Recherche adaptative à grand voisinage

1. Recherche à grand voisinage

Méthode de recherche locale introduite par Shaw en 1998.

Le voisinage de la solution courante est défini implicitement à l'aide d'un opérateur de destruction et d'un opérateur de reconstruction..

Degré de destruction :

Trop faible : recherche locale standard

Trop fort : suite de reconstructions indépendantes

Exemple. Problème de voyageur de commerce

Destruction : retrait de sommets choisis de façon aléatoire

Reconstruction : réinsertion séquentielle des sommets à moindre coût

Pseudo-code

Construire une solution initiale $s \in X$

$s^* \leftarrow s$

Tant que le *critère d'arrêt* n'est pas satisfait faire :

$s' \leftarrow$ recontruction (destruction(s))

Si s' satisfait le *critère d'acceptation* alors $s \leftarrow s'$

Si $f(s') < f(s^*)$ alors $s^* \leftarrow s'$

Retourner s*.

Critère d'arrêt : Nombre d'itérations, nombre d'itérations consécutives sans améliorer s*, temps de calcul, etc.

Critère d'acceptation :

- critère classique

si $f(s') < f(s)$ alors $s \leftarrow s'$

- critère de type recuit simulé

si $f(s') < f(s)$ alors $s \leftarrow s'$

sinon on accepte s' avec probabilité $e^{-(f(s') - f(s))/T}$

2. Recherche adaptative à grand voisinage

   Extension de la recherche à grand voisinage avec plusieurs opérateurs de destruction et de reconstruction.

   À chaque itération, un opérateur de destruction et un opérateur de reconstruction doivent être sélectionnés.

   Soit D et R les ensembles d'opérateurs de destruction et de reconstruction, respectivement.

   À chaque opérateur de destruction $d_j \in D$ et de reconstruction $r_j \in R$ est associé un poids w_j^D et w_j^R.

   La probabilité de sélection de chaque opérateur de destruction $d_j \in D$ est alors:

$$p_j^D = \frac{w_j^D}{\sum_{k=1}^{|D|} w_k^D}$$

   Une probabilité de sélection est associée à chaque opérateur de reconstruction $r_j \in R$ de la même façon.

   Adaptation ou ajustement des poids

   Le poids des opérateurs est le même au début de l'algorithme

   Un score est calculé pour les deux opérateurs choisis à l'itération courante en fonction de leur performance :

$$\varphi = \begin{cases} \varphi_1 & \text{si la nouvelle solution est la meilleure solution connue} \\ \varphi_2 & \text{si la nouvelle solution est meilleure que la solution courante} \\ \varphi_3 & \text{si la nouvelle solution est acceptée} \\ \varphi_4 & \text{si la nouvelle solution est rejetée} \end{cases}$$

   avec $\varphi_1 \geq \varphi_2 \geq \varphi_3 \geq \varphi_4$.

   Le poids de l'opérateur de destruction choisi d'indice j* est alors modifié avec la formule suivante où $\lambda \in [0, 1]$ est un paramètre de lissage:

$$w_{j*}^D = \lambda w_{j*}^D + (1-\lambda)\varphi$$

   Le poids de l'opérateur de reconstruction choisi est modifié de la même façon.

3. Opérateurs de destruction

   - Destruction aléatoire
   - Destruction d'éléments critiques (e.g., coût élevé)
   - Destruction d'éléments présentant des caractéristiques communes et qui peuvent être facilement interchangés afin de produire une nouvelle solution
     - Proximité (e.g., géographique ou temporelle)
     - Appartenance à un même sous-ensemble (e.g., clients dans une même route)
   - Destruction basée sur l'historique de la recherche
     - Identification d'éléments associés à une solution de mauvaise qualité

       Par exemple, dans un problème de tournées de véhicules défini sur un graphe orienté, on conserve le coût de la meilleure solution où un arc est utilisé, et ce pour tout arc dans le graphe. Si ce coût pour un arc (i, j) demeure élevé, il n'est pas désirable que le client j soit visité immédiatement après le client i dans une solution et on favorisera donc la destruction de cette structure.

4. Opérateurs de reconstruction

   - Reconstruction exacte
     - énumération systématique
     - programmation par contraintes
   - Reconstruction heuristique
     - insertion séquentielle à moindre coût

       On considère Δf_i la modification au coût de la solution suite à la meilleure réinsertion possible de l'élément i.

       Parmi tous les éléments qui doivent être réintroduits dans la solution on choisit celui qui minimise Δf_i.

     - méthode des regrets

       On considère Δf_i^1 et Δf_i^2 la modification au coût de la solution suite à la meilleure et à la deuxième meilleure réinsertion possible de l'élément i et on calcule le regret $r_i = \Delta f_i^2 - \Delta f_i^1$.

       Parmi tous les éléments qui doivent être réintroduits dans la solution on choisit celui qui maximise le regret.

       Le regret nous indique la perte encourue si l'élément i ne peut être réintroduit dans la solution à moindre coût.