

IDW : un algorithme de recherche locale combinant intensification et diversification

B. Neveu¹, G. Trombettoni¹ et F. Glover²

¹ projet COPRIN INRIA-I3S-CERTIS, route des lucioles, 06902 Sophia-Antipolis
`{neveu,trombe}@sophia.inria.fr`

² Leeds School of Business, University of Colorado, Boulder USA
`Fred.Glover@colorado.edu`

Résumé Cet article présente l'algorithme de recherche locale IDW pour l'optimisation combinatoire. C'est une métaheuristique simple qui a deux paramètres à régler, le nombre de voisins à examiner à chaque pas et le voisin sur lequel remonter en cas de blocage dans un minimum local.

Mots-Clefs. Métaheuristique ; Recherche locale ; Optimisation combinatoire.

1 Introduction

Le principe de la recherche locale est d'essayer de trouver la solution d'un problème d'optimisation combinatoire en effectuant une suite de mouvements élémentaires qui améliorent la configuration courante. L'ensemble des mouvements possibles à partir de la configuration courante est appelé voisinage. La méthode la plus simple (méthode de descente) se trouve alors bloquée quand elle arrive sur un minimum local du paysage de recherche.

Il existe déjà de nombreuses métaheuristicues pour sortir des minimums locaux : citons par exemple le recuit simulé, qui accepte avec une probabilité décroissant au cours de la marche une configuration moins bonne que la courante, la recherche avec liste taboue qui garde une mémoire partielle des configurations déjà rencontrées en gérant une liste de mouvements interdits, dits tabous : on prend le meilleur mouvement non-tabou qui peut être un mouvement détériorant permettant ainsi de sortir d'un minimum local.

Nous avons défini une nouvelle métaheuristique appelée IDW[1] (pour marche avec intensification et diversification). Cette métaheuristique a donné de premiers résultats très prometteurs sur différents problèmes comme le coloriage de graphe, l'affectation de fréquences, l'ordonnancement de voitures ...

2 La métaheuristique IDW

Cette métaheuristique s'inscrit dans le cadre de la recherche locale pour un problème d'optimisation combinatoire. Une recherche locale est définie par une fonction à minimiser, un voisinage (quels sont les voisins d'une configuration) et un algorithme qui indique comment choisir le prochain voisin.

IDW s'inscrit dans le cadre des métaheuristicues avec liste de candidats (*candidate list*) [2]. Elle permet à un algorithme de descente de sortir d'un minimum local en considérant un nombre maximum $MaxNeighbors(MN)$ de voisins à examiner. Si un de ces MN voisins améliore ou a la même valeur que la configuration courante, il est choisi, sinon un deuxième paramètre appelé *SpareNeighbor* (SN) indique sur quel voisin remonter. On choisit le moins mauvais voisin parmi les SN premiers examinés. Le premier paramètre indique l'effort d'intensification : combien de voisins regarde-t-on pour essayer de descendre et le second paramètre la diversification : jusqu'où remonte-t-on en terme de coût ?

Dans [1], nous avons fixé SN à deux valeurs possibles : $SN=1$ (un voisin quelconque) et $SN=MN$ (le meilleur voisin examiné). Nous avons remarqué que ce paramètre était déterminant pour la réussite de l'algorithme : en particulier, pour les problèmes d'affectation de fréquences du CELAR où le paysage est très chahuté, il est intéressant de se permettre de remonter assez haut ($SN=1$) alors que les problèmes d'ordonnancement de voitures avec des mouvements de permutation, il faut au contraire remonter le moins haut possible.

3 Réglage et expérimentations

Nous avons donc maintenant un algorithme à deux paramètres, qui se règlent assez facilement automatiquement. Le réglage choisit le paramétrage donnant la meilleure moyenne de la fonction à optimiser dans le même temps de calcul. Nous alternons phases de réglage (sur un nombre de mouvements réduit) et phases d'exécution de l'algorithme à paramètres fixés. Nous augmentons la longueur de la marche jusqu'à ce que tous les essais (10 par exemple) trouvent une solution (quand la valeur de l'optimum est connue) ou dépassent un temps limite.

On retrouve bien les réglages pressentis : SN faible pour les problèmes du CELAR (AF) et SN fort pour les problèmes d'ordonnancements de voitures (OV) de la CSPLib. On peut aussi trouver pour certains problèmes de coloriage de graphe (CG) du challenge DIMACS des valeurs intermédiaires pour SN .

Nous avons comparé l'algorithme IDW à deux paramètres avec IDW avec les deux valeurs extrêmes du paramètre SN . Nous obtenons en général de meilleurs résultats à l'exécution et le temps total (réglage plus exécution) est plus faible que la somme des temps totaux des essais avec les deux valeurs extrêmes.

Problème	IDWa			IDWb			IDW			
	MN	val.	temps	MN	val.	temps	MN	SN	val.	temps
CG : le450_15	46	0	74 (3510)	17	0	101 (7671)	28	3	0	7(4672)
CG : le450_25c	151	2.3	773 (13777)	70	2	670 (11669)	75	27	2	641 (14824)
AF : CELAR06	100	3473	1057 (16902)	15	3859	1552 (24824)	75	2	3544	971 (18183)
AF : CELAR07	93	369836	1291 (18108)	7	1.05 e+6	1647(24908)	93	1	369836	1294 (23600)
OV : pb10-93	1234	4	1014 (17008)	375	3	71 (1662)	450	252	3	94 (7256)
OV : pb16-81	729	0	407 (7547)	168	0	20 (1917)	200	37	0	20 (1170)

TAB. 1. Résultats. Pour chaque algorithme, on donne le paramétrage (MN) ou (MN,SN) et les résultats : valeur moyenne de la fonction d'évaluation sur 10 essais ; temps moyen d'un essai, en secondes (temps total des 10 essais réglage compris).

Références

1. B. Neveu, G. Trombettoni, F. Glover : IDWalk : a candidate list strategy with a simple identification device, CP 2004, Toronto, Canada.
2. F. Glover et M. Laguna : Tabu Search. Kluwer Academic Publishers, 1997.