

Métaheuristiques parallèles

Mathieu Larose

15 avril 2010

Article et auteurs

Article : *Parallel Meta-heuristics*, 2009.

Auteurs

- Teodor Gabriel Crainic, *CIRRELT*.
- Michel Toulouse, *CIRRELT*.

Calcul parallèle

Définition

Plusieurs tâches travaillent simultanément sur un même problème.

Calcul parallèle

Définition

Plusieurs tâches travaillent simultanément sur un même problème.

Coût du parallélisme

- Lancement des tâches
- Synchronisation
- Communication

Calcul parallèle

Définition

Plusieurs tâches travaillent simultanément sur un même problème.

Coût du parallélisme

- Lancement des tâches
- Synchronisation
- Communication

Efficacité du parallélisme (*speedup*)

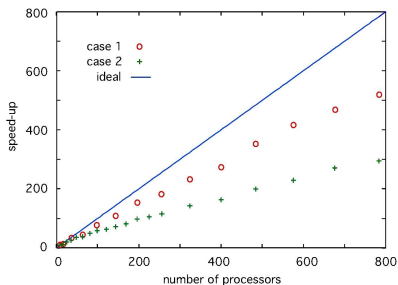
- $S_p = \frac{T_1}{T_p}$
- T_1 : temps de calcul du meilleur algorithme séquentiel.
- T_p : temps de calcul en parallèle avec p processeurs.

Calcul parallèle

Efficacité du parallélisme (*speedup*)

- $S_p = \frac{T_1}{T_p}$
- T_1 : temps de calcul du meilleur algorithme séquentiel.
- T_p : temps de calcul en parallèle avec p processeurs.

- Idéalement, si on double le nombre de processeurs, on voudrait que le programme s'exécute deux fois plus vite. ($S_p = p$)



Mesures de performance des métaheuristiques

- ① Qualité de la solution trouvée
- ② Temps de calcul
- ③ Robustesse

Trois dimensions des métaheuristiques parallèles

Nombre de processus contrôlant la recherche

1-control (1C) Contrôlée par un seul processus.

p-control (pC) Contrôlée par plusieurs processus.

Trois dimensions des métaheuristiques parallèles

Nombre de processus contrôlant la recherche

1-control (1C) Contrôlée par un seul processus.

p-control (pC) Contrôlée par plusieurs processus.

Communication

Rigid (RS) Système de communication simple.

Knowledge Synchronization (KS) échanges synchrones.

Collegial (C) échanges asynchrones sans inférence de nouvelles connaissances.

Knowledge Collegial (KC) échanges asynchrones avec inférence de nouvelles connaissances.

Trois dimensions des métaheuristiques parallèles

Variation dans la recherche

SPSS *Same initial Point/Population, Same search Strategies*

SPDS *Same initial Point/Population, Different search Strategies*

MPSS *Multiple initial Points/Populations, Same search Strategies*

MPDS *Multiple initial Points/Populations, Different search Strategies*

Quatre classes de métaheuristiques parallèles

- Parallélisme de bas niveau
- Décomposition du domaine
- Recherches multiples indépendantes
- Stratégies de recherche coopératives

Parallélisme de bas niveau

Métaheuristiques à base de voisinage

Évaluation en parallèle du voisinage.

Métaheuristiques à base de population

Évaluation en parallèle d'une population.

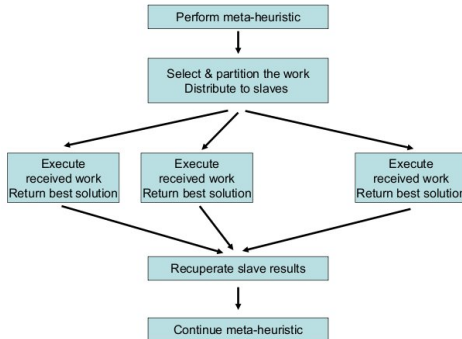
Parallélisme de bas niveau

Métaheuristiques à base de voisinage

Évaluation en parallèle du voisinage.

Métaheuristiques à base de population

Évaluation en parallèle d'une population.



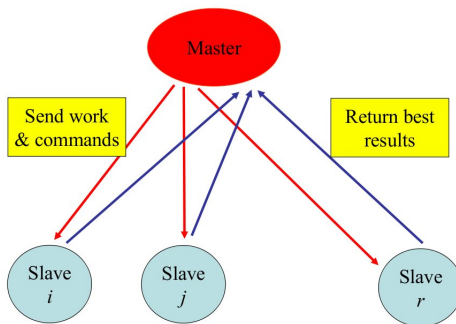
Parallélisme de bas niveau

Métaheuristiques à base de voisinage

Évaluation en parallèle du voisinage.

