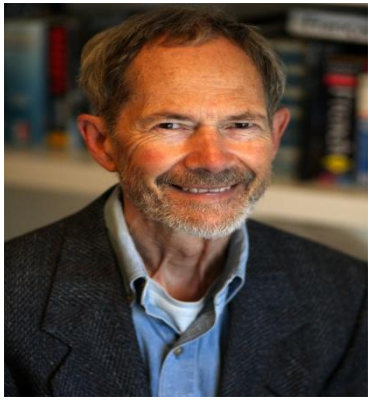
Algorithme de recherche taboue

Fred W. Glover
Optec Systems, Inc.
1919 Seventh Street
Boulder, Colorado 80302
Office Phone: (303) 447-3255 X106
Fax: (303) 447-3886
glover@OptTek.com
http://www.OptTek.com



Présentation

- La méthode de recherche tabou (historique et principe)
- Principe de base
  - Voisinage
  - Gestion liste tabou
  - Critère d'aspiration
- Adaptation de la méthode tabou à un problème d'optimisation à variables continues
  - Principe
  - Notions de "voisinage" et de "pas" de mouvement
  - Subdivision de l'espace des solutions
  - Partitionnement par arbre binaire
  - Mouvement par modification d'une seule composante
  - Notion de voisinage par topologie de boules
- La méthode de tabou avancée
  - Intensification
  - Diversification

Historique et principes

Fred Glover en 1986 et indépendamment Hansen (1986)

Amélioration de la méthode de la descente

Méthode inspirée de l'intelligence artificielle (mémoire)

La méthode utilise une mémoire (ou plusieurs mémoires) qui sont mises à jour et exploitées au cours de la recherche.

Algorithme tabou de base : mémoire à court terme (liste taboue)

Algorithme tabou évolué : mémoire à court terme (liste tabou) + mémoire à long terme pour assurer l'intensification et/ou la diversification

Elle a été appliquée avec succès pour résoudre de nombreux problèmes difficiles d'optimisation combinatoire :

- problèmes de routage de véhicule,
- problèmes d'affectation quadratique
- problèmes d'ordonnancement
- problèmes de coloration de graphes, etc.

Concepts de base

Cette méthode combine une procédure de recherche locale avec un certain nombre de règles et de mécanismes permettant à celle-ci de surmonter l'obstacle des optima locaux.

En effet, en partant d'une solution quelconque x appartenant à l'ensemble de solutions X, on se déplace vers une solution $s(x)$ située dans le voisinage $[S(x)]$ de x. Donc l'algorithme explore itérativement l'espace de solutions X.

Pour choisir le meilleur voisin $s(x)$ dans $S(x)$, l'algorithme évalue la fonction objectif f en chaque point $s(x)$, et retient le meilleur voisin même si ce dernier dégrade la fonction objectif .

Concepts de base

Ce critère autorisant les dégradations de la fonction objectif évite à l'algorithme d'être piégé dans un minimum local, induit un risque de cyclage.

Afin d'éviter ce cyclage, on introduit une liste appelée liste taboue qui consiste à mémoriser les configurations ou régions visitées et à introduire des mécanismes permettant d'interdire à la recherche de retourner trop rapidement vers ces configurations.

Dans certains cas, les interdictions occasionnées par la liste taboue risquent d'éliminer (en les rendant tabous), certains mouvements particulièrement utiles. On introduit un critère aspiration qui permet d'assouplir ce mécanisme de liste taboue.

la méthode de recherche tabou

$\boldsymbol{x}$:= solution aléatoire
$f_{min} := f(\boldsymbol{x})$
$\boldsymbol{x}_{min} := \boldsymbol{x}$

TABOU := liste de solutions $s(\boldsymbol{x})$, de longueur L
TABOU := VIDE

REPETER
générer un N-échantillon TEL QUE $s_i(\boldsymbol{x}) \in$ voisinage $S(\boldsymbol{x})$ et $\{\boldsymbol{x}, s_i(\boldsymbol{x})\} \notin$ TABOU
$f(s(\boldsymbol{x})) = \min_{1 \le i \le N}[f(s_i(\boldsymbol{x}))]$
ajouter ($\{\boldsymbol{x}, s_i(\boldsymbol{x})\}$, TABOU)
$\boldsymbol{x} := s(\boldsymbol{x})$

SI $f(\boldsymbol{x}) < f_{min}$
$f_{min} := f(\boldsymbol{x})$
$\boldsymbol{x}_{min} := \boldsymbol{x}$
FIN DE SI
TestCritereAspiration
JUSQU'A conditions d'arrêt satisfaites

Listes de candidats

Dans la recherche avec tabou, on cherche «normalement» le meilleur mouvement (ou voisins) non tabou. Cependant, cette manière de procéder peut se révéler trop coûteuse.

