Deux méthodes principales

* Master - slave -> parallélisation du processus de sélection (openMP)
* Island -> génération de plusieurs population plus ou moins indépendantes qui échangent occasionnellement leurs individu (MPI)

Master Slave

Une seule population.  
A chaque génération, l’évaluation de la valeur de fitness des individus est calculée en parallèle.

Island :

Algorithme distribué.  
Chaque site dispose de sa propre population.

A chaque génération les sites envoient un pourcentage des meilleurs individus de leur population à leurs voisins.  
Ces meilleurs individus remplacent les pires individus de la population.

Après la dernière génération, une réduction est réalisée pour définir le meilleur individu de tout les sites.

L’évaluation des individus peut être parallélisée selon le modèle Master Slave.

La disposition des sites peut différer selon les modélisations (anneau, grille torique).  
Cependant, le nombre de communications ne doit pas être trop grand (cela peut ralentir l’algorithme).  
  
Le principe du modèle en “île” est d’ajouter plus de pression sur la sélection des individus. En effet, les sites étant indépendants, ils peuvent réaliser leurs sélections selon des paramètres différents (taux de mutation, opérateurs…). Ainsi, chaque site peut générer des individus différents.

Par contre, ce modèle ne vise pas à accélérer l’exécution de l’algorithme en lui même mais d’améliorer la qualité de la solution approchée.

TP4 :

Implémentation :

La structure d’arbre de la méthode de séparation évaluation est une pile.  
Cette pile nous permet de mettre en place un parcours en profondeur.

Dans ce TP, un parcours était demandé, cependant, l’algorithme de séparation évaluation peut aussi être appliqué avec un parcours en largeur à l’aide d’une file.

Création d’un ACPM :

Pour réaliser l’ACPM du graphe excluant un sommet donné, nous utilisons l’algorithme de Kruskal. Cet algorithme nécessitant de trier le tableau d'arêtes par ordre croissant de poids, un algorithme de tri à bulles à été implémenté.  
  
Choix du tri à bulles :

Initialement, un tri par fusion avait été implémenté. Ce tri était utilisé pour son efficacité (O(n log n)).  
Cependant, la vaste majorité des cas, lorsque nous avons besoin d’utiliser cet algorithme, nous en avons besoin pour trier un tableau contenant uniquement une valeur nécessitant d’être triée.  
Ainsi, l’utilisation d’un tri par fusion semble peu judicieuse dans ce cas.

Notre attention s’est ainsi portée sur l’algorithme de tri à bulles. En effet, ce dernier a une complexité temporelle en O(n) dans le meilleurs cas (tableau déjà trié).

Comme indiqué précédemment, le tableau étant pratiquement déjà trié, un algorithme de tri à bulle s’avère être bien plus performant qu’un tri par fusion.

Evaluation d’un noeud :

L’évaluation d’un potentiel chemin hamiltonien se fait en deux temps.  
Dans un premier temps, la longueur du chemin est calculée afin de définir si ce dernier est un candidat potentiel.  
Si la longueur calculée est inférieure à la borne, on vérifie si le chemin est un chemin hamiltonien.

Pour définir ceci, on note le nombre d’occurence de chaque sommet dans le chemin.  
Si un sommet apparaît plus de deux fois dans le chemin, cela signifie qu’il a un degré supérieur à deux, rendant le chemin non hamiltonien.

Si un tel sommet est trouvé, la fonction d’évaluation retourne le numéro du sommet.  
Sinon, elle retourne -1.  
  
Séparation :

En fonction de ce que la fonction vérifiant le chemin retourne, l’algorithme de séparation crée les fils interdisant les arêtes nécessaires.

Chaque fils créé connait l’arête qu’il est censé interdir et sa valeur.  
Lorsque ce fils est dépilé pour être évalué, on interdit l'arête correspondante.