Métaheuristiques à base de population

Évaluation en parallèle d'une population.



Parallélisme de bas niveau

Métaheuristiques à base de voisinage

Évaluation en parallèle du voisinage.

Métaheuristiques à base de population

Évaluation en parallèle d'une population.

- Type de parallélisme : 1C/RS/SPSS.
- La logique de l'algorithme et l'espace de recherche (par rapport à la version séquentielle) ne sont pas modifiés.
- Le seul gain possible est en terme de vitesse d'exécution.

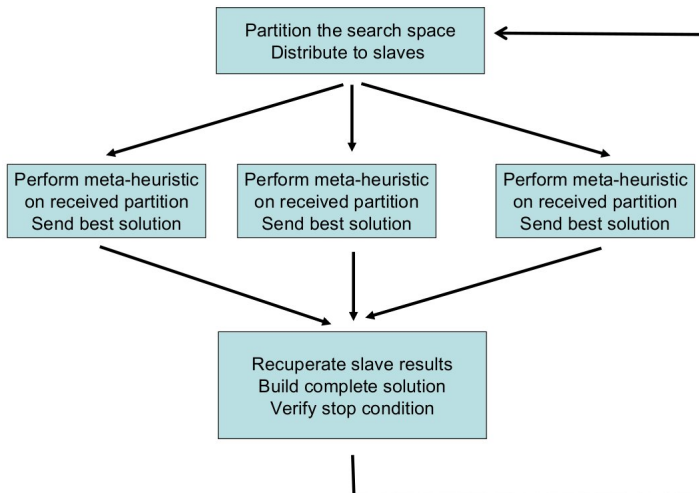
Décomposition du domaine

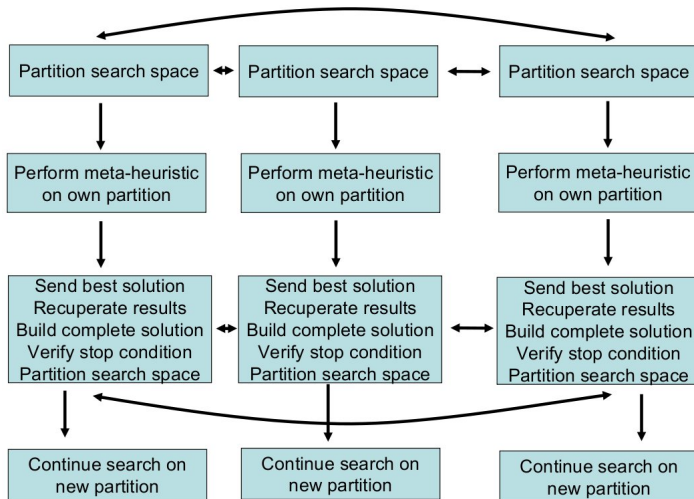
Décomposition de l'espace de recherche ou décomposition d'un problème en sous-problèmes.

Deux approches

- 1 1C/KS
- 2 pC/KS

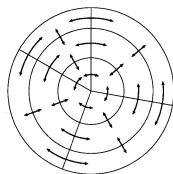
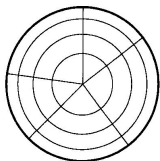
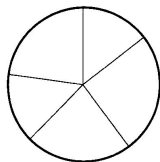
Implanté selon le modèle maître-esclave.





VRP (Taillard, 1993)

- Partition des clients en sous-ensembles disjoints et exhaustifs.
- Chaque sous-problème est résolu à l'aide d'une recherche tabou.
- La synchronisation entre les processus se fait à chaque n itérations.
- Un sous-problème ne communique qu'avec les sous-problèmes adjacents.
- Informations échangées : clients non desservis.



Recherches multiples indépendantes

Propriétés

- Les processus travaillent en parallèle de manière indépendante.
- pC/RS/{SPDS, MPSS, MPDS}
- La solution choisie est la meilleure parmi toutes les solutions trouvées.
- Facile d'implantation.

Stratégies de recherche coopérative

Stratégies de recherche coopérative

Description Plusieurs processus travaillent simultanément en s'échangeant de l'information.

But Obtenir de meilleurs résultats que la simple concaténation des processus pris individuellement.

Défi Partager de l'information pertinente au bon moment.

