Université Lille Nord de France
Pôle de Recherche et d'Enseignement Supérieur

UNIVERSITÉ D'ARTOIS

LGI2A

Métaheuristiques pour la Résolution de Problème de Covoiturage Régulier de Grande Taille et d'une Extension

Yuhan GUO

Composition du jury

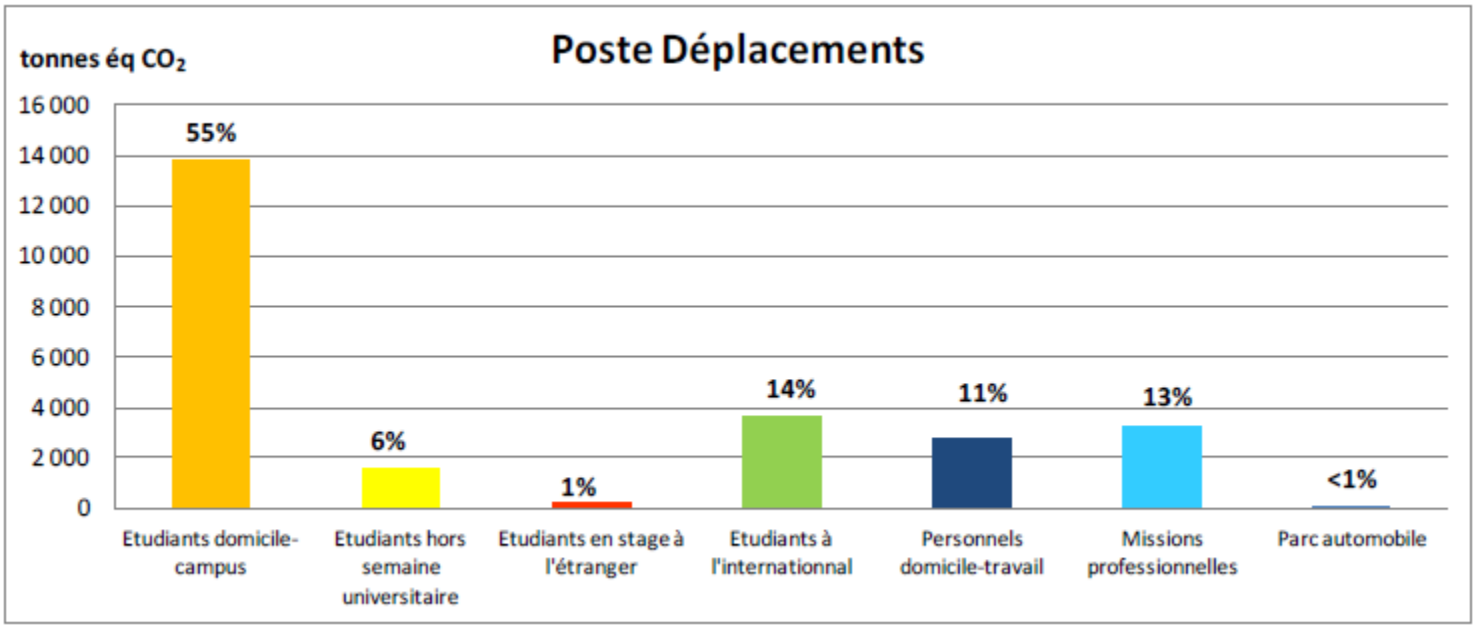
Rapporteurs:	M. Van-Dat CUNG	Professeur, Institut Polytechnique de Grenoble
	M. Jin-Kao HAO	Professeur, Université d'Angers
Examinateurs:	Mme. Marie-José HUGUET	HDR, MCF, Institut National des Sciences Appliquées de Toulouse
	M. El-Ghazali TALBI	Professeur, Université des Sciences et Technologies de Lille
Directeur de thèse:	M. Gilles GONCALVES	Professeur, Université d'Artois
Co-encadreur:	M. Tienté HSU	MCF, Université d'Artois

Plan

- **Introduction**
  - ✓ Définition du problème
  - ✓ Modèle mathématique
  - ✓ Objectifs de notre travail
- **Méthodes de résolution**
  - ✓ Recherche locale à voisinage variable
  - ✓ Algorithme à base de colonies de fourmis
  - ✓ Algorithme génétique guidé
  - ✓ Système multi-agents génétiques auto-adaptatif
- **Conclusions et perspectives**

Covoiturage

Dispersion spatiale de l'habitat



Bénéfices du covoiturage

Parking

Réduire l'occupation de parking.

Véhicules privés

Diminuer le nombre de véhicules privés sur la route.

Dépenses de carburant

Réduire les dépenses de carburant et de maintenance

Bruit et Pollution

Diminuer les bruits et la pollution.

Accidents

Diminuer les accidents de la route.

Embouteillage

Diminuer les embouteillages.

Sites Webs du covoiturage

- **USA**
  - CarPoolWorld.com
  - iCarPool.com
  - Zimride.com (100 Universités)

- **France**
  - Covoiturage.fr (1 000 000 membres inscrits)
  - Covoiturage.roulezmalin.com (50 000 membres actifs)

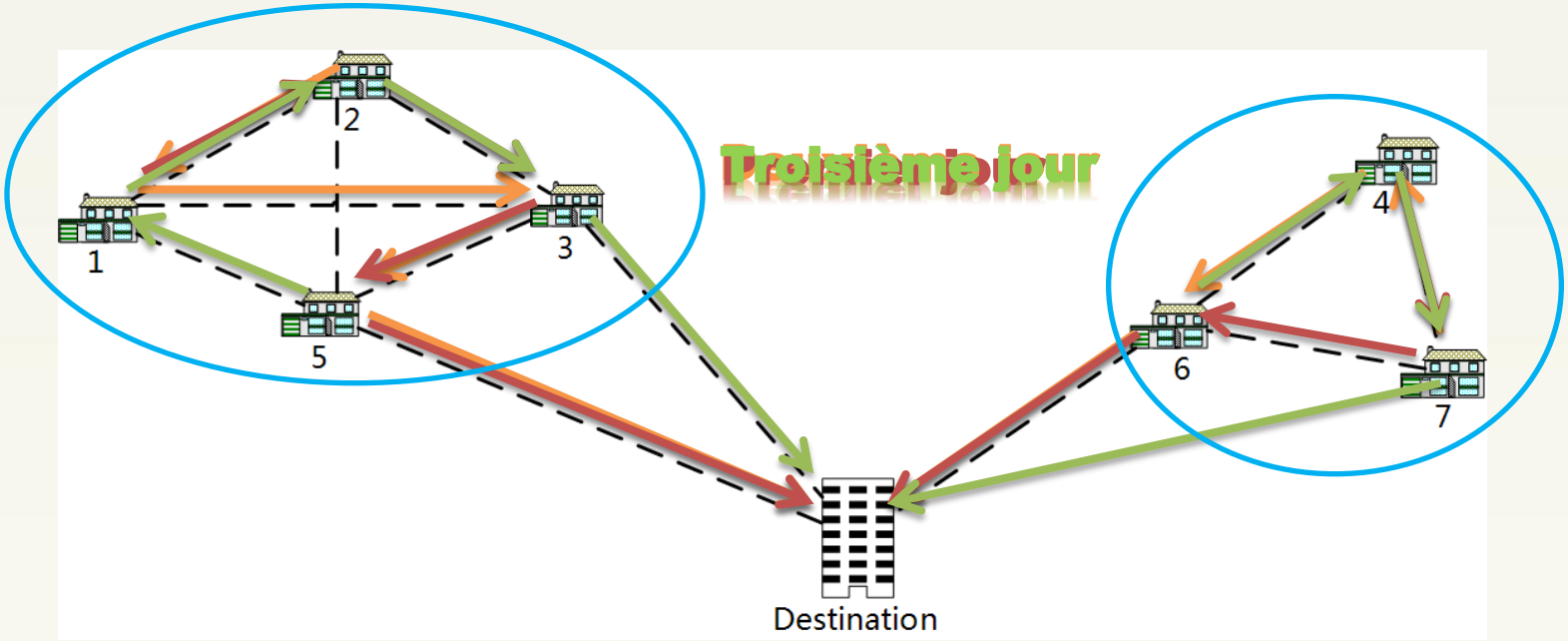
Catégorisation du problème de Covoiturage

- **Problème de covoiturage occasionnel**
  - Serveurs: un certain nombre d'usagers sont disponibles pour transporter d'autres usagers à un jour particulier.
  - Clients: les usagers prises en charge par les serveurs.
  - Affecter les clients aux serveurs et déterminer les trajets des serveurs.

- **Problème de covoiturage régulier**
  - Chaque usager agit alternativement comme serveur ou client;
  - Définir les groupes où chaque usager, à tour de rôle, ramasse les autres membres du groupe.
  - Très peu d'applications sont conçues pour cette forme.

Définitions du problème de covoiturage régulier

- Définir les groupes;
- Chaque usager agit alternativement comme serveur ou client;
- Chaque usager, à tour de rôle, ramasse les autres membres du groupe.



Définitions du problème de covoiturage régulier

- **Objectifs**:
  - Minimiser la distance totale parcourue par tous les usagers;
  - Minimiser le nombre de groupes;
  - ➡ Respecter les contraintes de capacité des véhicules et des fenêtres de temps.

- **Problème NP-complet**
  - Varrentrapp et al., 2004.

