

Recherche à Voisinage Variable

Pierre Hansen

*GERAD et École des Hautes Études Commerciales
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada, H3T 2A7
pierreh@crt.umontreal.ca*

Nenad Mladenović

*GERAD et École des Hautes Études Commerciales
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada, H3T 2A7
nenad@crt.umontreal.ca*

Février 2000

Révisé: Février 2001

Les Cahiers du GERAD

G-2000-08

Copyright © 2001 GERAD

Résumé

La Recherche à Voisinage Variable (RVV) est une métaheuristique récente basée sur l'idée d'un chargement systématique de voisinage, à la fois dans une phase de descente et dans une phase d'exploration, qui permet de sortir d'un optimum local. On présente le schéma de base de cette méthode et plusieurs extensions ayant pour but d'accélérer la résolution ou de permettre de résoudre de grands problèmes. Deux types d'applications sont brièvement discutés : d'une part des problèmes d'optimisation combinatoire classique, et d'autre part des applications de type nouveau, stimulées par le cadre de la RVV et basées sur son usage : analyse du choix des mouvements dans l'heuristique 2-opt pour le problème du voyageur de commerce, stabilisation de la méthode de génération de colonnes et obtention de conjectures en théorie des graphes à l'aide du système AutoGraphiX.

Abstract

Variable Neighborhood Search is a recent metaheuristic based on the idea of systematic change of neighborhood during both a descent phase and an exploration phase, allowing to get out of local optima. A basic scheme and extensions allowing solution of large problems are presented. Applications to well-known combinatorial optimization problems, as well as to stabilization of column generation algorithms and computer-aided discovery of conjectures in graph theory are outlined.

1 Introduction

Les heuristiques sont un outil essentiel de l'optimisation appliquée, et pour de nombreux problèmes pratiques de grande taille le seul applicable. Leur but principal est de fournir, en un temps de calcul raisonnable, des solutions proches de l'optimum (ou optimales sans que ceci ne soit prouvé) à des problèmes d'optimisation avec ou sans contraintes. De plus, les heuristiques peuvent s'avérer très utiles à l'intérieur d'algorithmes exacts complexes, pour accélérer un grand nombre d'étapes. Enfin, elles sont un outil privilégié des systèmes de découverte de connaissances basés sur l'optimisation. Les problèmes d'optimisation sont le pain quotidien de la Recherche Opérationnelle et de ses applications dans de nombreux domaines. L'Intelligence Artificielle s'intéresse plus aux problèmes de satisfaction de contraintes. Le rôle des heuristiques est alors de déterminer une solution admissible. De fait, les deux types de problèmes diffèrent moins qu'il ne semble au premier abord ; en effet, les problèmes de satisfaction de contraintes peuvent être exprimés comme des problèmes d'optimisation de diverses manières, par exemple en minimisant une somme de variables artificielles associées aux violations des contraintes ; la réciproque est aussi souvent vraie : divers problèmes d'optimisation (notamment ceux où la fonction économique prend des valeurs discrètes) peuvent être ramenés à une séquence de problèmes de satisfaction de contraintes.

Les cadres généraux pour la construction d'heuristiques, désignés sous le vocable "méta-heuristique", ont fait l'objet de très nombreuses études au cours des deux dernières décennies. Les plus connues d'entre elles comprennent la méthode de descentes multiples (multistart) [BOE 94], la recherche de profondeur variable (variable depth search) [LIN 73, PAP 82], le recuit simulé (simulated annealing) [KIR 83], la recherche avec tabous (tabu search) [GLO 89, GLO 90, GLO 93, GLO 95, GLO 97, HAN 90], la recherche gourmande randomisée (GRASP) [FEO 95], la recherche à voisinage variable (variable neighborhood search) [HAN 98a, HAN 98b, MLA 97], sujet du présent chapitre, ainsi que quelques autres. Le lecteur pourra se référer à [REE 93] pour une synthèse due à plusieurs auteurs, [OSM 96] pour une bibliographie extensive et aux autres chapitres du présent ouvrage. Les hybrides ont aussi été longuement explorés.

Alors que les heuristiques traditionnelles, par exemple les méthodes simples de descente, sont bloquées dès le premier optimum local rencontré, il n'en va pas de même pour les heuristiques construites selon les paradigmes métaheuristiques. Tous prévoient en effet des méthodes, déterministes ou stochastiques, de sortie d'un optimum local de peu de valeur. Comme la valeur des optima locaux diffère souvent considérablement de celle de l'optimum global, particulièrement s'ils sont nombreux, l'impact pratique des métaheuristiques est immense. Par contraste, la théorie des métaheuristiques demeure fragmentaire. Alors que de bonnes heuristiques sont souvent obtenues avec un peu d'ingéniosité et beaucoup de réglage de paramètres, les raisons pour lesquelles elles fonctionnent aussi bien qu'elles le font demeurent largement inconnues. La situation est pire encore pour les hybrides. Il semble donc qu'une réflexion sur les propriétés désirables de métaheuristiques qui garantiraient leur intérêt théorique et pratique, pourrait être utile. Étant donné la très grande diversité

tant des heuristiques que des métaheuristiques, une axiomatisation semble hors de portée à l'heure actuelle.

Une liste possible de telles propriétés est la suivante :

- (1) *Simplicité* : la métaheuristique devrait être basée sur un principe simple et clair, largement applicable;
- (2) *Cohérence* : toutes les étapes d'heuristiques pour des problèmes particuliers devraient découler du principe de la métaheuristique;
- (3) *Efficience* : les heuristiques pour des problèmes particuliers devraient fournir des solutions optimales ou quasi-optimales. De préférence, elles devraient découvrir des solutions optimales pour la plupart des problèmes-tests d'ensembles standards quand ils sont disponibles;
- (4) *Efficacité* : les heuristiques pour des problèmes particuliers devraient prendre un temps de calcul modéré pour fournir des solutions optimales ou quasi-optimales;
- (5) *Robustesse* : la performance des heuristiques devrait se maintenir pour des instances diverses, c'est-à-dire que les heuristiques ne devraient pas être réglées en fonction d'un ensemble de problèmes d'entraînement et moins performantes ailleurs;
- (6) *Facilité d'usage* : les heuristiques devraient être bien définies, aisées à comprendre et surtout faciles à utiliser. Ceci suppose qu'elles aient aussi peu de paramètres que possible et idéalement aucun;
- (7) *Innovation* : de préférence le principe de la métaheuristique et/ou l'efficience et l'efficacité des heuristiques qui en découlent devraient mener à de nouveaux types d'applications.

La Recherche à Voisinage Variable (RVV) est une métaheuristique récente qui s'efforce d'avoir les qualités de la liste ci-dessus. Elle est basée sur un principe simple et relativement inexploré : le changement systématique de voisinage durant la recherche. (Notons que les règles précises gouvernant ces changements sont d'importance cruciale; plusieurs auteurs ont proposé à l'occasion de combiner différents types de mouvements, ou de voisinages, au sein de la même heuristique, sans toutefois que ceci ne soit fait systématiquement, ni que ce soit l'idée principale de leurs heuristiques.)

Le chapitre est organisé comme suit : le schéma RVV de base est présenté et discuté dans la prochaine section. Diverses extensions sont passées en revue dans la Section 3. La résolution de divers problèmes d'optimisation combinatoire à l'aide de RVV et la comparaison des résultats obtenus avec ceux d'autres heuristiques illustre ces deux sections. D'autres types d'application sont discutés dans la Section 4 : l'analyse de mouvements d'une heuristique RVV basée sur l'heuristique 2-opt [CRO 58] pour le problème du voyageur de commerce, la stabilisation des méthodes de génération de colonnes et le système AutoGraphiX de découverte de conjectures en théorie des graphes. De brèves conclusions sont tirées dans la Section 5.