Stratégies de recherche coopérative

Stratégies de recherche coopérative

Description Plusieurs processus travaillent simultanément en s'échangeant de l'information.

But Obtenir de meilleurs résultats que la simple concaténation des processus pris individuellement.

Défi Partager de l'information pertinente au bon moment.

Sous-classes

- Synchrones
 - pC/KS
- Asynchrones
 - pC/C (sans inférence de nouvelles connaissances)
 - pC/KC (avec inférence de nouvelles connaissances)

pC/KS (Synchrone)

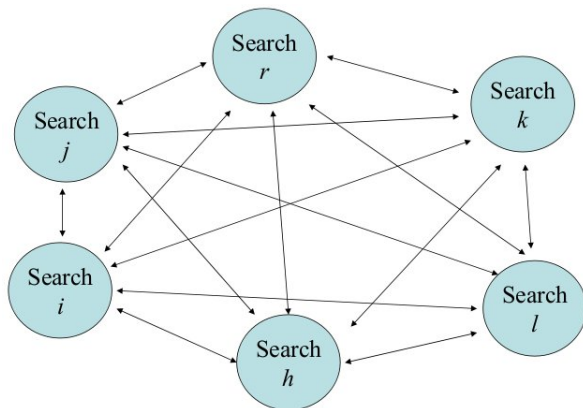
Algorithme

- 1 Les processus exécutent leur recherche.
- 2 Synchronisation entre les processus.
- 3 Communication entre les processus.
- 4 Vérification du critère d'arrêt.
- 5 Détermination de la prochaine phase de synchronisation (nombre d'itérations, temps, etc.)

pC/KS (Synchrone)

Recherche tabou

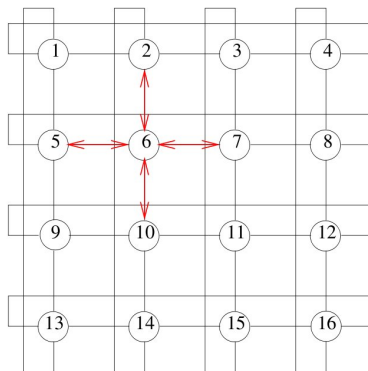
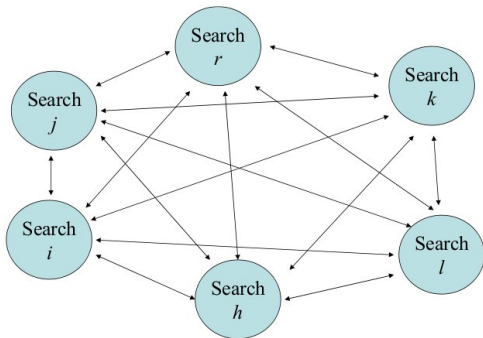
Les processus recommencent avec la meilleure solution trouvée. (SPDS)



pC/KS (Synchrone)

Algorithme génétique

Migration entre les populations.



Asynchrone

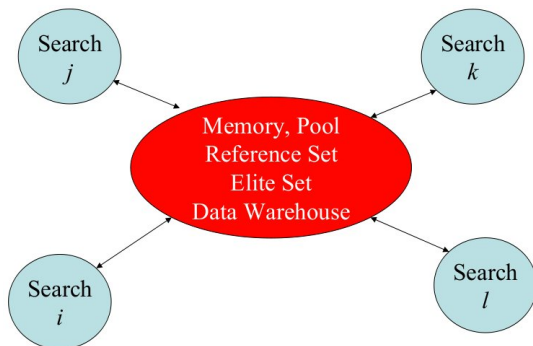
Caractéristiques d'une recherche coopérative asynchrone.

- Les processus sont autonomes.
- Échanges d'informations durant la recherche.
- Mémoire centrale
- Peu de temps perdu pour la communication.

Asynchrone

Caractéristiques d'une recherche coopérative asynchrone.

- Les processus sont autonomes.
- Échanges d'informations durant la recherche.
- Mémoire centrale
- Peu de temps perdu pour la communication.



Asynchrone

Caractéristiques d'une recherche coopérative asynchrone.

- Les processus sont autonomes.
- Échanges d'informations durant la recherche.
- Mémoire centrale
- Peu de temps perdu pour la communication.