Pour y remédier, on peut se limiter à engendrer seulement un sous-ensemble des voisins (ou mouvements). On parle de liste de candidats.

Construction d'une liste de candidats

Les candidats de la liste peuvent être engendrés de manière aléatoire (échantillon aléatoire).

Une autre approche consiste à sélectionner les voisins qui semblent les plus prometteurs – selon un critère quelconque.

Longueur de la liste tabou

Si la liste taboue est courte

Il y a moins d'interdictions (mouvements tabous).

La recherche épouse mieux les optima locaux rencontrés.

L'algorithme tend à parcourir de moins grandes distances dans l'espace de recherche. Il explore moins l'espace de recherche.

Le risque de cycles est plus grand.

Si la liste taboue est longue

Il y a davantage d'interdictions (mouvements tabous).

La recherche risque de manquer de nombreux optima locaux sur son chemin.

L'algorithme tend à parcourir de plus grandes distances dans l'espace de recherche. Il explore davantage l'espace de recherche.

Le risque de cycles est réduit.

Réglage auto-adaptatif de la longueur de la liste taboue

La longueur de la liste taboue est auto-adaptative

- Quand on rencontre une configuration déjà visitée, on augmente la durée.
- Sinon, on diminue progressivement la longueur de la liste.

Implémentation : au cours de la recherche, on mémorise les configurations visitées à l'aide d'une table de hachage.

Critère d'aspiration

Dans certains cas, les interdictions occasionnées par la liste taboue peuvent être jugées trop radicales. En effet, on risque d'éliminer (en les rendant tabous), certains mouvements particulièrement utiles. Autrement dit, il s'agit d'assouplir le mécanisme de liste taboue.

Un mécanisme d'aspiration détermine un critère selon lequel un mouvement, bien que tabou, peut quand même être accepté. Il faut faire attention, cependant, au risque d'introduire à nouveau des cycles dans la recherche.

Un critère d'aspiration rudimentaire peut consister à accepter un mouvement s'il conduit à une configuration meilleure que la meilleure configuration déjà trouvée.
Des mécanismes plus sophistiqués peuvent être introduits.
Cette correction permet aussi de revenir à une solution déjà visitée et de redémarrer la recherche dans une autre direction

Critères d’arrêt

La méthode ne s’arrête pas au premier optimum rencontré

Critères d’arrêt :

- arrêt après un nombre fixe d’itérations
- arrêt après un nombre d’itération sans amélioration de la fonction objectif

Adaptation de la méthode tabou à un problème d'optimisation à variables continues
Principe

- Une transposition directe de la méthode combinatoire
- Définir la notion de voisinage,
- Définir le pas
- Gestion de la liste tabou

Notion de voisinage et pas de mouvement

• Le choix des solutions voisines de la solution courante est important

• La définition d'un voisinage $S(\boldsymbol{x})$ de la solution courante $\boldsymbol{x}$ dans le cas d'une fonction à variables continues est délicat, et l'évaluation complète de ce voisinage continu est impossible.

• Dans la pratique, il est nécessaire de définir une stratégie d'échantillonnage de ce voisinage.

• La manière la plus simple d'échantillonner un voisinage déjà prédéfini consiste à générer aléatoirement un sous-ensemble discret de solutions $\boldsymbol{x}'$ voisines de $\boldsymbol{x}$ (appartenant à $S(\boldsymbol{x})$).

• Cette procédure n'est malheureusement pas la mieux adaptée aux problèmes à variables continues. En effet, les voisins $\boldsymbol{x}'$ d'une solution donnée $\boldsymbol{x}$, tirés aléatoirement, peuvent se concentrer dans une même zone.

• Dans la littérature, plusieurs stratégies de définition du voisinage et d'échantillonnage ont été proposées.