-
- Initialisation.* Sélectionner une structure de voisinage $\mathcal{N}$, qui sera utilisée dans la recherche; trouver une solution initiale x ;
- Répéter* les étapes suivantes jusqu'à ce que la condition d'arrêt (c'est-à-dire l'obtention d'un optimum local) soit satisfaite :
- (a) Trouver le meilleur voisin $x' \in \mathcal{N}(x)$ de x ;
 - (b) Si x' n'est pas meilleur que x , fin. Sinon poser $x = x'$ et retourner en (a);
-

FIG. 1 – Étapes d'une heuristique à recherche locale

2 RVV de base

Un schéma de base pour la *descente simple* ou *recherche locale* (*local search*) est présenté sur la Figure 1. Comme discuté plus haut, cette heuristique s'arrête aussitôt qu'un optimum local est atteint. Elle utilise un seul type de mouvement, ou, ce qui est équivalent, une seule structure de voisinage

Une première amélioration de ce schéma est la *descente multiple* (*multistart*) : la descente simple est réitérée de nombreuses fois à partir de solutions initiales tirées au hasard et la meilleure solution obtenue est conservée. Toutefois, si les optima locaux sont nombreux (c'est-à-dire si leur nombre dépasse largement le nombre de recherches locales qui peuvent être faites en un temps raisonnable) même la valeur de cette meilleure solution peut être fort éloignée de celle de la solution globalement optimale (un phénomène connu sous le nom de catastrophe de Tchebycheff [BOE 94]).

Il est clair que la descente multiple disperse ses efforts en explorant de nombreuses vallées sans se concentrer sur les régions les plus prometteuses. Il semble que c'est tout ce qui peut être fait si les optima locaux et leurs valeurs sont distribués aléatoirement dans tout l'espace des solutions. Cependant, ce n'est souvent pas le cas. En effet, on a fréquemment observé que les optima locaux tendent à s'agréger dans une ou un petit nombre de petites régions de cet espace des solutions (ainsi qu'on peut le vérifier lorsque tous les optima locaux peuvent être obtenus à l'aide des conditions du premier ordre, comme par exemple pour la programmation quadratique en variables 0-1 sans contraintes). Il est donc important, après avoir découvert un optimum local, d'exploiter l'information ainsi obtenue, c'est-à-dire le fait que de nombreuses variables auront la même valeur qu'à l'optimum global (mais sans qu'on ne sache lesquelles). Donc la descente multiple peut être modifiée pour explorer les voisinages proches, puis de plus en plus éloignés, de la meilleure solution connue de manière probabiliste. C'est précisément ce qui est fait par la RVV, lorsqu'elle utilise des structures de voisinages les plus simples, c'est-à-dire des voisinages imbriqués. Nous présentons maintenant une formulation générale du schéma de base de la RVV puis discutons ce cas particulier.

Notons par $\mathcal{N}_k (k = 1, 2, \dots, k_{\max})$ un ensemble fini de structures de voisinages pré-sélectionnées et par $\mathcal{N}_k(x)$ l'ensemble de solutions du $k^{\text{ème}}$ voisinage de x . Les étapes du schéma de base de la RVV sont présentées sur la Figure 2.

Initialisation : Sélectionner l'ensemble de structures de voisinages $\mathcal{N}_k (k = 1, 2, \dots, k_{\max})$ qui sera utilisé dans la recherche; déterminer (ou choisir) une solution initiale x ; choisir une condition d'arrêt;

Répéter les étapes suivantes jusqu'à ce que la condition d'arrêt soit satisfaite :

- (1) Poser $k = 1$;
- (2) Répéter les étapes ci-dessous jusqu'à ce que $k = k_{\max}$:
 - (a) *Perturbation*. Générer un point x' au hasard dans le $k^{\text{ème}}$ voisinage de $x (x' \in \mathcal{N}_k(x))$;
 - (b) *Recherche locale*. Appliquer une méthode de recherche locale avec x' comme solution initiale; soit x'' l'optimum local ainsi obtenu;
 - (c) *Déplacement ou continuation*. Si l'optimum local est meilleur que la meilleure solution connue, recentrer la recherche en ce point ($x \leftarrow x''$) et retourner au premier voisinage $\mathcal{N}_1 (k \leftarrow 1)$; sinon poser $k \leftarrow k + 1$;

FIG. 2 – Schéma de base de la RVV

On peut choisir comme condition d'arrêt, parmi d'autres, le temps CPU maximum alloué, le nombre maximum d'itérations ou le nombre maximum d'itérations depuis la dernière amélioration de la meilleure valeur connue.

Un moyen aisé d'obtenir un ensemble de structures de voisinage est de considérer un mouvement élémentaire et de définir $\mathcal{N}_k(x)$ comme l'ensemble de solutions obtenu en appliquant k tels mouvements à x (sans mouvements inverses). Par exemple si x est un vecteur Booléen le mouvement élémentaire peut être la complémentation d'une composante. Dans ce cas, les voisinages $\mathcal{N}_k(x)$ sont les ensembles de vecteurs à distance de Hamming k de x . Un autre type de mouvement est souvent utilisé dans les heuristiques pour le problème du voyageur de commerce et d'autres problèmes de routage de véhicules proches : désigné sous le nom de 2-opt [CRO 58], il consiste à retirer deux arêtes de la solution courante et à reconnecter leurs sommets de la seule manière qui conserve une solution connexe.

Le schéma de la Figure 2 a recours à une méthode de recherche locale. N'importe laquelle peut être, en principe, utilisée et les modifications nécessaires pour transformer une heuristique de recherche locale en RVV sont souvent minimales. Toutefois, on peut faire beaucoup mieux en utilisant plusieurs voisinages dans cette phase de descente également. En effet, un optimum local pour un voisinage n'en est pas nécessairement un pour un autre. De plus, l'exploration de voisinages peut être rapide ou lente selon la simplicité ou la complexité des mouvements qui les définissent. Ceci suggère un schéma de base pour la *Descente à Voisinage Variable* (DVV), présenté sur la Figure 3.

Une heuristique DVV peut aisément être construite pour un problème donné en assemblant les heuristiques proposées dans la littérature (ou, plus précisément, en considérant l'union de leurs étapes). L'ordre dans lequel elles seront utilisées est de grande importance et une bonne règle empirique est de les ranger par ordre de taille non-décroissante des voisinages qu'elles utilisent. On peut aller plus loin dans cette direction : les voisinages

Initialisation : Sélectionner l'ensemble de structures de voisinages $\mathcal{N}'_k (k = 1, \dots, k'_{\max})$ qui seront utilisées dans la descente; déterminer une solution initiale x (si elle n'est pas déjà fournie);

Répéter les étapes suivantes jusqu'à ce qu'il n'y ait plus d'améliorations :

- (1) Poser $k \leftarrow 1$;
- (2) Répéter les étapes ci-dessous jusqu'à ce que $k = k_{\max}$:
 - (a) *Exploration de voisinage*. Déterminer le meilleur voisin x' de $x (x' \in \mathcal{N}'_k(x))$;
 - (b) *Déplacement ou non*. Si la solution x' ainsi obtenue est meilleure que la solution x précédente, poser $(x \leftarrow x')$; sinon poser $k \leftarrow k + 1$;

FIG. 3 – Schéma de base de la DVV

utilisés par une heuristique peuvent être décomposés en un (ou des) ensemble(s) de voisinages imbriqués de taille croissante. Par exemple lorsqu'on utilise des mouvements 2-opt pour le problème du voyageur de commerce, on peut considérer les ℓ arêtes les plus courtes incidentes à chaque sommet, puis les 2ℓ plus courtes, etc. pour un ℓ donné [HAN 00c]. De plus, si certains voisinages sont beaucoup plus aisés à explorer que d'autres, il peut être profitable de retourner à l'étape (1), c'est-à-dire au premier voisinage, aussitôt qu'une meilleure solution est obtenue plutôt que de poursuivre l'exploration des voisinages en séquence.