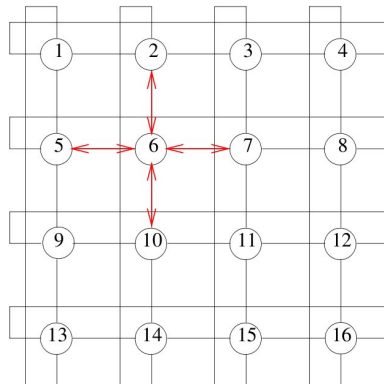
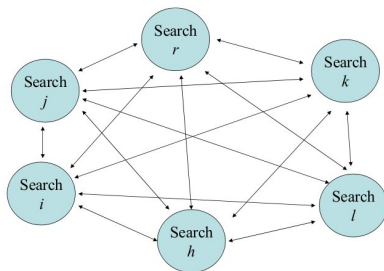
Deux approches

- ① pC/C (sans inférence de nouvelles connaissances)
- ② pC/KC (avec inférence de nouvelles connaissances)

pC/C (Asynchrone)

Algorithme génétique

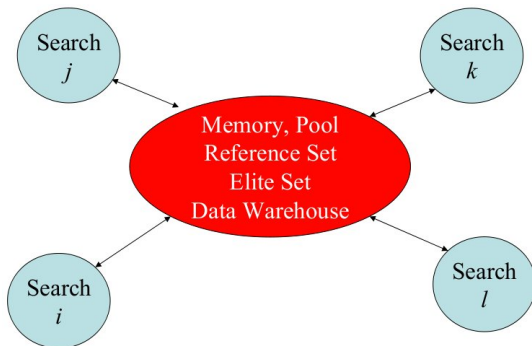
- Plusieurs populations avec beaucoup d'individus.
- Migration entre les populations.



pC/C (Asynchrone)

Métaheuristiques non exclusivement génétiques

- Les n meilleurs éléments sont gardés dans la mémoire centrale.
- Sélection aléatoire (biaisée ou non) des éléments dans la mémoire centrale.



pC/C (Asynchrone)

Métaheuristiques non exclusivement génétiques

- Les n meilleurs éléments sont gardés dans la mémoire centrale.
- Sélection aléatoire (biaisée ou non) des éléments dans la mémoire centrale.

Caractéristiques

- Plusieurs métaheuristiques peuvent être utilisées (ainsi que plusieurs types de méthodes).
- Diversification simple : suppression d'une partie de la mémoire centrale.

pC/KC (Asynchrone)

Observation

Les éléments dans la mémoire centrale forment une population élite.

Idées

- Produire des statistiques.
- Appliquer des méthodes pour améliorer cette population.

Le Bouthiller et Crainic, 2005

Problème

VRPTW

Deux algorithmes génétiques

- Order (OX)
- Edge Recombination (ER)

Deux recherches tabous

- Unified Tabu
- Taburoute

Mémoire centrale

Les éléments stockés dans la mémoire centrale sont des solutions complètes.

Le Bouthiller et Crainic, 2005

Post-optimisation

Un processus applique une post-optimisation (2-opt, 3-opt, Or-opt et une procédure de chaîne d'éjection) à toutes les nouvelles solutions dans la mémoire centrale.

Statistiques

Trois groupes de solutions sont créés :

- ① 10% meilleures
- ② 10% - 90% meilleures
- ③ 10% pires

La fréquence des arcs dans chacun des groupes est utilisée lors des périodes d'intensification et de diversification.

Le Bouthiller et Crainic, 2005

Conclusion

- Sans trop de calibration, l'algorithme donnent de bons résultats (robustesse).
- *Speedup* linéaire (pour 5 processeurs).

Recherche coopérative par intégration (*Integrative Cooperative Search, ICS*)

Métaheuristique parallèle pour résoudre des problèmes où les attributs sont de natures différentes. Ce sont des problèmes *riches* (*rich problem*).

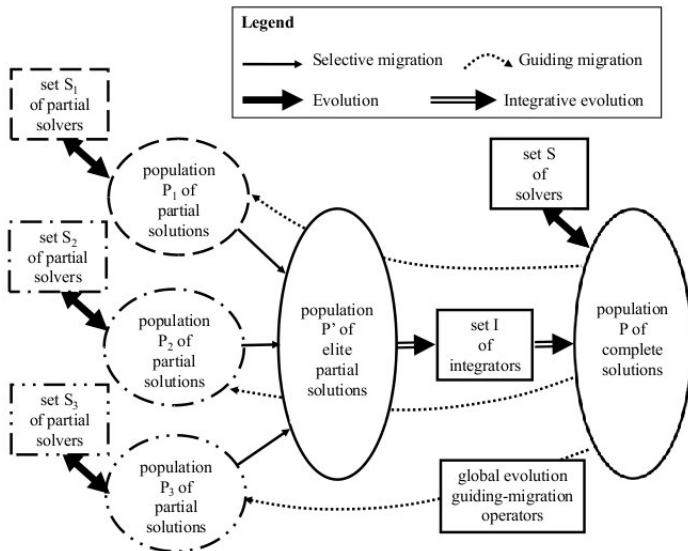
Quatre composantes

Solver partiel Métaheuristiques ou méthodes exactes travaillant de manière indépendante sur un sous-ensemble des attributs.