Subdivision de l'espace des solutions

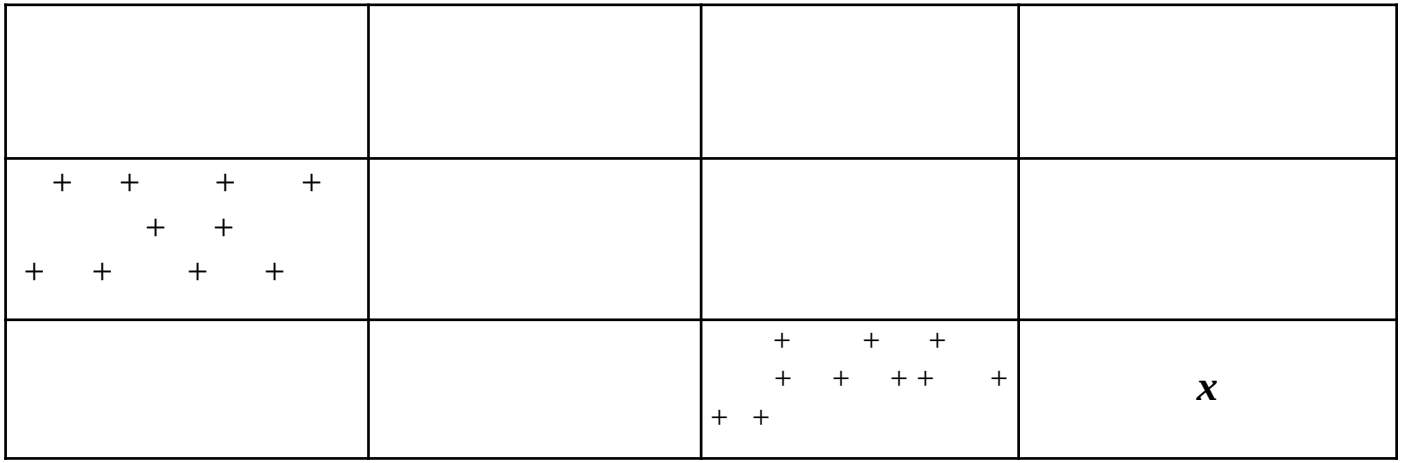
Cvijovic et Klinowski ont introduit dans un espace continu une structure de voisinage, qu'ils ont appelée "voisinage conditionnel". L'espace solution X (hypercube dans R^n, $X \subset R^n$) est partitionné en cellules disjointes par division des intervalles des coordonnées x_i en p_i parties.

Le vecteur de partionnement empirique $P = (p_1, p_2, ..., p_n)$ détermine un partionnement unique de X en cellules C de tailles plus ou moins grandes, et spécifie ainsi l'adresse de chaque cellule.

A chaque itération, le voisinage de la solution courante x est obtenu par tirage, suivant une loi de distribution uniforme, d'un échantillon de n_s points sur n_c cellules tirées aléatoirement.

La taille du voisinage de S(x) est égale à = n_s . n_c. Si un mouvement s d'une solution x vers x', de la cellule C(x) vers C(x'), est accepté, alors l'adresse de la cellule C(x'), contenant la solution x', est ajoutée à la liste tabou des adresses des cellules visitées.

Partition de l'espace des solutions et définition du voisinage de la solution **x :** exemple dans le cas n = 2.



L'exemple précédent (***P*** = (4**,**3), n_s =10 et n_c =2) montre l'inconvénient de cette méthode : le nombre de cellules croît avec la dimension du problème

Partitionnement par arbre binaire

Les auteurs ont introduit une structure de données dynamique pour représenter la structure de voisinage.