Exemple 1 *Problème du Voyageur de Commerce (PVC).*

Étant donné n villes et les distances entre toutes les paires d'entre elles, le PVC consiste à déterminer un tour de ces villes de longueur minimum (ou, en d'autres termes, une permutation des villes qui minimise la somme de n distances entre paires de villes successives). Ce problème NP-complet classique est sans doute le plus étudié en optimisation combinatoire. Une revue et comparaison détaillée des heuristiques pour le PVC est donnée en [JOH 97]. Une des premières heuristiques pour le PVC, encore étudiée à l'heure actuelle, est 2-opt, décrite plus haut. Le plongement de 2-opt dans le schéma de base de la RVV (RVV-1) donne des améliorations de performance notables, tout en réduisant le temps de calcul, comme le montre le Tableau 1. L'utilisation du voisinage 2-opt complet (RVV-2) donne une réduction de longueur de 2,73% en moyenne, alors que négliger de 40 à 80% des arêtes les plus longues donne une amélioration de 4,43% en moyenne. À titre de comparaison une amélioration de 0,5% obtenue en utilisant l'heuristique 2.5-opt [BEN 92] plutôt que l'heuristique 2-opt, moyennant un accroissement du temps de calcul de 30 à 40% est signalée dans [JOH 97]. (Il est bien connu que les solutions du PVC obtenues à l'aide de l'heuristique 2-opt seule tendent à avoir des erreurs croissant rapidement avec la taille des problèmes. Une application complète de la RVV, incorporant les heuristiques 1-opt, 2-opt, 2.5-opt, celle de Lin et Kernighan [LIN 73] et d'autres est actuellement à l'étude [HAN 00c]).

	Meilleure solution trouvée			% d'amélioration		Temps CPU		
n	2-OPT	RVV-1	RVV-2	RVV-1	RVV-2	2-OPT	RVV-1	RVV-2
100	825.69	817.55	811.95	0.99	1.66	0.25	0.18	0.17
200	1156.98	1143.19	1132.63	1.19	2.10	3.88	3.21	2.82
300	1409.24	1398.16	1376.76	0.79	2.30	12.12	10.29	9.35
400	1623.60	1602.59	1577.42	1.29	2.84	46.13	40.03	34.37
500	1812.08	1794.59	1756.26	0.96	3.07	110.64	99.57	91.00
600	1991.56	1959.76	1925.51	0.97	3.32	204.60	191.85	173.07
700	2134.86	2120.59	2089.33	0.67	2.13	347.77	307.93	259.06
800	2279.18	2242.11	2190.83	1.63	3.88	539.94	480.50	462.23
900	2547.43	2399.52	2342.01	5.81	8.06	699.33	656.96	624.74
1000	2918.10	2555.56	2483.95	12.42	14.88	891.61	844.88	792.88
Av.	1869.87	1803.36	1768.67	2.73	4.43	285.63	263.54	244.97

TAB. 1 – PVC : résultats moyens pour des problèmes aléatoires Euclidiens; 100 essais pour $n = 100, \dots, 500$ et 10 essais pour $n = 600, \dots, 1000$. Temps de calcul en secondes CPU sur un SUN SPARC 10 à 135,5 Mips (comme les autres résultats de ce chapitre).

n	p	Meilleure solution trouvée	% ERREUR					TEMPS CPU			
			FI	RVV	HC	CSTS	TS	FI	RVV	CSTS	TS
200	10	48912.00	1.21	0.00	0.00	0.02	0.68	5.2	80.3	59.1	381.9
	15	31153.00	1.29	0.00	0.00	0.00	2.80	7.5	121.2	118.9	401.1
	20	23323.00	1.96	0.00	0.65	0.00	0.74	9.6	161.6	163.2	416.6
300	10	82664.00	3.58	0.00	0.00	1.08	0.47	12.3	248.2	179.2	1241.0
	15	52685.00	4.47	0.00	0.17	0.50	1.98	19.4	373.3	311.8	1321.6
	20	38244.00	3.34	0.00	1.35	0.72	2.49	24.4	475.8	467.0	1378.3
400	10	123464.00	0.19	0.00	0.00	0.12	3.79	24.5	463.4	361.3	2910.6
	15	79872.00	5.20	0.00	0.75	2.50	5.15	32.0	631.5	463.1	3096.8
	20	58459.00	7.08	0.46	0.00	0.92	1.17	45.4	958.6	653.3	3218.3
500	10	150112.00	2.03	0.00	0.00	0.14	1.52	40.6	864.9	474.3	9732.2
	15	97624.00	3.54	0.00	0.00	1.55	0.79	54.7	1164.2	887.2	9731.1
	20	72856.00	4.86	0.00	0.41	1.46	0.41	74.5	1593.4	1350.0	9748.4
Moyenne			3.23	0.04	0.27	0.75	1.83	29.2	594.7	457.4	3631.5

TAB. 2 – PM : résultats pour les problèmes tests de [ROL 96], temps maximum alloué pour CSTS et RVV égal à 30 fois le temps de FI; les meilleures solutions obtenues pour 50 essais de FI, CSTS et RVV sont présentées. Les temps de calcul de HC ne sont pas disponibles.

Exemple 2 *Problème de la p -médiane (PM).*

Étant donné un ensemble L de m localisations potentielles pour p facilités et un ensemble U de localisations données pour n usagers, le problème de la p -Médiane (PM) est de localiser simultanément les p facilités de sorte à minimiser la somme des distances des usagers aux facilités (ou le coût correspondant), chaque usager étant desservi par la facilité la plus proche. Les solutions de PM sont donc caractérisées par des vecteurs 0-1 x avec p composantes parmi $|L|$ égales à 1 indiquant où les facilités sont localisées. Des structures de voisinage sont alors définies en déplaçant $1, 2, \dots, k_{\max}$ facilités et correspondent aux ensembles de vecteur 0-1 à distance de Hamming $2, 4, \dots, 2k_{\max}$ de x . L'heuristique de descente utilisée est 1-échange, avec le schéma de calcul efficient d'échange rapide de [WHI 83] ("Fast interchange" ou FI).

Les résultats d'une comparaison d'heuristiques pour des problèmes PM difficiles, avec des données tirées au hasard (ne satisfaisant pas l'inégalité triangulaire) sont présentés dans le Tableau 2.

RVV est comparée avec

- a) la recherche locale avec échange rapide (FI) ;
- b) un algorithme de Recherche avec Tabous sophistiqué (TS) [ROL 96], qui comprend l'oscillation stratégique [GLO 89]–[GLO 95] ;
- c) la méthode récente de concentration heuristique (HC) [ROS 97, ROS 98] ;
- d) la Recherche avec Tabous et substitution en chaîne (CSTS) [MLA 96].

Il est clair que RVV et HC sont plus performants que les deux heuristiques TS. La meilleure solution connue est atteinte 11 fois par RVV, avec une erreur moyenne de 0,04%, et 7 fois par HC avec une erreur moyenne de 0,27%. FI est plus rapide que les autres heuristiques, mais donne des solutions beaucoup moins bonnes.