Approches existantes

Catégorie	Approche	Auteur	Méthode	Taille max des instances	Temps de calcul	Qualité de la solution
Heuristiques	Saving Functions Based Algorithm	Ferrari et Manzini, 2004	Algorithme glouton	100	15s	+ +
	Simulation Based approach	Correia et Viegas, 2008	Kmeans & CPLEX	200	500s	+ + +
Metaheuristique	Approximated Tree Search (ANTS)	Maniezzo et al., 2006	Colonie de fourmis	225	422s	+ + +

Objectifs de cette thèse

- ✓ **Définir un modèle mathématique pour le problème de covoiturage régulier.**
- ✓ **Développer des métaheuristiques qui fournissent rapidement de bonnes solutions.**
- ✓ **Développer des métaheuristiques qui résolvent efficacement de grandes instances.**
- ✓ **Développer une plate-forme de covoiturage régulier dans un contexte réel.**

Modèle mathématique

Coût d'un groupe:

cost (min_path (32, 1)) = 110

cost (min_path (29, 1)) = 120

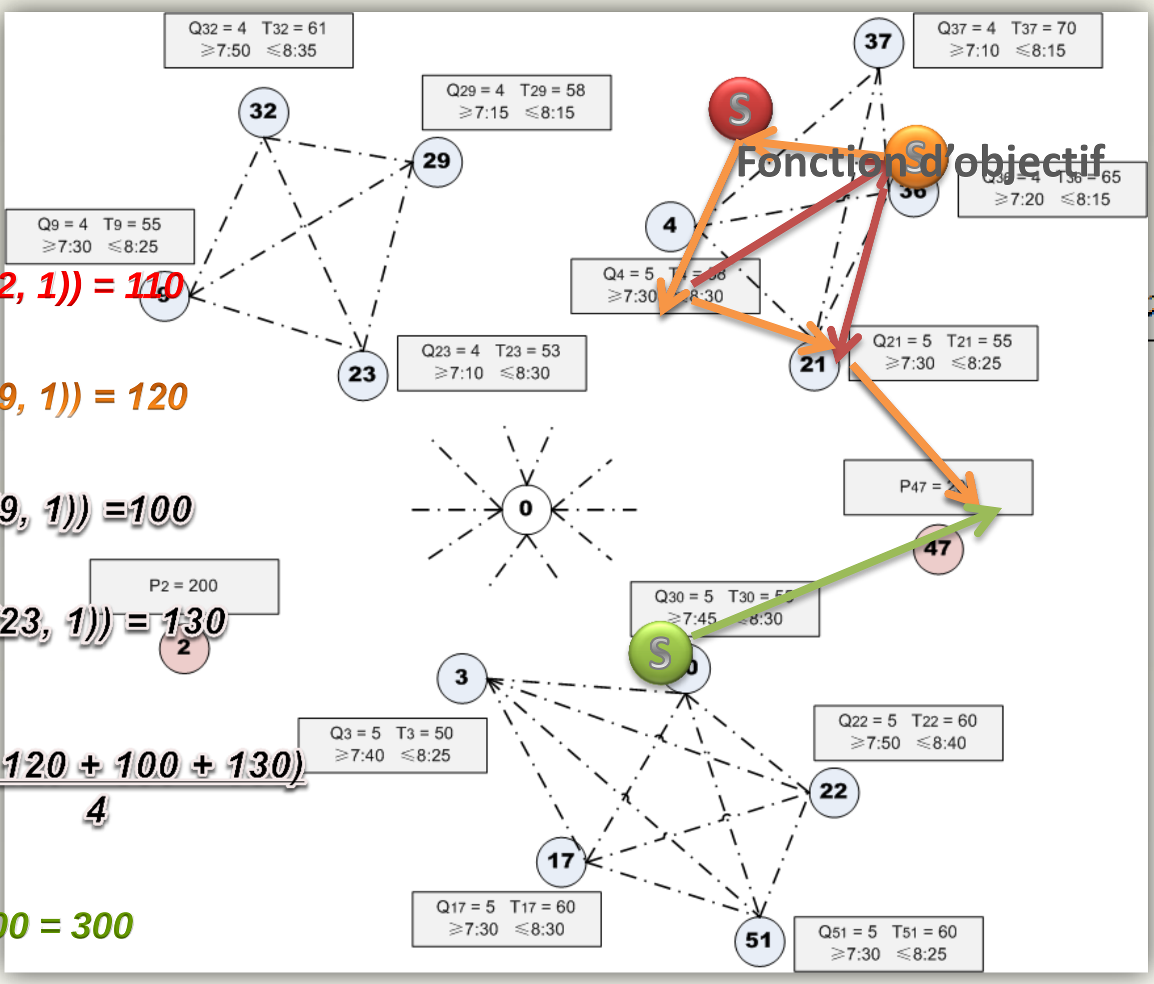
cost (min_path (9, 1)) =100

cost (min_path (23, 1)) = 130

$$cost(1) = \frac{(110 + 120 + 100 + 130)}{4}$$

$$= 115$$

cost (2) = 100 + 200 = 300



$$x_{ij}^{hk}\Big) + \sum_{i \in C} p_i \xi_i \Big)$$

Modèle mathématique

Contraintes:

$$\sum_{j \in U/\{h\}} x_{ij}^{hk} = y_{ik} \qquad i,h \in U, k \in K$$

$$\sum_{j \in U} x_{ji}^{hk} = y_{ik} \qquad i,h \in U, k \in K$$

$$\sum_{j \in U} x_{ij}^{hk} = \sum_{j \in U} x_{ji}^{hk} \qquad i,h \in U, k \in K$$

Contraintes de groupe de covoiturage

$$\sum_{k \in K} y_{ik} + \xi_i = 1 \qquad i \in U$$

Contrainte de pénalité

$$\sum_{(i,j) \in A} x_{ij}^{hk} \leq Q_h \qquad h \in U, k \in K$$

Contrainte de la capacité de véhicule

$$\sum_{(i,j) \in A} x_{ij}^{hk} t_{ij} \leq T_h \qquad h \in U, k \in K$$

$$S_i^h \geq e_i \qquad i,h \in U$$

$$S_j^h - S_i^h \geq t_{ij} - M\left(1 - \sum_{k \in K} x_{ij}^{hk}\right) \qquad (i,j) \in A, h \in U$$

$$F_i^h \geq S_i^h + t_{i0} - M\left(1 - \sum_{k \in K} x_{i0}^{hk}\right) \qquad i,h \in U$$

$$F_i^h \leq r_i + M\left(1 - \sum_{k \in K} \sum_{j \in U} x_{ij}^{hk}\right) \qquad i,h \in U$$

Contraintes de fenêtres de temps

Méthodes de résolution

Métaheuristique à base de trajectoire

Algorithme de Recherche Locale à Voisinage Variable

- **Opérateurs de voisinage différent**

Métaheuristique basée sur la population

Colonies de Fourmis

- **Mécanisme de préférence**

Algorithme Génétique Guidé

- **Nouveaux opérateurs**
- **Taux de croisement et de mutation adaptatifs**

Métaheuristique de multi-populations

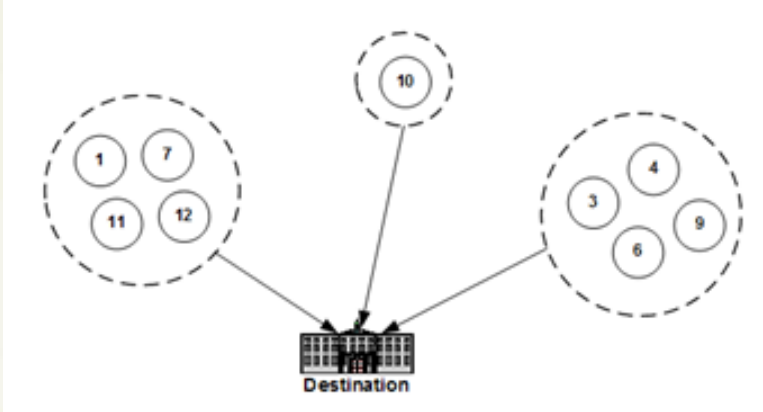
Système Multi-agents Génétiques Auto-adaptatif

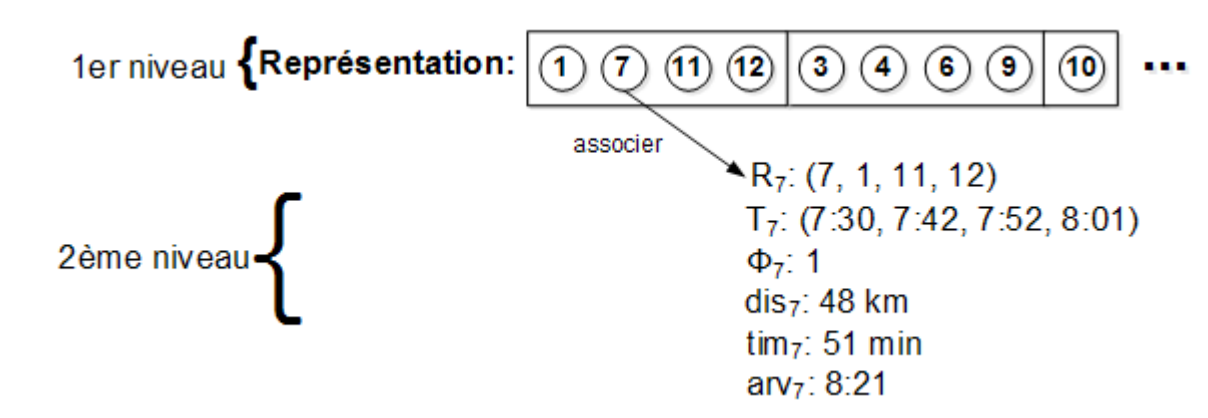
- **Système multi-agents**
- **Hyperheuristique**
- **Mécanisme d'apprentissage**