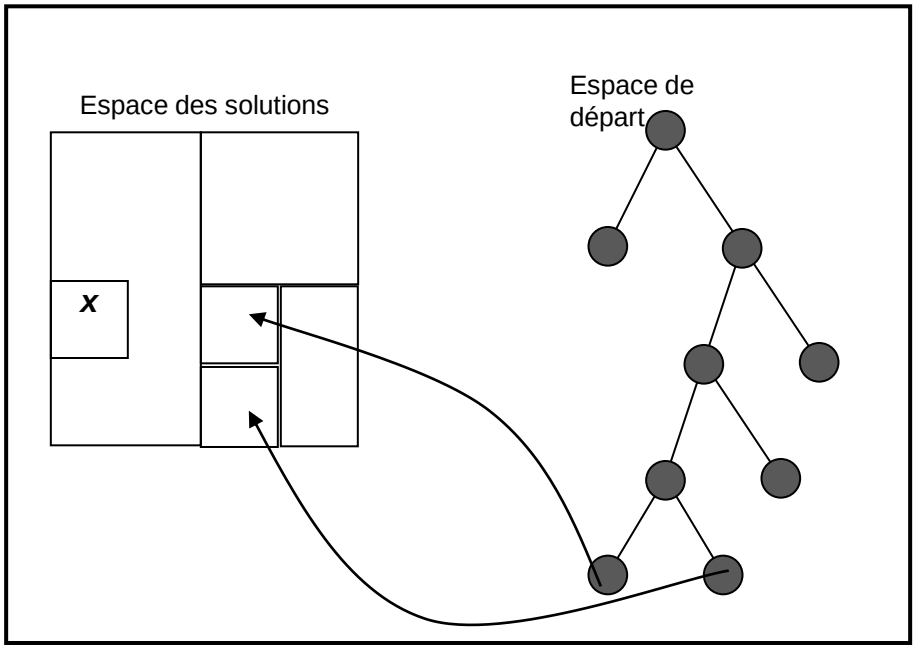
L'espace des solutions $\boldsymbol{X}$ (hypercube dans R^n, $\boldsymbol{X} \subset R^n$), est représenté par un arbre binaire à k degrés de liberté, où chaque niveau de l'arbre représente une partition binaire d'un de ses degrés de liberté.

Chaque nœud de l'arbre pourrait être interprété comme représentant un hyperrectangle parent, et ses nœuds-fils représentent les sous hyperrectangles fils résultant de la division par 2 de l'hyperrectangle parent, suivant un degré de liberté particulier.

Le voisinage de la solution courante $\boldsymbol{x}$ est obtenu par tirage, suivant une loi de distribution uniforme, d'un échantillon de n_s points sur les hyperrectangles voisins de celui qui contient $\boldsymbol{x}$.

Partition de l'espace des solutions à deux dimensions en utilisant un arbre binaire.

Cette figure illustre un exemple à deux dimensions d'un arbre de partition,
Le nombre de niveaux verticaux dans l'arbre correspond aux raffinements successifs.



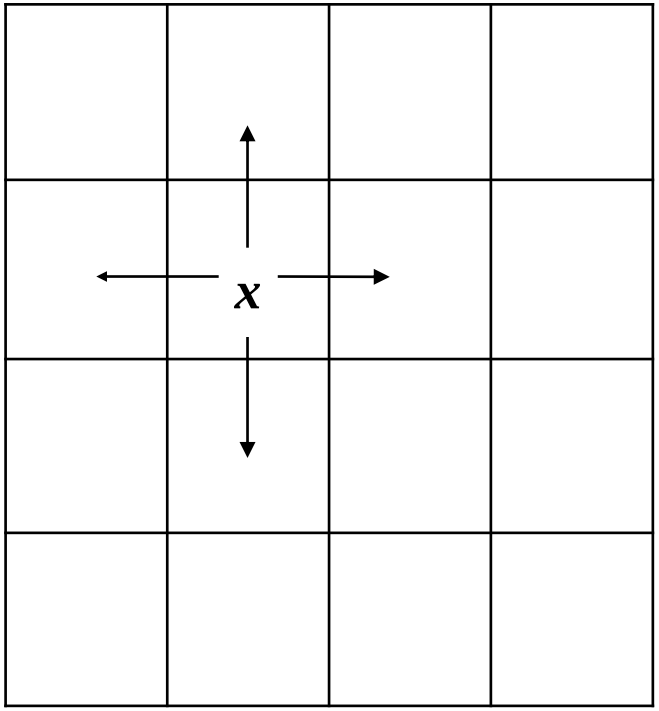
Cette méthode est efficace lorsque le nombre de variables est réduit ; dans le cas contraire, on s'expose à une explosion combinatoire des possibilités lorsqu'on veut raffiner la recherche.

