

### 3. Les heuristiques

Malgré leur manque de comportement théoriquement prouvable, les algorithmes heuristiques ont eu un succès impressionnant dans plusieurs problèmes.

Ces techniques permettent d'extraire des informations globales (pour se guider) à partir des grands volumes de données traitées au cours de temps par le processus de recherche. On qualifie généralement leur comportement par des mouvements aléatoire associé avec une certaine intelligence. C'est pour cette raison que les algorithmes heuristique sont évalués par des techniques de statistique.

Méta-heuristiques est une autre terminologie légèrement différente mais très utilisé. Le terme *méta* est donc pris au sens où les algorithmes peuvent regrouper plusieurs heuristiques.

#### 3.1. Classification

Il existe plusieurs critères de classification mais on va se limité aux critères les plus connus :

##### a. Parcours et population

Dans le type Parcours (*trajectoire*), l'algorithme fait évoluer un seul point (*particule*) sur l'espace de recherche à chaque itération. De ce fait, la notion de voisinage est alors primordiale. Les plus connues dans cette classe sont le **recuit simulé**, la **recherche avec tabous**, la **recherche locale**, la **recherche à voisinages variables**,

Dans le type Population. La métaheuristique manipule un ensemble de points en parallèle, à chaque itération. On peut citer les algorithmes **génétiques** et **PSO**.

##### b. Évolutif ou non

L'algorithme sera considéré comme évolutif s'il manipule une population via des opérateurs, selon un algorithme général donné. On parlera d'opérateurs pour toute action modifiant l'état d'une ou plusieurs points (*solutions*). Un opérateur construisant un nouveau point sera dénommé générateur, alors qu'un opérateur modifiant un point existante sera appelé mutateur.

### c. Méthodes avec ou sans mémoire

Dans les algorithmes sans mémoire, l'action à réaliser est totalement déterminée par la situation courante on ne prend pas le passé de la recherche en considération.

Généralement, on fait même la différence entre les méthodes ayant une mémoire à court terme de celles qui ont une mémoire à long terme.

### 3.2. Théorème du « no free lunch »

Le théorème du « no free lunch » explique qu'aucun algorithme métaheuristique ne peut prétendre être la meilleure sur tous les problèmes. Une métaheuristique (M) n'est performante que pour une classe de problème (P) donnée.