Exemple 3 *Classification de moindre somme des carrés (CMSC).*

Étant donné n points de l'espace Euclidien à q dimensions, le problème de la classification selon le critère de la moindre somme des carrés des erreurs (CMSC) consiste à partitionner cet ensemble en classes de sorte que la somme des carrés des distances des points aux centroïdes de leurs classes soit minimum.

Parmi de nombreuses heuristiques pour CMSC, la recherche locale K -MEANS [JAN66, MAC67] est la plus populaire. C'est une heuristique d'échange où les points sont réaffectés à une autre classe que la leur, un à la fois, jusqu'à l'obtention d'un optimum local. Une autre heuristique populaire, appelée H -MEANS [AND73] sélectionne une partition initiale, calcule les centroïdes de ses classes puis réaffecte les points au centroïde le plus proche et itère cette procédure jusqu'à la stabilité. Une nouvelle heuristique, appelée J -MEANS, est proposée en [HAN 01a] ; les centroïdes sont relocalisés en certains des points, qui ne coïncident pas encore avec l'un d'entre eux. Les résultats d'une comparaison de K -MEANS, H -MEANS, $H + K$ -MEANS (où H -MEANS et K -MEANS sont utilisés en séquence) et deux versions de RVV sont donnés dans le Tableau 3 ; RVV-2, qui étend J -MEANS s'avère de loin la meilleure heuristique.

m	Meilleure solution trouvée	% d'erreur par rapport à la meilleure solution trouvée									
		K-MEANS		H-MEANS		H+K-MEANS		RVV-1		RVV-2	
		Moy.	Meil.	Moy.	Meil.	Moy.	Meil.	Moy.	Meil.	Moy.	Meil.
10	1754840264.9	0.19	0.19	0.01	0.00	0.01	0.00	0.14	0.00	0.14	0.04
20	791925963.7	2.89	0.00	1.87	1.29	1.31	0.46	3.50	1.84	0.74	0.01
30	482302357.1	9.34	6.68	11.77	9.01	8.20	5.79	10.64	5.16	0.86	0.00
40	342844809.0	15.50	12.04	20.01	15.28	16.86	11.53	15.95	9.64	0.81	0.00
50	256892529.0	27.16	24.57	35.60	30.59	28.66	17.14	29.95	13.50	1.42	0.00
60	199151542.6	35.79	32.53	44.59	33.56	36.21	30.00	35.19	20.50	0.59	0.00
70	159781533.1	44.28	33.08	56.63	47.90	47.39	38.89	43.85	25.59	0.78	0.00
80	130038918.6	53.15	46.64	62.34	50.77	56.69	43.98	51.08	35.07	0.75	0.00
90	111322621.7	56.12	48.94	63.94	51.38	54.40	48.38	44.94	28.17	0.73	0.00
100	97352045.7	60.41	54.74	46.21	46.21	35.95	35.95	42.99	28.00	1.16	0.00
110	86287804.2	60.69	52.78	59.92	49.73	41.44	40.79	43.97	23.76	1.34	0.00
120	76380389.5	62.90	54.00	62.32	52.66	48.96	41.28	38.58	33.56	1.02	0.00
130	68417681.6	65.91	50.73	54.66	38.95	42.34	24.64	38.46	26.30	0.67	0.00
140	61727504.5	62.16	49.82	53.05	45.51	36.00	36.00	30.85	22.04	1.43	0.00
150	56679822.6	66.06	55.05	47.82	40.74	33.43	26.88	25.41	20.05	1.34	0.00
160	52210995.2	59.37	53.16	41.74	34.88	30.85	25.61	25.83	19.32	0.70	0.00
Erreur moyenne		42.62	35.93	41.40	34.28	32.42	26.71	30.08	19.53	0.90	0.00

TAB. 3 – CMSC : résultats pour un problème avec 1060 points; moyenne et meilleurs résultats en 10 essais; règle d'arrêt : 150 secondes par essai.

3 Extensions

L'étude de la distribution des optima locaux pour une série de problèmes combinatoires montre qu'ils sont souvent concentrés dans les parties inférieures de quelques larges vallées. En d'autres termes, il y a quelques régions importantes proches d'être convexes, avec du bruit. Le schéma de base de la RVV est efficace pour trouver le meilleur optimum local au sein d'une telle région (la première explorée) mais moins bon dans la découverte des autres régions, qui peuvent contenir de meilleurs optima locaux. En effet, l'information résidant dans la meilleure solution connue étant d'importance décroissante au fur et à mesure que l'on s'éloigne d'elle, le schéma de base RVV dégénère en la recherche locale multiple.

Une première tentative pour résoudre cette difficulté [HAN 00b] a consisté à choisir des points bas dans les voisinages de x . Une manière de le faire est de partir de x vers l'un parmi m (un paramètre) de ses voisins avec la plus petite valeur, ce qui donne un point de $\mathcal{N}_1(x)$ puis d'itérer pour obtenir des points de $\mathcal{N}_2(x), \mathcal{N}_3(x), \dots$. Ceci rappelle la métaheuristique GRASP [FEO 95] (et chapitre 7 de ce livre) dans laquelle toutefois une telle randomisation a lieu au début, au sein d'un schéma de descente multiple, et non à la fin, après qu'un premier optimum local (ou un optimum meilleur que le précédent) ait été trouvé. L'heuristique RVV à *voisinage bas* ainsi définie a été testée sur des instances du problème de la satisfiabilité maximum pondérée, discuté ci-dessous. En moyenne, la performance ne fut pas meilleure, et parfois pire, que celle du schéma de base.

Une seconde modification s'adresse plus directement au problème d'atteindre des vallées éloignées de la meilleure solution connue. Dans le schéma de base, ou le schéma à voisinage

Initialisation : Sélectionner l'ensemble de structures de voisinages $\mathcal{N}_k (k = 1, \dots, k_{\max})$, qui seront utilisées dans la recherche ; déterminer une solution initiale x et sa valeur $f(x)$; poser $x_{\text{opt}} \leftarrow x, f_{\text{opt}} \leftarrow f(x)$; choisir une condition d'arrêt, une valeur pour le paramètre α et une mesure ρ de la distance entre vecteurs ;

Répéter les étapes suivantes jusqu'à ce que la condition d'arrêt soit satisfaite :

- (1) Poser $k \leftarrow 1$;
 - (2) Répéter les étapes suivantes jusqu'à ce que $k = k_{\max}$:
 - (a) *Perturbation*. Générer un point x' au hasard dans le $k^{\text{ème}}$ voisinage de $x (x' \in \mathcal{N}_k(x))$;
 - (b) *Recherche locale*. Appliquer une méthode de recherche locale avec x' comme solution initiale ; noter par x'' l'optimum local ainsi obtenu ;
 - (c) *Amélioration ou non*. Si $f(x'') < f_{\text{opt}}$ poser $f_{\text{opt}} \leftarrow f(x)$ et $x_{\text{opt}} \leftarrow x''$;
 - (d) *Recentrage de la recherche ou non*. Si $f(x'') - \alpha \rho(x, x') < f(x)$ poser $x \leftarrow x''$ et $k \leftarrow 1$; sinon poser $k \leftarrow k + 1$.
-

FIG. 4 – Étapes de la RVV biaisée.

bas, un point x' est tiré dans $\mathcal{N}_k(x)$ et une recherche locale mène à un point x'' . Une correction proportionnelle à la distance entre x et x'' est soustraite de la valeur $f(x'')$ dans le test pour la relocalisation de la recherche (qui est alors distinct du test sur l'amélioration de la meilleure solution connue). Le schéma RVV biaisé (RVVB) qui en résulte est présenté sur la Figure 4. Les deux modifications peuvent être combinées (RVVB-bas).