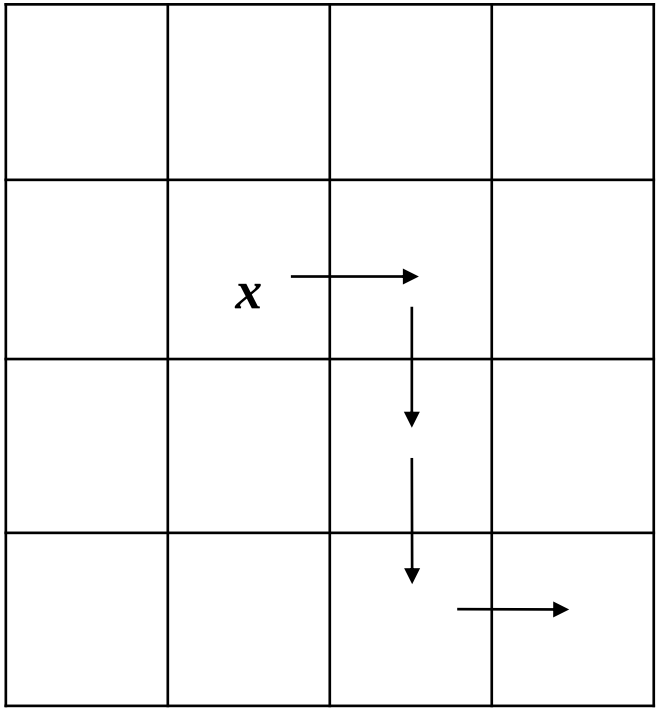
Mouvement par modification d'une seule composante

Le mouvement dans le voisinage de la solution courante consiste à modifier une par une les composantes de cette solution, et à choisir le meilleur voisin.

Le pas du mouvement dans chaque direction est constant. Cela revient à subdiviser l'espace des solutions en hyperrectangles égaux suivant chacune de ses cordonnées, à partir d'une solution $\boldsymbol{x}$, centre de l'un des hyperrectangles, et à définir comme voisins possibles les centres des hyperrectangles voisins.

Exemple de succession de mouvements à l'intérieur de l'espace des solutions





Cette exemple montre les mouvements possibles pour une fonction à deux variables.

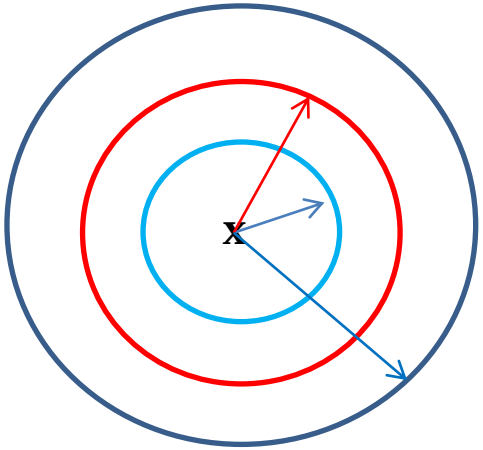
Notion de voisinage par topologie de boules

Un concept de "boule" est utilisé pour définir le voisinage.

Une boule $B(\boldsymbol{x},\varepsilon)$ est centrée sur $\boldsymbol{x}$ avec un rayon ε ; elle contient tous les points $\boldsymbol{x}'$ tels que : $// \boldsymbol{x}' - \boldsymbol{x} // = \varepsilon$ *(le symbole //.../ / est utilisé pour exprimer la norme euclidienne).*

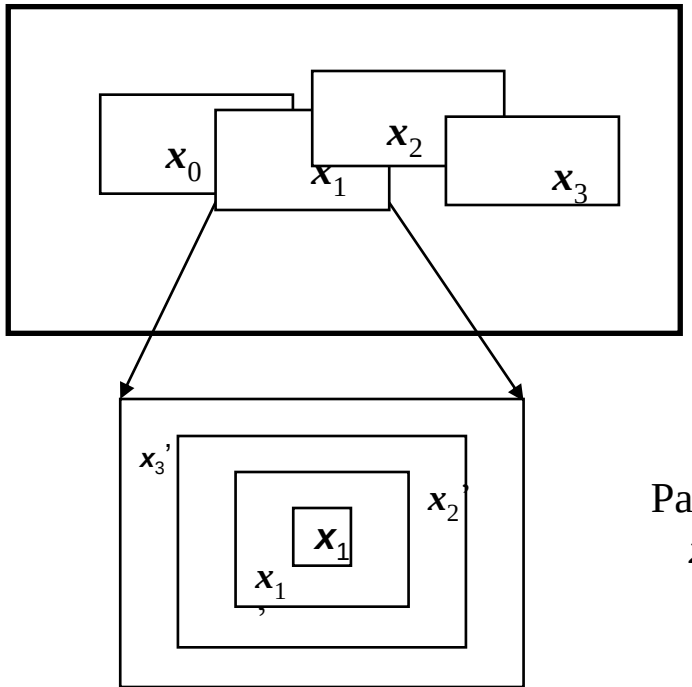
Pour obtenir une exploration homogène de l'espace de voisinage, on considère un ensemble de boules centrées sur la solution courante $\boldsymbol{x}$, avec des rayons h_0 , h_1 ,.., h_η. On obtient ainsi un espace partitionné en "couronnes" concentriques $C_i(\boldsymbol{x}, h_{i-1}, h_i)$

telles que $\quad C_i(\boldsymbol{x}, h_{i-1}, h_i) = \{\boldsymbol{x}' / h_{i-1} \leq // \boldsymbol{x}' - \boldsymbol{x} // \leq h_i \}.$