Représentation de solution

- **Représentation de solution**

  Deux niveaux:



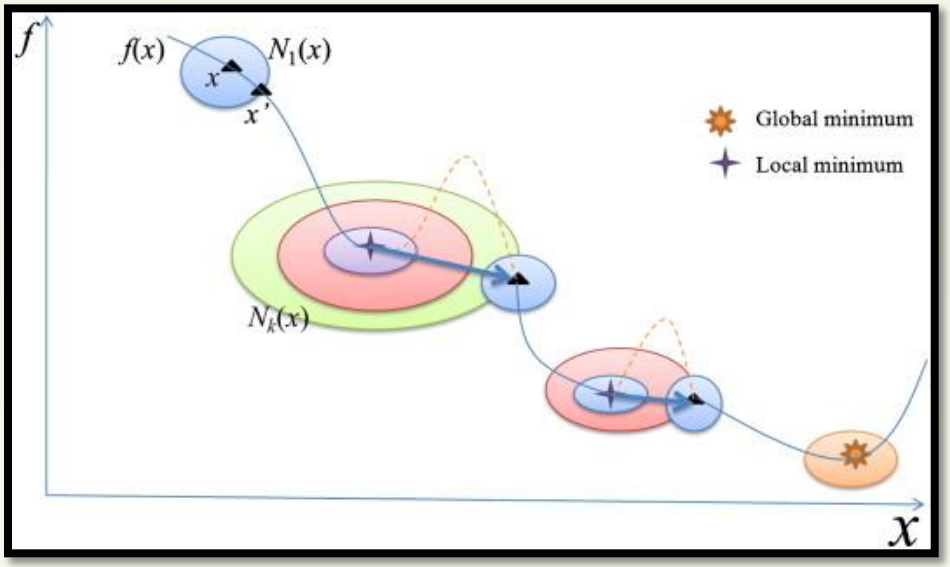


Recherche locale à voisinage variable

Métaheuristique à base de trajectoire

- **Structure de l'algorithme**

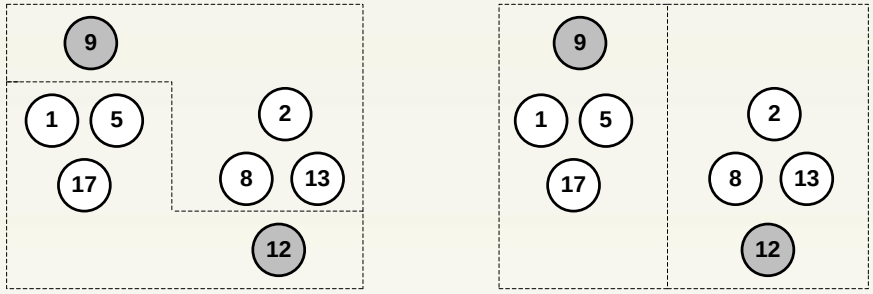
Définir un ensemble de voisinage N_k *;*
x = solution_initiale();
Tantque *non critères_d'arrêt() faire*
  k = 1;
  Tantque *k < n* ***faire***
    x' = Nk(x)
    Si *f(x') < f(x)* ***alors***
      x = x';
      k = 1;
    sinon
      k = k + 1;
    fsi
  ftq
ftq



Recherche locale à voisinage variable

Métaheuristique à base de trajectoire

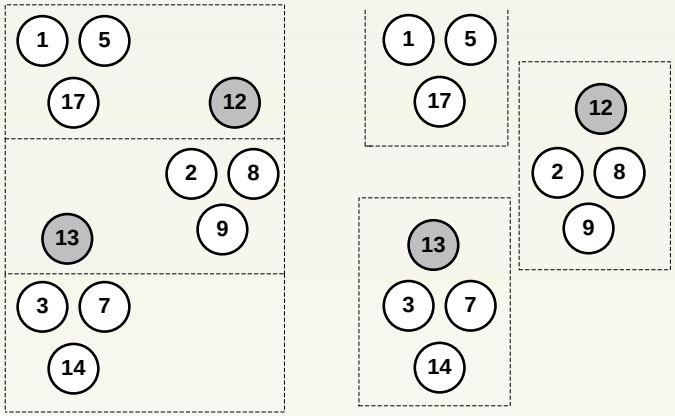
- **Conception des voisinages**
  - Opérateur Swap (N1)



(a) Avant

(b) Après

  - Opérateur Chain (N2)



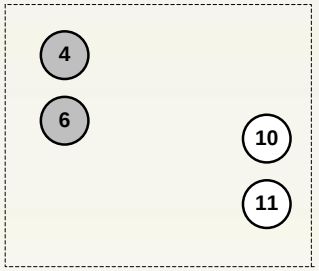
(a) Avant

(b) Après

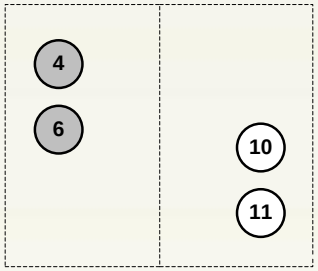
Recherche locale à voisinage variable

Métaheuristique à base de trajectoire

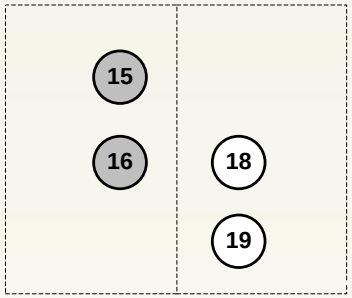
- **Conception des voisinages**
  - Opérateur Divide (N3)
  - Opérateur Merge (N4)



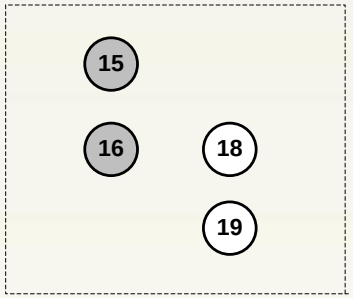
(a) Avant



(b) Après



(a) Avant



(b) Après

Résultats expérimentaux

Benchmark:

- **Le benchmark est composé de trois ensembles d'instances de structures différentes. Chaque ensemble comporte neuf instances avec un nombre d'usagers variant de 100 à 400.**
- **Les deux premiers ensembles d'instances ont été dérivés des instances du problème de PDPTW de Li et Lim [2003].**
- **Les usagers du premier ensemble (C) sont regroupés en cluster, et ceux du second ensemble (R) sont répartis aléatoirement.**
- **Le dernier ensemble d'instances (W) est obtenu avec des données réelles : les étudiants de l'université d'Artois.**

Résultats expérimentaux

Recherche locale à voisinage variable (RVV) VS Simulation Based Approach (SB)

- 9 instances par ensemble, 30 exécutions par instance, Système d'exploitation Windows avec processeur Intel Core i7 2,9 GHz et 4 Go de RAM

Instances	RVV			SB		
	Meilleure solution trouvée	Qualité moyenne de solution	Temps de calcul moyen	Meilleure solution trouvée	Qualité moyenne de solution	Temps de calcul moyen
Ensemble C	7	6	95s	2	3	402s
Ensemble R	5	7	105s	4	2	501s
Ensemble W	7	7	171s	2	2	526s

- Pour tous les ensembles, la Recherche à Voisinage Variable (RVV) :
  - Donne des solutions de meilleure qualité (Amélioration moyenne de 4.3% par rapport à SB).
  - Plus rapide