Exemple 4 *Problème de la satisfiabilité maximum pondérée (WMAXSAT).*

Le problème de la satisfiabilité (SAT), sous forme clausale, consiste à déterminer si un ensemble donné de m clauses (toutes sous forme disjonctive ou toutes sous forme conjonctive) construites sur n variables logiques a une solution ou pas [GAR 78]. Le problème de la satisfiabilité maximum (MAXSAT) [HAN 90] consiste à trouver une solution satisfaisant le nombre maximum de clauses. Dans le problème de la satisfiabilité maximum pondéré (WMAXSAT, [RES 97]) des poids positifs sont assignés aux clauses et il faut trouver une solution qui maximise la somme des poids des clauses satisfaites.

Les résultats d'expériences avec des instances ayant 500 variables, 4500 clauses et 3 variables par clause, sous forme directe ou complémentée, sont présentés dans le Tableau 4.

	RVV	RVV-bas	RVVB-bas	TS
Nombre d'instances où la meilleure solution est obtenue	6	4	23	5
% moyen d'erreur pour 10 essais	0.2390	0.2702	0.0404	0.0630
% d'erreur minimum pour 10 essais	0.0969	0.1077	0.0001	0.0457
Nombre d'instances	25	25	25	25

TAB. 4 – Résultats pour les instances GERAD de WMAXSAT ($n = 500$).

Il est clair que RVVB-bas donne de meilleurs résultats que RVV ou une heuristique TS.

Lorsque la taille des instances du problème considéré s'accroît le temps requis par RVV peut devenir élevé, voire même prohibitif. De plus, trouver des solutions proches de l'optimum (ou optimales, mais sans preuve de ce qu'elles le sont) devient plus difficile. Cependant, avec l'arrivée des magasins de données ("data warehouses") la demande pour la résolution de très grandes instances est importante et croissante. À nouveau, deux manières d'affronter ce problème ont été considérées. La première vise à réduire le temps de calcul du schéma de base, tout en continuant d'obtenir des solutions acceptables. Un changement élémentaire, mais radical, du schéma de base permet d'atteindre cet objectif : la suppression complète de la phase de descente locale (qui est d'ordinaire celle qui requiert le plus de temps de calcul). Le schéma résultant porte le nom de Recherche à Voisinage Variable Réduite (RVVR, [HAN 01b]). Il est surprenant de constater que RVVR donne de bons résultats en des temps de calcul très faibles.

Exemple 5 *Problème de la p -Médiane (à nouveau).*

Les résultats obtenus par RVVR pour de grandes instances de PM (obtenues à partir des données de l'exemple à 3038 usagers de la base de donnée ORLIB, pour diverses valeurs de p) sont présentés dans le Tableau 5. On voit que RVVR donne des solutions avec la même erreur moyenne que FI, en 18 fois moins de temps de calcul.

Une seconde manière vise à exploiter plus avant la structure des instances considérées. Pour ce faire RVV est combiné avec la décomposition. Plus précisément, on construit un schéma RVV à deux niveaux en plongeant le schéma de base dans une procédure d'approximations successives. Dans ce cas, tous les attributs (ou variables), sauf k d'entre eux, d'une solution x donnée sont fixés dans la phase de recherche locale. Toutes ces fixations définissent un voisinage $\mathcal{N}_k(x)$. Comme dans RVV une solution x' est tirée au hasard dans $\mathcal{N}_1(x)$. Puis la recherche locale est appliquée à x' mais seulement sur la variable libre. Une nouvelle valeur pour cette variable est donc calculée ainsi que la valeur correspondante de la fonction objectif. Les autres étapes sont semblables à celles de RVV : si la nouvelle solution n'est pas meilleure que la meilleure solution connue, on pose $k \leftarrow k + 1$, c'est-à-dire qu'on cherche des améliorations dans l'espace où toutes les variables sauf 2 sont fixées, etc. Sinon un recentrage de la recherche autour de la nouvelle solution a lieu et k est remis à 1. Les étapes de la Recherche et Décomposition à Voisinage Variable (RDVV) ainsi définies sont décrites sur la Figure 5.

Notons que RDVV diffère de RVV seulement dans les étapes 2a et 2b. La recherche locale dans l'étape 2b peut se faire à l'aide du schéma RVV de base. Ceci donne un schéma RVV à deux niveaux.

Exemple 6 *Problème de la p -Médiane (suite).*

Le Tableau 5 présente les résultats de RDVV pour les problèmes tests décrits ci-dessus. Il est clair que RDVV donne de bien meilleurs résultats que FI en des temps de calcul semblables ; RDVV donne aussi de meilleurs résultats que RVV dans tous les cas, en 5 fois moins de temps. Les valeurs négatives dans la dernière colonne indiquent que toutes les meilleures solutions connues ont été améliorées (l'erreur des autres heuristiques que RDVV

Initialisation : Sélectionner l'ensemble de structures de voisinages $\mathcal{N}_k (k = 1, \dots, k_{\max})$ qui seront utilisées dans la recherche; déterminer une solution initiale x ; choisir une condition d'arrêt;

Répéter les étapes suivantes jusqu'à ce que la condition d'arrêt soit satisfaite :

- (1) Poser $k \leftarrow 1$;
 - (2) Répéter les étapes suivantes jusqu'à ce que $k = k_{\max}$:
 - (a) *Perturbation*. Générer un point x' au hasard dans le $k^{\text{ème}}$ voisinage de $x (x' \in \mathcal{N}_k(x))$; en d'autres termes, soit y un ensemble de k attributs (ou composantes) présents dans x' mais non dans $x (y = x' \setminus x)$.
 - (b) *Recherche locale*. Trouver le meilleur optimum local dans l'espace y soit par inspection soit par une heuristique; soit y' la meilleure solution ainsi obtenue et x'' la solution correspondante dans l'espace complet des solutions ($x'' = (x' \setminus y) \cup y'$);
 - (c) *Recentrage de la recherche ou non*. Si la solution ainsi obtenue est meilleure que la meilleure solution connue, recentrer la recherche autour d'elle ($x \leftarrow x''$) et poursuivre la recherche dans le voisinage $\mathcal{N}_1 (k = 1)$; sinon poser $k \leftarrow k + 1$.
-

FIG. 5 – Étapes du schéma de base de la RDVV.

par rapport aux nouvelles meilleures solutions est approximativement la somme des erreurs de leurs colonnes respectives et de l'opposé des valeurs de la dernière colonne).

4 Nouveaux types d'applications

Comme mentionné dans l'introduction, une propriété désirable des métaheuristiques est l'innovation, c'est-à-dire le fait que la nouvelle manière de considérer les problèmes qu'elles expriment mène à de nouveaux types d'application. Nous présentons ci-dessous trois cas où la RVV a mené à de tels résultats.