Les η voisins de $\boldsymbol{x}$ ont été obtenus *par sélection aléatoire d'un point à l'intérieur de chaque couronne* C_i , pour i variant de 1 à η.

Pour facilité le tirage aléatoire avec contrainte, on les boules par des hyperrectangles pour la partition du voisinage de la solution courante et nous générons des voisins de la solution courante $\boldsymbol{x}$ par modification d'une partie des composantes $\mathbf{x}_i$ en nous assurant que les autres composantes seront considérées aux générations suivantes de voisins. Le pas de variation $\Delta\mathbf{x}_i$ pour chaque composante $\mathbf{x}_i$ de la variable $\mathbf{x}$ est choisi de telle manière que le voisin généré se trouve dans la zone hyperrectangulaire voulue



Espace des solutions

Par exemple,
$\boldsymbol{x}_2 = \boldsymbol{x}_2'$

La méthode de tabou avancée

Un algorithme tabou de base comprend une liste taboue (mémoire à court terme) et un critère d'aspiration.

Un algorithme tabou «évolué» comprend en outre une technique de diversification et / ou une technique d'intensification.

Les techniques de diversification et d'intensification font appel à des mémoires à long terme.

Mémoire à long terme

Dans la méthode tabou, on peut utiliser des mémoires à long terme. Celles-ci peuvent servir à implanter des techniques d'intensification et/ou de diversification.

Solution d'élite (solutions prometteuses)
On mémorise certaines de meilleures solutions rencontrées au cours de la recherche.

Recency memory
Pour chaque attribut (ou composant = solution component), nombre d'itérations consécutives pendant lesquelles l'attribut a été présent dans la solution courante (sans interruption)

Frequency memory
Pour chaque attribut, nombre total d'itérations pendant lesquelles l'attribut a été présent dans la solution courante depuis le début de la recherche.
Ou nombre de fois que l'attribut a été impliqué dans un mouvement (a été modifié)

Techniques d'intensification

L'idée à la base de l'intensification est qu'on devrait explorer de façon plus approfondies les régions qui semblent les plus prometteuses.

Le principe de l'intensification consiste à retourner périodiquement visiter des zones de l'espace de recherche qui semblent particulièrement prometteuses.

De nombreuses techniques ont été proposées :

- Repartir de bonnes solutions déjà rencontrées ;
- Reconstruire une solution de départ qui tente de combiner des attributs qui ont été présents souvent dans les configurations visitées ;

Techniques de diversification

Le principe de la diversification consiste à inciter l'algorithme à se diriger vers des régions qui n'ont pas encore été visitées.

De nombreuses techniques ont été proposées :

Repartir d'une configuration aléatoire

Reconstruire une solution de départ qui tente de combiner des attributs qui ont été présents le moins souvent dans les configurations visitées.

Modifier la fonction de coût pour

- favoriser les attributs peu fréquents
- pénaliser les attributs fréquents

Techniques de diversification

Diversification par relance

On construit une solution qui contient des composants rarement utilisés, et on effectue une relance à partir de cette solution.

Diversification en continu

On biaise l'évaluation des mouvements en ajoutant à l'objectif un terme relié à la fréquence des attributs :

-Les attributs les plus fréquents sont pénalisés

-Les attributs les moins fréquents sont encouragés

Algorithme tabou avancé

INITIALISATION

de paramètres

↓

DIVERSIFICATION

détection de « zones prometteuses »

$\Rightarrow$ liste prometteuse

↓

SELECTION de la MEILLEURE ZONE PROMETTEUSE

parmi celles de la liste prometteuse

↓

INTENSIFICATION

à l'intérieur de la meilleure zone prometteuse

↓

MEILLEURE SOLUTION RENCONTREE

Algorithme tabou avancé