Recherche locale à voisinage variable

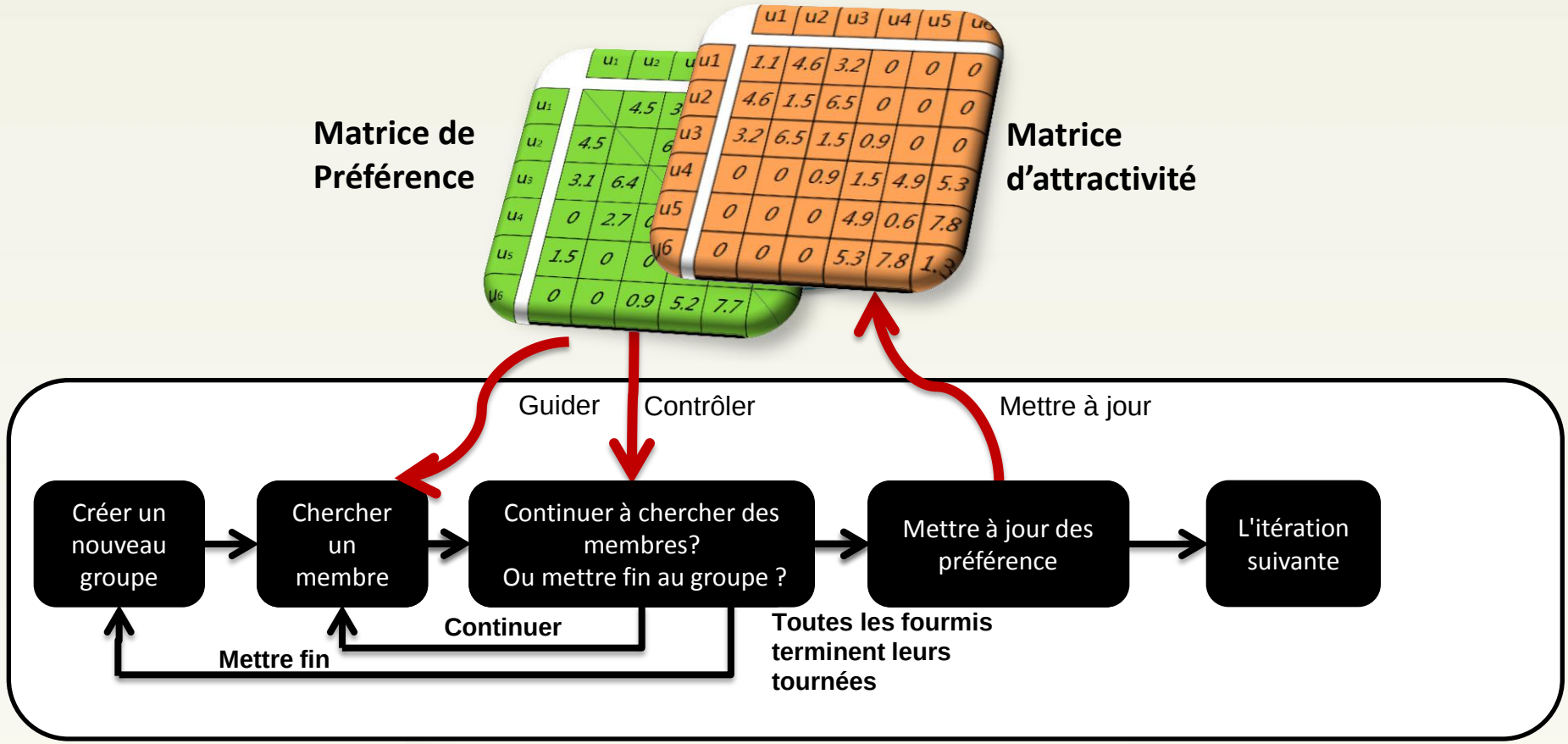
Métaheuristique à base de trajectoire

- La caractéristique principale de cette approche consiste en sa capacité de passer d'un voisinage à un autre tout au long du processus d'optimisation;
- Utilisation de plusieurs opérateurs a permis d'améliorer la capacité de recherche
- Vitesse de calcul rapide;
- Facile à mettre en œuvre;
- Algorithme adapté pour l'intensification mais a peu de capacité pour la diversification.

Les métaheuristiques basées sur la population.

Algorithme de colonies de fourmis

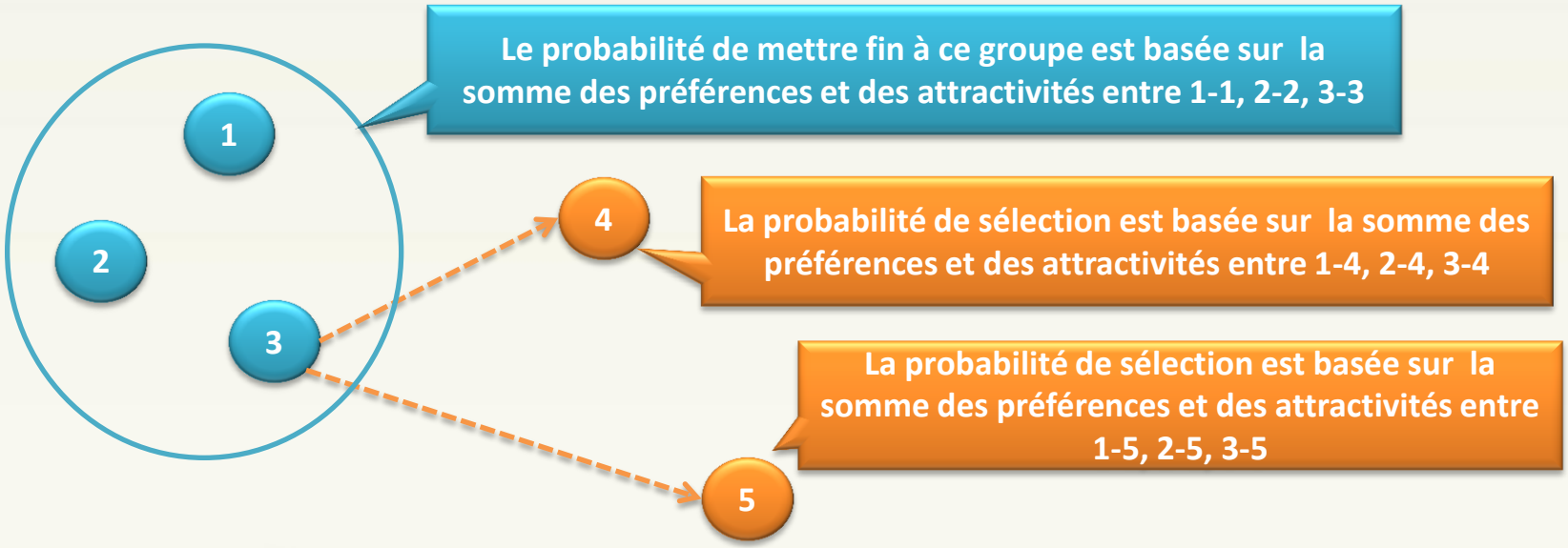
Métaheuristique à base de population



Algorithme de colonies de fourmis

Métaheuristique à base de population

- Préférence: la tendance de mettre deux usagers dans le même groupe;
- Sélection de mouvement de fourmis:

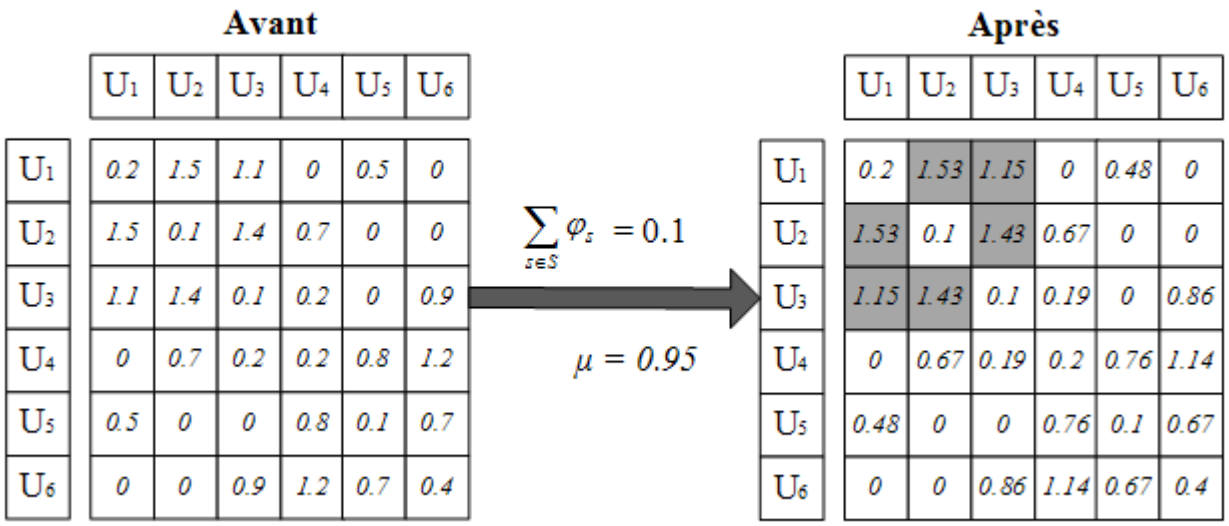


Algorithme de colonies de fourmis

Métaheuristique à base de population

- Initiale:

- Mise à jour:



	U_1	U_2	U_3	U_4	U_5	U_6
U_1	0.2	1.5	1.1	0	0.5	0
U_2	1.5	0.1	1.4	0.7	0	0
U_3	1.1	1.4	0.1	0.2	0	0.9
U_4	0	0.7	0.2	0.2	0.8	1.2
U_5	0.5	0	0	0.8	0.1	0.7
U_6	0	0	0.9	1.2	0.7	0.4