Un premier exemple est la découverte d'une propriété surprenante dans la sélection de mouvements des heuristiques. Comme mentionné précédemment de nombreuses heuristiques ont été appliquées au problème du voyageur de commerce, dont plusieurs de type RVV. L'étude la plus extensive de ces dernières faite à ce jour porte sur l'heuristique 2-opt, également décrite plus haut. Cette étude a mené à la découverte d'un phénomène imprévu [HAN 99b] : le choix à chaque itération de l'échange d'arêtes donnant la meilleure amélioration (ou réduction de valeur de la fonction objectif) donne en moyenne des résultats moins bons que le choix du premier échange donnant une amélioration. Ceci s'observe tant dans le cas où la première amélioration est définie en considérant les arêtes le long du tour à partir d'une origine fixe, que dans le cas où les arêtes à insérer incidentes à chaque sommet sont triées par ordre de longueurs croissantes. Une explication, corroborée par une analyse expérimentale plus poussée, est aisément obtenue dans le second cas : si la meilleure amélioration est choisie, deux arêtes courtes, mais pas nécessairement très courtes, sont introduites dans le tour et seront difficiles à éliminer par la suite; si la première amélioration est choisie, une arête très courte et une arête de longueur moyenne sont introduites dans

p	MEILLEURE SOLUTION TROUVÉE				TEMPS CPU			% D'ERREUR		
	RVV	FI	RVVR	RVVB	FI	RVVR	RDVV	FI	RVVR	RDVV
50	507809.5	510330.2	510216.4	507655.2	612.9	60.7	311.1	0.50	0.47	-0.03
100	354488.7	356005.1	356666.3	353255.2	1040.7	132.4	885.8	0.43	0.61	-0.35
150	281911.9	284159.0	283024.6	281772.1	1459.2	128.5	1432.4	0.80	0.39	-0.05
200	239086.4	240646.2	241355.6	238623.0	1943.6	107.6	1796.8	0.65	0.95	-0.19
250	209718.0	210612.9	210727.7	209343.3	2395.6	150.3	2189.7	0.43	0.48	-0.18
300	188142.3	189467.5	188709.3	187807.1	2583.4	130.6	1471.7	0.70	0.30	-0.18
350	171726.8	172668.5	172388.5	171009.3	2804.3	153.1	2270.3	0.55	0.39	-0.42
400	157910.1	158549.5	158805.0	157079.7	4083.3	158.7	3670.9	0.40	0.57	-0.53
450	146087.8	146727.2	147062.0	145449.0	4223.8	179.5	1652.7	0.44	0.67	-0.44
500	136081.7	136680.5	136665.0	135468.0	4649.3	209.7	2599.8	0.44	0.43	-0.45
<i>Average</i>					2579.6	141.1	1828.12	0.53	0.53	-0.28

TAB. 5 – Resultats pour PM avec 3038 usagers ; Méthodes : RVV - Schéma de base ; FI - Échange rapide ; RVVR - Recherche rapide ; RVVB - Recherche biaisée.

le tour et cette dernière pourra aisément être éliminée à une étape ultérieure. Une étude systématique de phénomènes semblables dans une série d'heuristiques pourrait mener à de nombreuses améliorations.

Un second exemple a rapport à l'usage d'heuristiques au sein d'algorithmes exacts. Il est bien connu que de nombreux problèmes d'optimisation combinatoire tels que le problème de la p -Médiane, le problème de Weber Multisource (WM), les problèmes de classification et de partitionnement suivant divers objectifs [HAN 97, DUM 00], le problème des rotations d'équipages [DES 95] et bien d'autres peuvent être résolus en combinant la génération de colonnes et la programmation en nombres entiers. Comme la solution optimale de la relaxation linéaire est souvent massivement dégénérée la convergence peut être lente et même si la solution optimale a été trouvée, de nombreuses itérations peuvent encore être requises pour le prouver. Le dual de la procédure de génération de colonnes est la méthode d'approximation extérieure de Kelley [KEL 60]. La convergence lente provient de la génération de nombreux hyperplans loin du cône pointé en la solution optimale. Il serait donc désirable d'estimer quelle est cette solution duale optimale et de concentrer la procédure de recherche en pénalisant la génération d'hyperplans qui en sont éloignés.

Considérons un problème primal P et son dual D , résolus par génération de colonnes :

$$\begin{array}{llll}
 (P) \quad \text{s.c.} & \min c^T x & (D) \quad \text{s.c.} & \max b^T \pi \\
 & Ax = b & & A^T \pi \leq c \\
 & x \geq 0 & & (\pi)
 \end{array}$$

Un schéma de stabilisation qui demeure entièrement dans le cadre de la programmation linéaire est proposé en [DUM 99]. Il fusionne une méthode de perturbation et une méthode exacte de pénalisation.

Un problème primal $\tilde{P}$ et son dual $\tilde{D}$ sont définis comme suit :

$$\begin{array}{ll}
 \min c^T \tilde{x} + \delta_- y_- + \delta_+ y_+ & \max b^T \tilde{\pi} - \varepsilon_- w_- - \varepsilon_+ w_+ \\
 (\tilde{P}) \quad \text{s.c.} \quad A\tilde{x} - y_- + y_+ = b & (\tilde{D}) \quad \text{s.c.} \quad A^T \tilde{\pi} \leq c \\
 & y_- \leq \varepsilon_- \\
 & y_+ \leq \varepsilon_+ \\
 & \tilde{x}, y_-, y_+ \geq 0 \\
 & -\tilde{\pi} - w_- \leq \delta_- \\
 & \tilde{\pi} - w_+ \leq \delta_+ \\
 & (\tilde{\pi}), w_-, w_+ \geq 0
 \end{array}$$

Dans le primal y_- et y_+ sont des vecteurs de variables d'écart et de surplus avec des bornes respectives de ε_- et ε_+ . Ces variables sont pénalisées dans la fonction objectif par des vecteurs δ_- et δ_+ . Lors de l'application de la génération de colonnes, lorsqu'il ne reste plus de colonne avec un coût réduit négatif, les paramètres δ et ε sont mis à jour selon des règles spécifiques au problème considéré.

Pour que cette méthode de *génération de colonnes stabilisée* fonctionne bien, de bonnes estimations des variables duales à l'optimum doivent être obtenues avant la résolution. Plusieurs manières d'obtenir des intervalles de valeurs heuristiques pour ces variables duales sont esquissées en [DUM 99]. De plus, la RVV peut être utilisée pour résoudre heuristiquement le sous-problème de génération d'une colonne de coût réduit du bon signe (à condition de vérifier qu'il n'en existe plus avec un algorithme exact lorsqu'elle n'en trouve plus). Ce problème a une forme spécifique à chaque application, par exemple problème polynomial pour PM, programme hyperbolique en variables 0-1 pour CMSC, problème d'optimisation globale continue pour WM. Pour le problème PM ceci a mené à la résolution exacte d'instances avec 3038 usagers contre 900 dans la littérature ; pour le problème correspondant dans le plan, soit WM, la méthode a permis la résolution exacte d'instances avec 1060 usagers, contre 30 dans la littérature [KRA 97, HAN 00a]. Pour le problème de classification CMSC, la combinaison de la génération de colonnes avec la méthode de points intérieurs ACCPM, la programmation hyperbolique et quadratique en variables 0-1 et l'heuristique RVV a permis la résolution exacte d'instances avec jusqu'à 150 entités, y compris du célèbre problème des iris de Fisher [DUM 00, FIS 36].

Un troisième exemple d'innovation a trait au domaine de la découverte scientifique assistée par ordinateur. Un système pour l'étude de graphes extrémaux basé sur RVV et appelé AutoGraphiX (AGX) a été récemment développé ([CAP 99a]–[CAP 99e] et [CVE 01]). Il comprend une composante RVV utilisant une série de transformations graphiques simples : *suppression* ou *addition* d'une arête, *déplacement* d'une arête, *détour*, c'est-à-dire suppression d'une arête et addition de deux arêtes joignant ses extrémités à un sommet qui ne leur était pas adjacent, *raccourci*, soit l'inverse de la transformation précédente, mouvement *2-opt* comme pour le PVC, *insertion* ou *suppression* d'un sommet, etc. La structure de voisinages de la RVV elle-même est définie par la distance de Hamming entre graphes (ou, en d'autres termes, par une succession de déplacements d'arêtes).

