

Aicha Miloudi, 1872134

Maude Prévile, 1805556

Résolution par recherche locale

Nous avons décidé de représenter un état (solution) par une classe State qui a comme attributs une liste des générateur disponibles, une liste des générateurs allumés, une liste des assignations des générateurs aux machines et un coût total.

- `available_generators`: La liste des générateurs disponibles est une liste qui représente les générateurs qu'on peut ou non allumer. Si un générateur ne se trouve pas dans cette liste, il est considéré comme fermé et ne pourra pas être réouvert dans une prochaine itération. C'est cette liste qu'on utilise pour décrire le voisinage possible
- `opened_generators`: La liste des générateurs allumés représente les générateurs actuellement allumés. Lors d'une itération, si un générateur de cette liste est éteint, il est possible de le rallumer plus tard (dans une prochaine itération) puisqu'il demeure dans la liste des générateurs disponibles.
- `assigned_generators`: Dans la liste des assignations, pour chaque machine on retrouve le numéro du générateur qui lui est assigné.
- `total_cost`: le coût de la configuration de la solution

La classe a aussi une méthode `generate_state()` qui s'occupe d'associer chaque machine au générateur le plus proche à partir de la liste des générateurs disponibles et ainsi générer les listes `opened_generators` et `assigned_generators` et le coût total.

L'état initial est celui où tous les générateurs sont disponibles.

Fonction de voisinage : on enlève un générateur de la liste des générateurs disponibles de l'état actuel

Fonction d'évaluation : le coût total calculé en associant chaque machine au générateur disponible le plus proche et en prenant compte du coût d'activation des générateurs.

Fonction de validation : si le coût total du voisin qu'on obtient en appliquant sur l'état actuel la fonction de voisinage est plus petit que le coût de l'état actuel, cet état voisin est valide

Fonction de sélection : Parmi les voisins valides, nous prenons le minimum, c'est-à-dire la solution qui a le plus petit coût total.

Critère d'arrêt : on s'arrête dès qu'on rencontre une des conditions suivantes :

- On a fait $n-1$ itérations où n représente le nombre de générateurs
- On arrive à un état qui ne génère aucun voisinage meilleur que celui actuel.