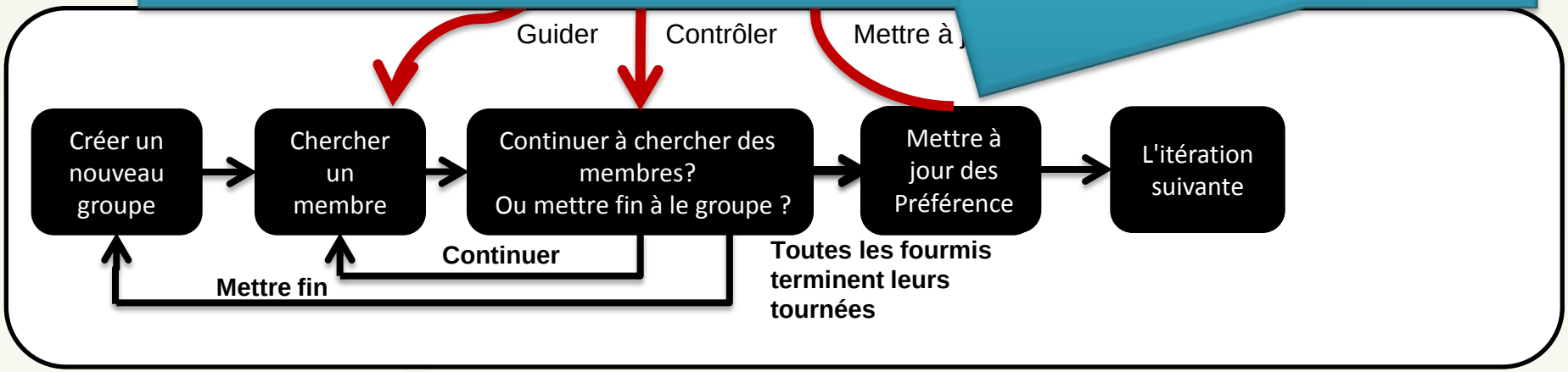
	U_1	U_2	U_3	U_4	U_5	U_6
U_1	0.2	1.53	1.15	0	0.48	0
U_2	1.53	0.1	1.43	0.67	0	0
U_3	1.15	1.43	0.1	0.19	0	0.86
U_4	0	0.67	0.19	0.2	0.76	1.14
U_5	0.48	0	0	0.76	0.1	0.67
U_6	0	0	0.86	1.14	0.67	0.4

Algorithme de colonies de fourmis

Métaheuristique à base de population

- **Recherche locale**
  - Divide Opérateur
  - Merge Opérateur
  - Swap Opérateur
  - Move Opérateur
- **Liste Interdite:**

Target pool 1	Target pool 2	Operator
12	—	Divide
3	4	Swap
17	22	Move
…	…	…



Résultats expérimentaux

Algorithme à base de colonies de fourmis (ARCF) VS ANTS et SB

Instances	ARCF			ANTS			SB		
	Meilleure solution trouvée	Qualité moyenne de solution	Temps de calcul moyen	Meilleure solution trouvée	Qualité moyenne de solution	Temps de calcul moyen	Meilleure solution trouvée	Qualité moyenne de solution	Temps de calcul moyen
Ensemble C	8	7	92s	1	1	168s	0	1	482s
Ensemble R	4	6	111s	2	1	228s	3	2	501s
Ensemble W	7	8	148s	0	0	218s	2	1	526s

- La qualité des solutions du ARCF est significativement meilleure que celles des autres approches sur les ensembles d'instances C et W.
- Toutefois, la capacité de recherche sur l'ensemble R doit être améliorée.
- La recherche locale peut améliorer la qualité de la solution de 2% à 11% selon la taille de l'instance.
- Cette approche améliore la qualité de la solution en moyenne de 8.4% par rapport à SB et de 4.2% par rapport à ANTS.

* 30 exécutions pour chaque instance, Système d'exploitation Windows avec processeur Intel Core i7 2,9 GHz et 4 Go de RAM

Algorithme de colonies de fourmis

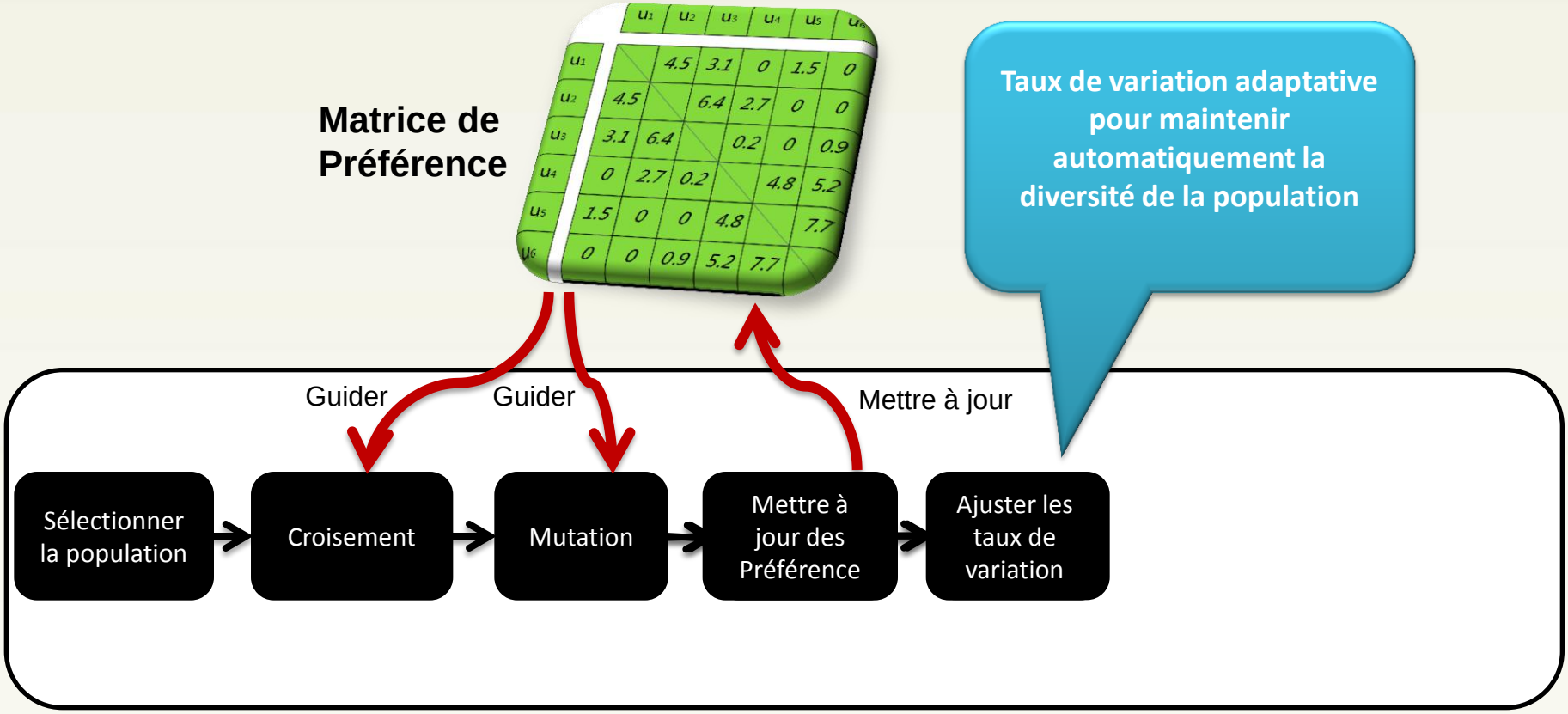
Métaheuristique à base de population

- Il peut fournir une bonne qualité de solution pour les instances C (distribués en cluster);
- Il peut être facilement adapté à d'autres problèmes de regroupement;
- La qualité de la solution des instances avec des usagers distribués aléatoirement peut être encore améliorée par d'autres métaheuristiques.