AGX aborde les problèmes suivants : (a) trouver un graphe satisfaisant des contraintes données ; (b) trouver des valeurs optimales ou quasi-optimales pour un invariant sous des contraintes (avec éventuellement une paramétrisation sur le nombre de sommets et/ou d'arêtes) ; (c) réfuter une conjecture ; (d) trouver ou renforcer une conjecture ; (e) suggérer une preuve.

Jusqu'à présent AGX a trouvé plus de 50 conjectures, dont une douzaine ont été démontrées et une réfutée. Certaines sont beaucoup plus simples que les résultats proches trouvés auparavant. Par exemple, une étude à l'aide d'AGX de l'énergie E d'un graphe chimique [CAP 99a] a mené aux formules nouvelles

$$E \geq 2\sqrt{m}$$

et

$$E \geq \frac{4m}{n}$$

où n et m désignent les nombres de sommets et d'arêtes d'un graphe; ces deux formules ont été prouvées.

Un autre résultat surprenant a été obtenu lors d'une étude du polynôme de Hosoya (ou des distances) d'un arbre, défini par

$$H(T) = a_0 + a_1x + a_2x^2 + \cdots + a_dx^d$$

où $a_0 = n$, $a_1 = m$, a_k est le nombre de paires de sommets de l'arbre T à distance k et d est le diamètre de l'arbre. Gutman [GUT 92, GUT 93] a conjecturé qu'il n'y avait pas d'arbre tel que $H(T)$ soit palindromique, c'est-à-dire que $a_0 = a_d$, $a_1 = a_{d-1}$ et ainsi de suite. AGX a trouvé un contre-exemple avec 21 sommets et plusieurs autres légèrement plus grands [CAP 99b]. Le cas où le diamètre est impair est plus difficile que celui où il est pair, qui laisse un coefficient non contraint au centre. Considérons alors la distance à la condition de palindrome définie par

$$dp(T) = \sum_{i=0}^{\lfloor d/2 \rfloor} |a_{d-i} - a_i|.$$

Pour tout n jusqu'à 50, il a été trouvé que

$$dp(T) \geq \lceil \frac{n}{2} \rceil$$

et pour $n \geq 10$, la borne est toujours atteinte. D'autres résultats obtenus par AGX pour les graphes chimiques et pour les graphes généraux sont décrits dans [CVE 01, CAP 99c, CAP 99d]; ce dernier article contient également un algorithme en $O(p^3)$ pour trouver une base de relations affines entre p invariants à partir d'un échantillon de graphes extrémaux. Son application à l'ensemble d'arbres d'index (ou plus grande valeur propre de la matrice d'adjacence) minimum obtenu à l'aide d'AGX a conduit à la relation (ou conjecture) imprévue

$$2\alpha - m - n_1 + 2r - d = 0$$

où α désigne le nombre de stabilité, n_1 le nombre de sommets pendants et r le rayon de l'arbre considéré. Il est peu probable qu'une telle relation aurait été trouvée sans l'aide de l'ordinateur étant donné son grand nombre d'invariants.

5 Conclusions

En passant en revue les sept qualités des métaheuristiques énumérées dans l'introduction, on voit que RVV les possède dans une large mesure. En effet, son principe est simple et toutes les étapes du schéma de base comme de ses extensions en découlent naturellement. La RVV s'est montrée efficace dans la résolution de batteries de problèmes tests pour plusieurs classes de problèmes, donnant des solutions optimales ou quasi-optimales et ce en des temps de calcul modérés (ou du moins raisonnables). De plus, ses performances sont robustes, son principe de base est aisé à appliquer, et les heuristiques résultantes d'un usage très facile, le nombre de paramètres étant très modéré et parfois zéro. La simplicité de la RVV est sans doute sa principale qualité. En effet, la RVV donne des résultats aussi bons ou meilleurs que ceux de la plupart des autres métaheuristiques pour de nombreux problèmes et d'une manière bien plus simple. Ceci explique son potentiel pour l'innovation, déjà illustré par plusieurs types d'études ou d'applications originales; analyse et sélection de mouvements dans l'heuristique 2-opt pour le PVC, génération de colonnes stabilisée et système AGX de découverte assistée par ordinateur en théorie des graphes. Tout ceci suggère que la RVV pourrait jouer un rôle important dans la conception de nouvelles heuristiques performantes pour une multitude de problèmes et mener aussi à une meilleure compréhension des raisons de leur succès.

Références

- [AND73] ANDERBERG, M.R. *Cluster Analysis for Application*. Academic Press, New York, 1973.
- [BEN 92] BENTLEY, J.L. "Fast Algorithms for Geometric Traveling Salesman Problem". *ORSA J. Comput.*, 4 :387–411, 1992.
- [BOE 94] BOESE, K.D., KAHNG, A.B. ET MUDDU, S. "A New Adaptive Multi-Start Technique for Combinatorial Global Optimizations". *Oper. Res. Lett.*, 16 :101–113, 1994.
- [CAP 99a] CAPOROSI, G., CVETKOVIĆ, D., GUTMAN, I. ET HANSEN, P. "Variable Neighborhood Search for Extremal Graphs. 2. Finding Graphs with Extremal Energy". *Journal of Chemical Information and Computer Sciences*, 39 :984–996, 1999.
- [CAP 99b] CAPOROSI, G., DOBRYNIN, A.A., GUTMAN, I. ET HANSEN, P. "Trees with Palindromic Hosoya Polynomials". *Graph Theory Notes of New York*, XXXVII :10–16, 1999.
- [CAP 99c] CAPOROSI, G., GUTMAN, I. ET HANSEN, P. "Variable Neighborhood Search for Extremal Graphs. 4. Chemical Trees with Extremal Connectivity Index". *Computers & Chemistry*, 23 :469–477, 1999.
- [CAP 99d] CAPOROSI, G. ET HANSEN, P. "Finding Relations in Polynomial Time". *Proceedings of the XVIth International Joint Conference on Artificial Intelligence*, 2 :780–785, Stockholm, 1999.
- [CAP 99e] CAPOROSI, G. ET HANSEN, P. "Variable Neighborhood Search for Extremal Graphs. 1. The AutoGraphiX System". *Discrete Mathematics*, 212 :29–44, 2000.

