield. A comparison of heuristic, interactive, and unaided methods of solving a shortest-route problem. *Machine intelligence*, 3 :245–255, 1968. 33, 37, 42
- S Ormerod, Tand Slavin. Human solutions to the capacitated vehicle routing problem. 1st Workshop on Human Problem Solving, November 2005. Talk. 33, 35
- C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12) :1985–2002, 2004. 25, 26, 28, 30, 32, 33
- C. Rego and C. Roucairol. Le problème de tournées de vehicules : Étude et résolution approchée. Technical Report 2197, INRIA - Institut National de Recherche em Informatique et en Automatique, 1994. 5, 7, 10, 13, 17, 22, 31
- S.J. Russell, P. Norvig, J.F. Canny, J. Malik, and D.D. Edwards. *Artificial intelligence : a modern approach*. Prentice hall Englewood Cliffs, NJ, 1995. 14
- S.D. Scott, N. Lesh, and G.W. Klau. Investigating human-computer optimization. In *Proceedings of the SIGCHI conference on Human factors in computing systems : Changing our world, changing ourselves (CHI'02)*, pages 155–162, New York, NY, USA, 2002. ACM. 38
- C. Solnon. Résolution de problèmes combinatoires et optimisation par colonies de fourmis. Notes de cours, 2010. URL <http://www710.univ-lyon1.fr/~csolnon/>. 14, 16

- Marius M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35 :254–265, 1987. 31, 44
- J. Louis Sushil and Tang Rilun. Interactive genetic algorithms for the traveling salesman problem. In *Proceedings of the 1999 Genetic and Evolutionary Computation Conference (GECCO 1999)*, volume 1, pages 385–392, Orlando, Florida, USA, 1999. 30, 40
- H. Takagi. Interactive evolutionary computation : Fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, 89(9) : 1275–1296, September 2001. 40, 42
- E. Talbi. Métaheuristiques pour l’optimisation combinatoire multi-objectif : Etat de l’art. Technical report, Laboratoire d’Informatique Fondamentale de Lille, Université de Lille 1, France, 2001. 12
- El-Ghazali Talbi. *Metaheuristics : From Design to Implementation*. John Wiley and Sons, Hoboken, NJ, 2009. 16
- V. Tam and K.T. Ma. Combining meta-heuristics to effectively solve the vehicle routing problems with time windows. *Artificial Intelligence Review*, 21 :87–112, 2004. 4, 20, 21, 44
- P. Toth and D. Vigo. *The vehicle routing problem*. Society for Industrial Mathematics (SIAM) Monographs on Discrete Mathematics and Applications, Philadelphia, 2001a. 5, 7, 42
- P. Toth and D. Vigo. An overview of vehicle routing problems. In *The vehicle routing problem*, chapter 1, pages 1–26. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001b. 10, 46
- I. Van Rooij, U. Stege, and A. Schactman. Convex hull and tour crossings in the Euclidean traveling salesperson problem : Implications for human performance studies. *Memory & cognition*, 31(2) :215, 2003. 35
- C. D. J. Waters. Interactive vehicle routeing. *The Journal of the Operational Research Society*, 35(9) :821–826, 1984. 38, 42
- W. Wezel, J. Cegarra, and J-M. Hoc. Allocating functions to human and algorithm in scheduling. In Jan C. Fransoo, Toni Waefer, and John R. Wilson, editors, *Behavioral Operations in Planning and Scheduling*, pages 339–370. Springer Berlin Heidelberg, 2011. 4, 36, 37
- L.C. Yeun, W.R. Ismail, K. Omar, and M. Zirour. Vehicle routing problem : models and solutions. *Journal of Quality Measurement and Analysis JQMA*, 4(1) :205–218, 2008. 16
- G. Zäpfel, R. Braune, and M. Bögl. *Metaheuristic Search Concepts : A Tutorial with Applications to Production and Logistics*. Springer Verlag, 2010. 22

www.telecom-bretagne.eu

Campus de Brest

Technopôle Brest-Iroise
CS 83818
29238 Brest Cedex 3
France
Tél. : + 33 (0)2 29 00 11 11
Fax : + 33 (0)2 29 00 10 00

Campus de Rennes

2, rue de la Châtaigneraie
CS 17607
35576 Cesson Sévigné Cedex
France
Tél. : + 33 (0)2 99 12 70 00
Fax : + 33 (0)2 99 12 70 19

Campus de Toulouse

10, avenue Edouard Belin
BP 44004
31028 Toulouse Cedex 04
France
Tél. : +33 (0)5 61 33 83 65
Fax : +33 (0)5 61 33 83 75

© Télécom Bretagne, 2011

Imprimé à Télécom Bretagne
Dépôt légal : janvier 2011
ISSN : 1255-2275