→ Algorithme génétique guidé

Algorithme génétique guidé

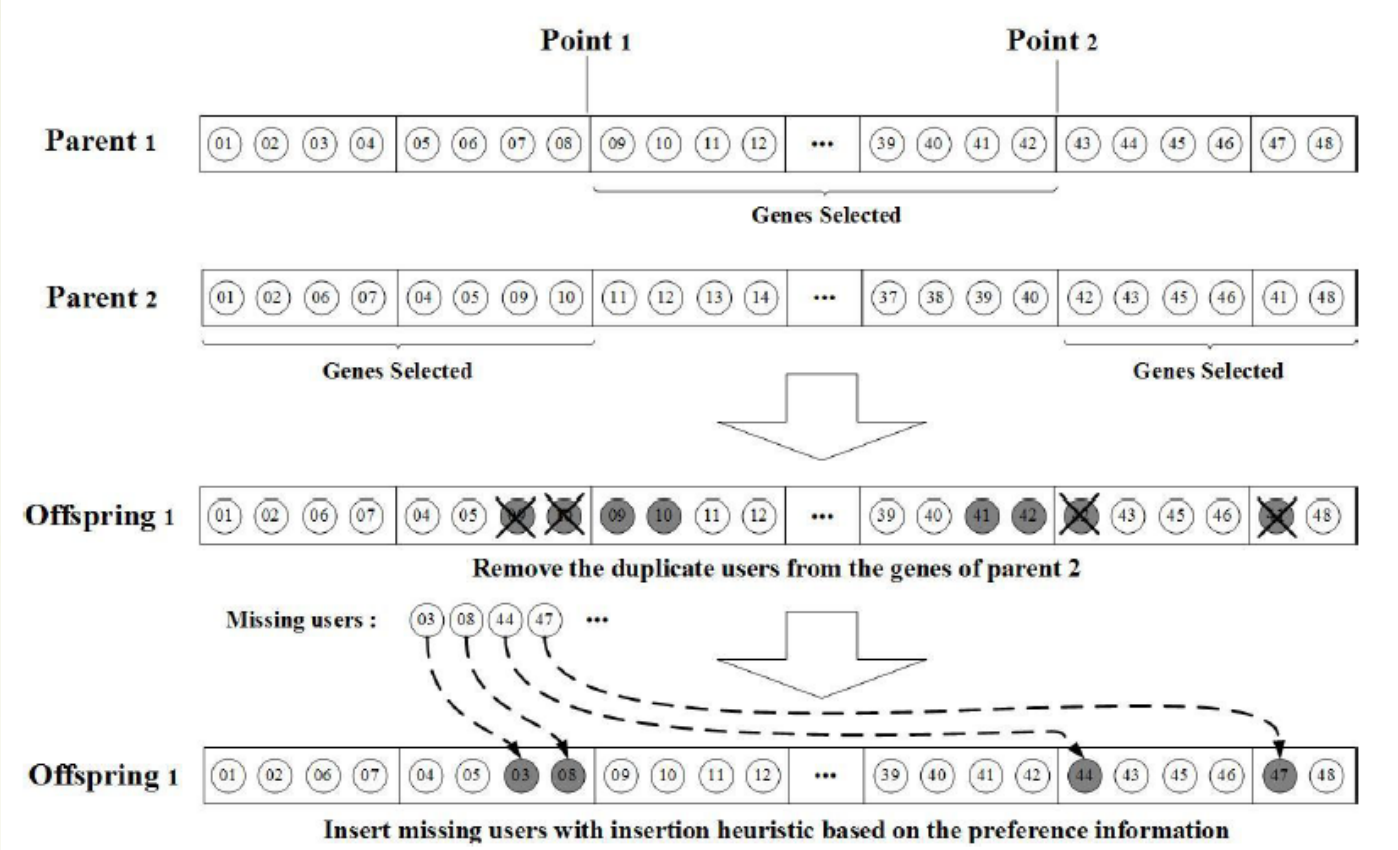
Métaheuristique à base de population



Algorithme génétique guidé

Métaheuristique à base de population

- **Croisement** :



Algorithme génétique guidé

Métaheuristique à base de population

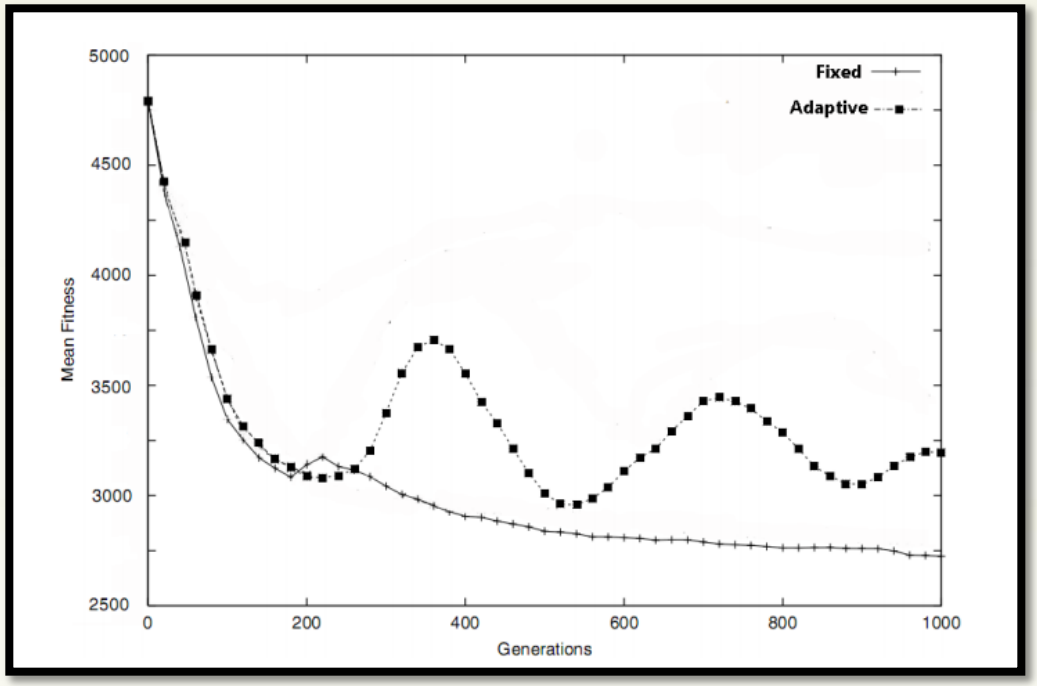
- **Mutation:**
  - ✓ Opérateur Divide
  - ✓ Opérateur Merge
  - ✓ Opérateur Swap
  - ✓ Opérateur Reinsert
- Chaque opérateur est utilisé avec une probabilité égale (25%).

Algorithme génétique guidé

Métaheuristique à base de population

- Taux de variation adaptatif.
- Évaluation de la diversité en mesurant les fitness moyens des parties différentes de la population.
- Mise à jour des taux.

$$v = \begin{cases} v' \times (1+\eta) & if \quad \dfrac{f^{w}_{avg} - f^{b}_{avg}}{f^{w}_{avg}} \leq \rho \\ v' \times (1-\eta) & otherwise \end{cases}$$



Résultats expérimentaux

Algorithme génétique guidé VS ARCF, ANTS et SB

Instances	AGG			ARCF			ANTS		
	Meilleure solution trouvée	Qualité moyenne de solution	Temps de calcul moyen	Meilleure solution trouvée	Qualité moyenne de solution	Temps de calcul moyen	Meilleure solution trouvée	Qualité moyenne de solution	Temps de calcul moyen
Ensemble C	5	2	98s	3	6	92s	1	1	168s
Ensemble R	7	8	117s	0	0	111s	2	1	228s
Ensemble W	4	6	152s	5	3	148s	0	0	218s

- La qualité des solutions de AGG est meilleure que les autres approches en traitant les instances d'ensemble R.
- Toutefois, l'ARCF est encore une meilleure approche pour les instances distribués en cluster.
- Le taux de variation adaptatif peut améliorer la qualité de la solution en moyenne de 5%.
- Cette approche améliore la qualité de la solution en moyenne de 1.2% par rapport à ARCF et de 5.2% par rapport à ANTS.

* 30 exécutions pour chaque instance, Système d'exploitation Windows avec processeur Intel Core i7 2,9 GHz et 4 Go de RAM

Algorithme génétique guidé

Métaheuristique à base de population

- Les informations de préférences améliorent la qualité de la population;
- Le mécanisme d'apprentissage augmente la qualité de la recherche;
- Les taux de variation adaptative permettent de maintenir la diversité de la population;
- Une bonne qualité de solution pour les instances distribuées aléatoirement.

Inconvénients

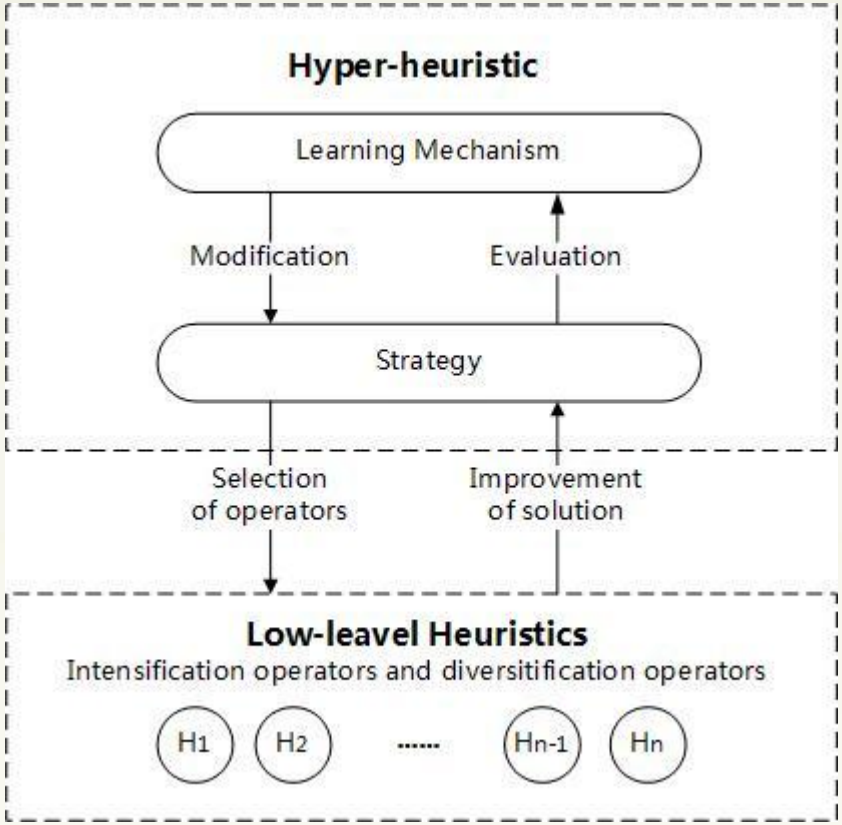
Métaheuristique à base de population

- Les métaheuristiques à base de population nécessitent beaucoup de temps pour les instances de grande taille.
- Ces algorithmes nécessitent un réglage difficile et précis d'un grand nombre de paramètres afin d'obtenir une bonne qualité de recherche.
- Les structures de ces algorithmes sont fixes, donc quand un nouvel opérateur est développé, la structure du système entier de l'algorithme doit être actualisée.