Intégrateur Combine les solutions partielles.

Solver complet Métaheuristiques travaillant sur une solution complète.

Coordonnateur Surveille la mémoire centrale et l'information échangée afin de coordonner la recherche.



Crainic et al., 2009

Periodic Vehicle Routing Problem (PVRP)

- Un client doit être visité plusieurs fois.
- Exemple des préférences pour un client qui doit être visité deux fois : (lundi, mercredi) ou (mardi, vendredi).

Crainic et al., 2009

Periodic Vehicle Routing Problem (PVRP)

- Un client doit être visité plusieurs fois.
- Exemple des préférences pour un client qui doit être visité deux fois : (lundi, mercredi) ou (mardi, vendredi).

Multi-Depot Vehicle Routing Problem (MDVRP)

Plusieurs dépôts.

Crainic et al., 2009

Periodic Vehicle Routing Problem (PVRP)

- Un client doit être visité plusieurs fois.
- Exemple des préférences pour un client qui doit être visité deux fois : (lundi, mercredi) ou (mardi, vendredi).

Multi-Depot Vehicle Routing Problem (MDVRP)

Plusieurs dépôts.

Multi-Depot Periodic Vehicle Routing Problem with Time Windows (MDPVRPTW)

- Les clients sont visités plusieurs fois (PVRP)
- et pour chaque visite il y a une fenêtre de temps pour le servir (VRPTW)
- de plus, les véhicules peuvent être associés à n'importe quel dépôt (MDVRP).

Crainic et al., 2009

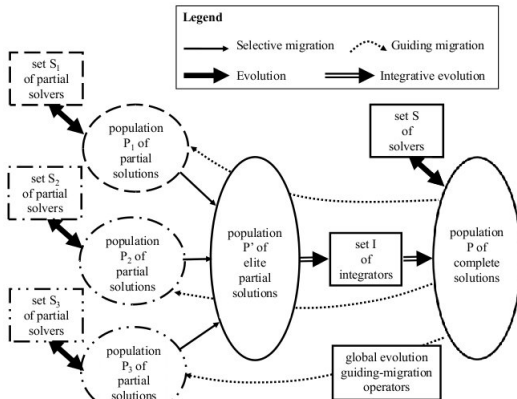
Deux types de *solver* partiel

P_1

- Résoud des instances où chaque client a un dépôt assigné.
- Chaque dépôt, avec ses clients qu'il dessert, représente une instance de type PVRPTW.

P_2

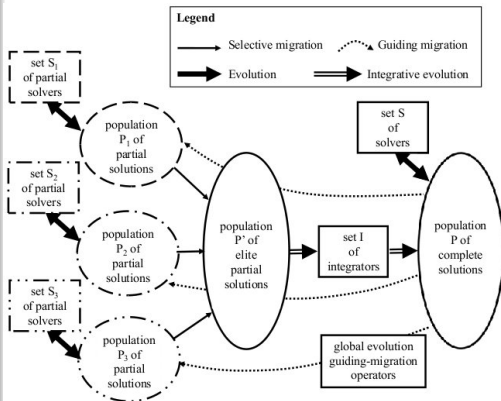
- Résoud des instances où les visites aux clients sont fixées.
- Chaque journée représente une instance de type MDVRPTW.



Crainic et al., 2009

Intégrateur

- Les solutions des instances PVRPTW indiquent de bonnes paires (client, visites).
- Les solutions des instances MDVRPTW indiquent de bonnes paires (client, dépôt).
- Construit des instances de VRPTW à partir de ces informations, puis les résout pour former une solution complète.
- Possibilité de combiner des morceaux des solutions partielles.



Crainic et al., 2009

Coordonnateur (Mécanisme de feedback)

Détecte les situations indésirables :

- Peu de diversité dans P .
- Stagnation dans l'amélioration de la valeur de la meilleure solution.
- Certaines zones de l'espace de recherche n'ont pas été explorées (diversification)

et réoriente la recherche.