- [CRO 58] CROES, G.A. “A Method for Solving Traveling Salesman Problems”. *Operations Research*, 6 :791–812, 1958.
- [CVE 01] CVETKOVIĆ, D., SIMIC, S., CAPOROSSI, G. ET HANSEN, P. “Variable Neighborhood Search for Extremal Graphs. 3. On the Largest Eigenvalue of Color-Constrained Trees”. *Linear and Multilinear Algebra* (à paraître), 2001.
- [DES 95] DESROSIERS, J., DUMAS, Y., SOLOMON, M.M. ET SOUMIS, F. “Time Constrained Routing and Scheduling”. M.O. Ball *et al.* (eds.), *Network Routing, Handbooks in Operations Research and Management Science*, 8 :35–139, Elsevier Science, B.V., 1995.
- [DUM 99] DU MERLE, O., VILLENEUVE, D., DESROSIERS, J. ET HANSEN, P. “Stabilized Column Generation”. *Discrete Mathematics*, 194 :229–237, 1999.
- [DUM 00] DU MERLE, O., HANSEN, P., JAUMARD, B. ET MLADENOVIC, N. “An Interior Point Algorithm for Minimum Sum of Squares Clustering”. *SIAM Journal on Scientific Computing*, 21(4) :1485–1501, 2000.
- [FEO 95] FEO, T. ET RESENDE, M. “Greedy Randomized Adaptive Search”. *J. Global Optim.*, 6 :109–133, 1995.
- [FIS 36] FISHER, R.A. “The Use of Multiple Measurements in Taxonomic Problems”. *Ann. Eugenics*, VII, part II :179–188, 1936.
- [GAR 78] GAREY, M.R. ET JOHNSON, D.S. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. Freeman, New-York, 1978.
- [GLO 89] GLOVER, F. “Tabu Search – Part I”. *ORSA J. Comput.*, 1 :190–206, 1989.
- [GLO 90] GLOVER, F. “Tabu Search – Part II”. *ORSA J. Comput.*, 2 :4–32, 1990.
- [GLO 95] GLOVER, F. “Tabu Thresholding : Improved Search by Nonmonotonic Trajectories”. *ORSA J. Comput.*, 7 :426–442, 1995.
- [GLO 93] GLOVER, F. ET LAGUNA, M. “Tabu Search”. C. Reeves, editor, *Modern Heuristic Techniques for Combinatorial Optimization*, pp. 70–150, Blackwell, Oxford, 1993.
- [GLO 97] GLOVER, F. ET LAGUNA, M. *Tabu Search*. Kluwer, Boston, 1997.
- [GUT 92] GUTMAN, I. “Independent Vertex Palindromic Graphs”. *Graph Theory Notes of New York*, XXIII :3, New York Academy of Sciences, 21–24, 1992.
- [GUT 93] GUTMAN, I. “A Contribution to the Study of Palindromic Graphs”. *Graph Theory Notes of New York*, XXIV :6, New York Academy of Sciences, 51–56, 1993.
- [GUT 99] GUTMAN, I., MILJKOVIĆ, O., CAPOROSSI, G. ET HANSEN, P. “Alkanes with Small and Large Randić Connectivity Indices”. *Chemical Physics Letters*, 306 :366–372, 1999.
- [HAN 90] HANSEN, P. ET JAUMARD, B. “Algorithms for the Maximum Satisfiability Problem”. *Computing*, 44 :279–303, 1990.
- [HAN 97] HANSEN, P. ET JAUMARD, B. “Cluster Analysis and Mathematical Programming”. *Mathematical Programming*, 79 :191–215, 1997.
- [HAN 00a] HANSEN, P., JAUMARD, B., KRAU, S. ET DU MERLE, O. “A Stabilized Column Generation Algorithm for the Multisource Weber Problem”, (en préparation).

- [HAN 00b] HANSEN, P., JAUMARD, B., MLADENOVIC, N. ET PARREIRA, A. "Variable Neighborhood Search for Weighted Maximum Satisfiability". *Les Cahiers du GERAD* G-2000-62, École des HEC, Canada, 2000.
- [HAN 98a] HANSEN, P. ET MLADENOVIC, N. "An Introduction to Variable Neighbourhood Search". S. Voss *et al.* (eds.), *Metaheuristics, Advances and Trends in Local Search Paradigms for Optimization*, pp. 433–458, Kluwer, Dordrecht, 1999.
- [HAN 98b] HANSEN, P. ET MLADENOVIC, N. "Variable Neighborhood Search : Principles and Applications". *European Journal of Operational Research*, 2001.
- [HAN 99a] HANSEN, P. ET MLADENOVIC, N. "Variable Neighborhood Search for the p -Median". *Location Science*, 1999.
- [HAN 99b] HANSEN, P. ET MLADENOVIC, N. "First Improvement may be Better than Best Improvement : An Empirical Study". *Les Cahiers du GERAD* G-99-40, École des HEC, Canada, 1999.
- [HAN 01a] HANSEN, P. ET MLADENOVIC, N. "J-MEANS, A New Local Search Heuristic for Minimum Sum-of-Squares Clustering". *Pattern Recognition*, 34 :405–413, 2001.
- [HAN 00c] HANSEN, P., MLADENOVIC, N., LABBÉ, M. ET GENDREAU, M. "Variable Neighborhood Search for the Traveling Salesman Problem" (en préparation).
- [HAN 01b] HANSEN, P., MLADENOVIC, N. ET PEREZ-BRITTO, D. "Variable Neighborhood Decomposition Search". *Journal of Heuristics* (à paraître), 2001.
- [JAN66] JANCEY, R.C. "Multidimensional Group Analysis". *Australian J. Botany*, 14 :127–130, 1966.
- [JOH 97] JOHNSON, D.S. ET MCGEOCH, L.A. "The Traveling Salesman Problem : a Case Study in Local Optimization". E.H.L. Aarts and J.K. Lenstra (eds.), *Local Search in Combinatorial Optimization*, pp. 215–310, Wiley, London, 1997.
- [KEL 60] KELLEY, J.E. "The Cutting-Plane Method for Solving Convex Programs". *J. SIAM*, 8 :703–712, 1960.
- [KIR 83] KIRKPATRICK, S., GELATT, C.D. ET VECCHI, M. "Optimization by Simulated Annealing. *Science*, 220 :671–680, 1983.
- [KRA 97] KRAU, S. *Extensions du problème de Weber*, Thèse de doctorat, École Polytechnique de Montréal (sous la direction de P. Hansen et B. Jaumard), 1997.
- [LIN 73] LIN, S. ET KERNIGHAN, B.W. "An Effective Heuristic Algorithm for the Traveling Salesman Problem". *Oper. Res.*, 21 :498–516, 1973.
- [MAC67] MACQUEEN, J.B. "Some Methods for Classification and Analysis of Multivariate Observations". *Proc. of the 5th Berkeley Symp. on Mathem. Statist. and Probab.*, 1 :281–297, 1967.
- [MLA 97] MLADENOVIC, N., ET HANSEN, P. "Variable Neighborhood Search". *Computers and Operations Research* 24 :1097–1100, 1997.
- [MLA 96] MLADENOVIC, N., MORENO, J.P. ET MORENO-VEGA, J. "A Chain-Interchange Heuristic Method". *Yugoslav J. Oper. Res.*, 6(1) :41–54, 1996.
- [OSM 96] OSMAN, I.H. ET LAPORTE, G. "Metaheuristics : A bibliography". *Ann. Oper. Res.*, 63 :513–628. 1996.

- [PAP 82] PAPADIMITRIOU, C.H. ET STEIGLITZ, K. *Combinatorial Optimization, Algorithms and Complexity*. Prentice Hall, New Jersey, 1982.
- [REE 93] REEVES, C. (eds.). *Modern Heuristic Techniques for Combinatorial Problems*. Blackwell, Oxford, 1993.
- [RES 97] RESENDE, M.G.C., PITSOULIS, L.S. ET PARDALOS, P.M. “Approximate Solution of Weighted MAX-SAT Problems Using GRASP”. Dingzhu Du, Jun Gu et Panos M. Pardalos (eds.), *Satisfiability Problem : Theory and Applications*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 35, American Mathematical Society, Providence, Rhode Island, 1997.
- [ROL 96] ROLLAND, E., SCHILLING, D.A. ET CURRENT, J.R. “An Efficient Tabu Search Procedure for the p -median Problem”. *European Journal of Operational Research*, 96 :329-342, 1996.
- [ROS 97] ROSING, K.E. ET REVELLE, C.S. “Heuristic Concentration : Two Stage Solution Construction”. *European Journal of Operational Research*, 97 :75-86, 1997.
- [ROS 98] ROSING, K.E., REVELLE, C.S., ROLLAND, E., SCHILLING, D.A. ET CURRENT, J.R. “Heuristic Concentration and Tabu Search : A Head to Head Comparison”. *European Journal of Operational Research*, 104 :93-99, 1998.
- [WHI 83] WHITAKER, R. “A Fast Algorithm for the Greedy Interchange for Large-Scale Clustering and Median Location Problems”. *INFOR*, 21 :95-108, 1983.