Système multi-agents génétiques auto-adaptatif

Métaheuristique à base de multi-population

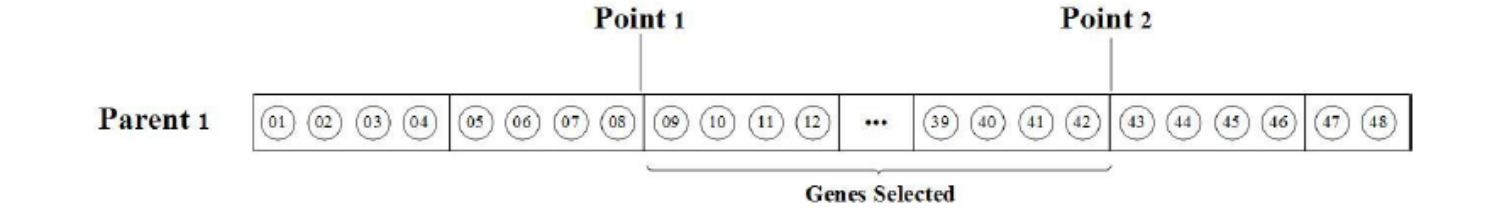
- Combinaison de système multi-agents et l'algorithme génétique, avec l'aide d'hyperheuristique; Chaque agent génétique gère une population différente.
- Sélectionner automatiquement et appliquer les opérateurs appropriés au cours du processus d'optimisation;
- Mécanisme non supervisé : utiliser les expériences précédentes pour améliorer les choix dans les itérations futures.



Système multi-agents génétiques auto-adaptatif

Métaheuristique à base de multi-population

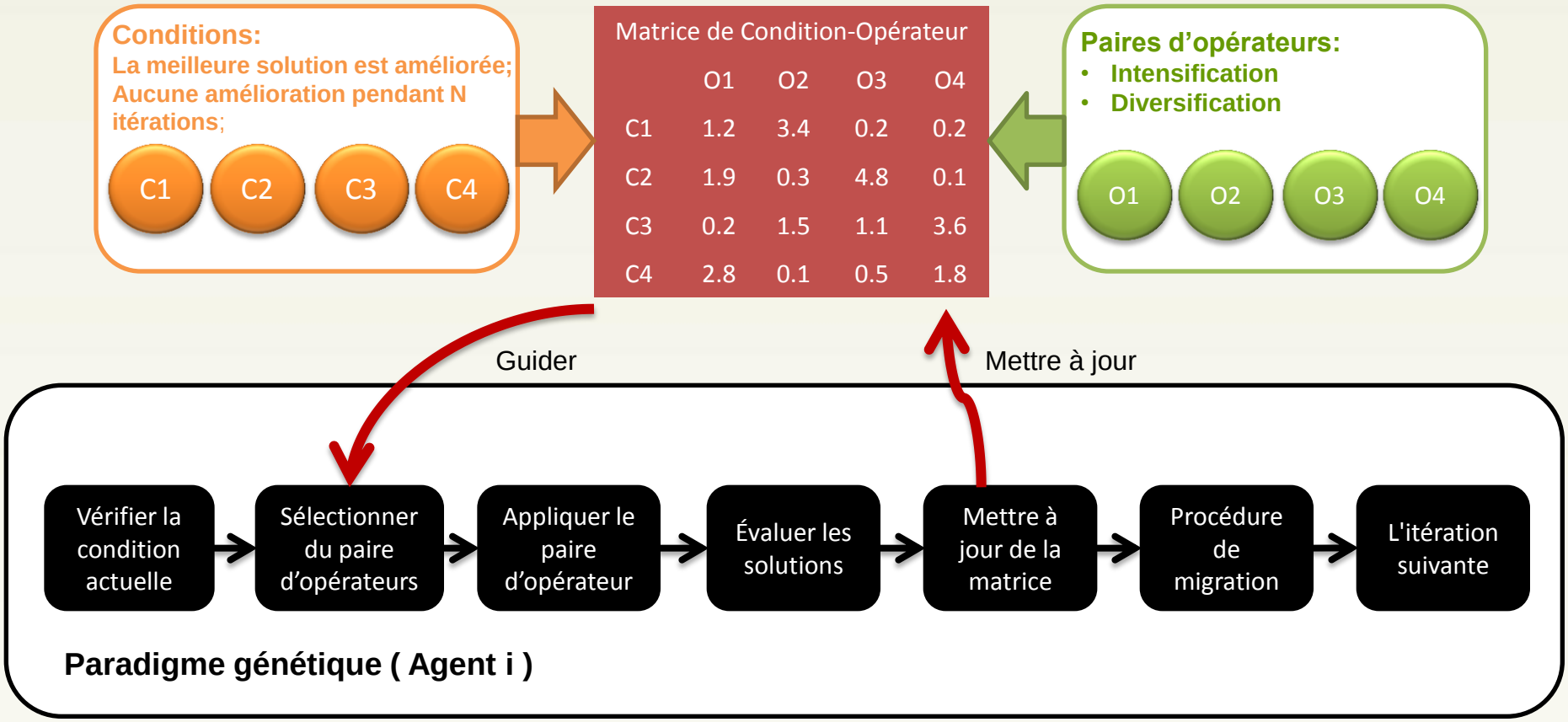
- **Paires d'opérateurs**
  - Opérateurs de croisement
    - 4 operateurs de croisement à 2 points : en fonction de la distance entre ces 2 points



  - Opérateurs de mutation
    - Opérateurs de mutation M1 et M2 (intensification)
    - Opérateurs de mutation M3 et M4 (diversification)

Système multi-agents génétiques auto-adaptatif

Métaheuristique à base de multi-population



Système multi-agents génétiques auto-adaptatif

- Sous la condition i, si on applique l'opérateur j et obtient une amélioration, on met à jour la matrice avec un facteur σ. Le facteur σ est calculé comme suit:

$$\sigma = \alpha \times \frac{Localbest_{current} - Localbest_{new}}{Localbest_{current}} + \beta \times \frac{Globalbest_{current} - Globalbest_{new}}{Globalbest_{current}}$$

Par exemple sous la condition 1, si on applique l'opérateur 3, et que la meilleure solution globale est améliorée, alors le facteur d'amélioration est calculé comme suit:

$$\sigma = 2 \times \frac{3750 - 3400}{3750} + 5 \times \frac{3500 - 3400}{3400} = 0.187 + 0.143 = 0.33$$

0.33

Matrice initiale à jour:

	Opt1	Opt2	Opt3	Opt4
C1	1	1	1.33	1
C2	1	1	1	1
C3	1	1.25	1	1
C4	1	1	1	1.19

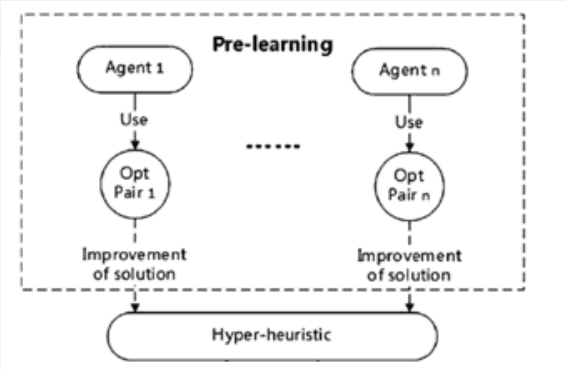
0. 25

0. 19

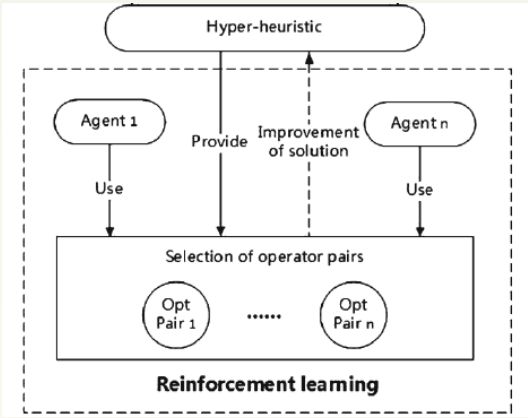
Système multi-agents génétiques auto-adaptatif

Métaheuristique à base de multi-population

- **Pré-apprentissage:**
  - ✓ Les agents utilisent les paires d'opérateurs fixes;
  - ✓ Évaluer chaque paire d'opérateurs;
  - ✓ offrir des expériences initiales des performances de chaque paire d'opérateurs dans les conditions différentes;



- **Apprentissage de renforcement:**
  - ✓ Mécanisme de prise de décision est auto-adaptative;
  - ✓ Modifier les règles de prise de décision en fonction du niveau de l'amélioration;
  - ✓ Favoriser la paire d'opérateurs qui améliore souvent la meilleure solution dans chaque condition.



Résultats expérimentaux

Système multi-agents génétiques auto-adaptatif VS AGG, ARCF et ANTS

I	AMGA			AGG			ARCF			ANTS		
	Me	Mo	T	Me	Mo	T	Me	Mo	T	Me	Mo	T
C	6	9	34s	1	0	98s	1	0	92s	1	0	168s
R	7	9	48s	1	0	117s	0	0	111s	1	0	228s
W	7	9	52s	1	0	152s	1	0	148s	0	0	218s

- Lla qualité de solution de l'AMGA est significativement meilleure que les autres approches à la fois dans la qualité de la meilleure solution trouvée et la qualité de la solution moyenne.
- En outre, compte tenu du temps de calcul total de toutes les instances, l'AMGA est le plus rapide.
- Cette approche améliore la qualité de la solution de AGG, ARCF et ANTS en moyenne respective de 2.6%, 3.6% et 7.5%.

* 30 exécutions pour chaque instance, Système d'exploitation Windows avec processeur Intel Core i7 2,9 GHz et 4 Go de RAM

Système multi-agents génétiques auto-adaptatif

Métaheuristique à base de multi-population

- Approche du problème non focalisée, peut être facilement adaptée à d'autres problèmes;
- Méthode auto-adaptative, le réglage des paramètres est plus souple;
- De nouveaux opérateurs peuvent facilement être introduits sans modifier la structure de l'algorithme;
- Solutions de bonne qualité pour les instances de grande taille;
- Exécution en parallèle augmente la vitesse du processus;
- Migration entre les agents maintient la diversité de la population.

Plate-formes pour le problème de covoiturage

Plate-forme java de test et d'analyse

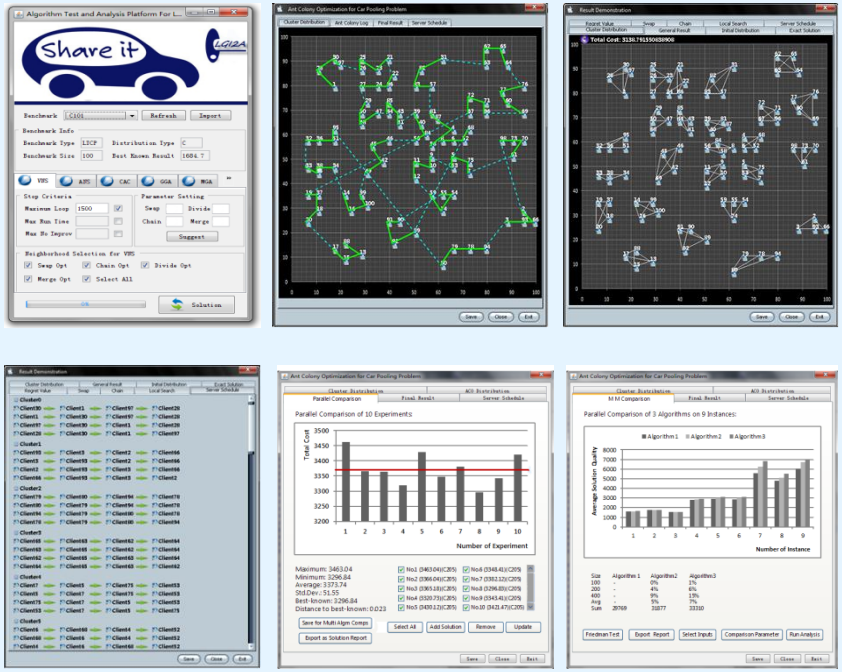
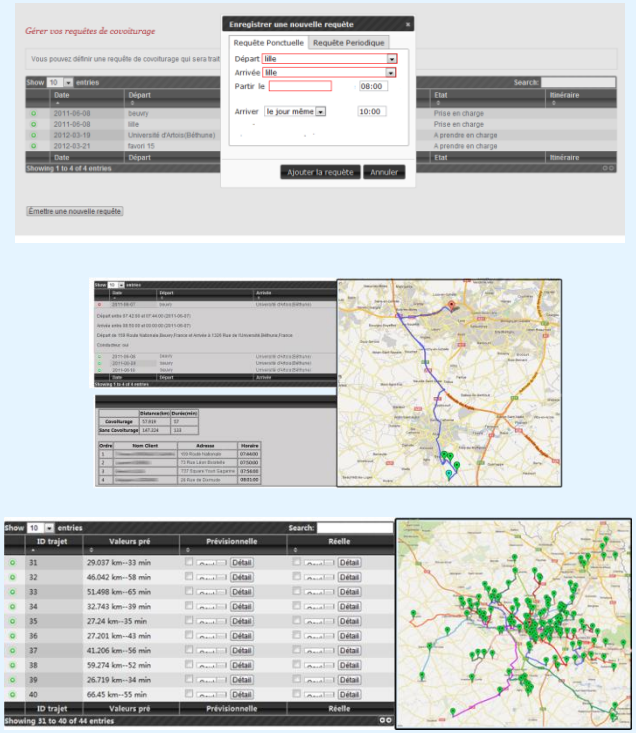


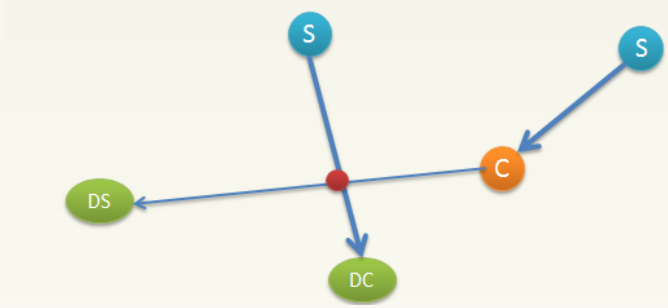
Plate-forme de covoiturage

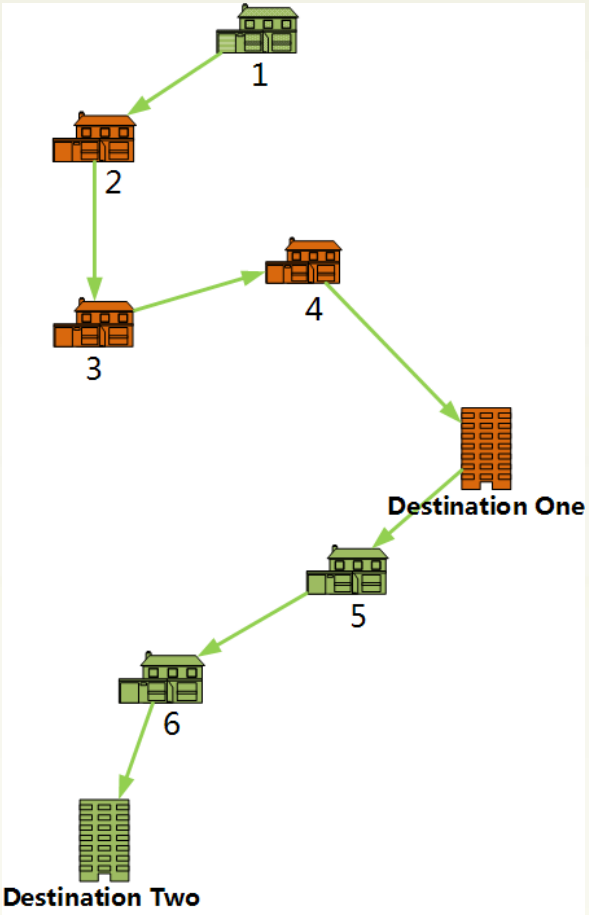


Covoiturage occasionel comportant plusieurs destinations

Extension

- Les usagers peut avoir plusieurs destinations.
- Deux serveurs peuvent avoir un point de transfert où leurs clients peuvent changer les véhicules.
- Un nouveau modèle mathématique intégrant les points de transfert.
- Une méthode de résolution efficace à base de colonie de fourmis .





Contributions

- Un état de l'art sur le problème de covoiturage .
- Trois ensembles d'instances structurellement différents pour le problème de covoiturage régulier.
- Une recherche à voisinage variable qui est efficace et facile à mettre en œuvre.
- Un algorithme de colonie de fourmis avec haute performance pour le problème de covoiturage régulier.
- Un algorithme génétique performant intégrant un mécanisme de contrôle adaptatif et un mécanisme directif.
- Un algorithme efficace de multi-agents avec un mécanisme d'apprentissage.
- Une plate-forme pour le test, la démonstration, l'évaluation et la comparaison des approches pour résoudre le problème de covoiturage régulier.
- Une plate-forme Web pour le covoiturage des étudiants.
- Proposer un nouveau type de problème de covoiturage occasionnel comportant plusieurs destinations,
  - Modèle mathématique
  - Méthode de résolution efficace.

Perspectives

- Certains algorithmes proposés peuvent être améliore’s.
  - recherche à voisinage variable avec multi-démarrages
  - Taux d'évaporation adaptatif pour l’algorithme de colonie de fourmis
- L'efficacité des algorithmes développés encourage leur application à d'autres problèmes de transport.
  - Problème de covoiturage occasionnel
  - Problème de transport public
- Le problème de covoiturage comportant plusieurs destinations peut être encore étudié ainsi que sa méthode de résolution.
  - Intégrer l’heuristique de recherche de points de transfert dans l'algorithme principal
  - Mise en œuvre des points de transfert dans monde réel
- Les deux plates-formes conçues peuvent encore être améliorées.
  - Optimisation de la programmation
  - Applications mobiles

Publications

- **Journals**
  - **1.** Guo Y., Goncalves G., Hsu T. Multi-agent based self-adaptive genetic algorithm for the long-term car pooling problem. International Journal of Mathematical Modeling and Algorithms. Springer, 2012.
  - **2.** Guo Y., Goncalves G., Hsu T. A clustering ant colony algorithm for the long-term car pooling problem. International Journal of Swarm Intelligence. IGI Global, 2012.
  - **3.** Guo Y., Goncalves G., Hsu T. The multi-destination car pooling problem and its resolution method. RAIRO Operations Research. Cambridge, 2013. (Submitted)
  - **4.** Guo Y., Goncalves G., Hsu T. The guided genetic algorithm for solving the long-term car pooling problem. Metaheuristic Research, 2013. (Submitted)
- **Conferences**
  - **5.** Guo Y., Goncalves G., Hsu T. A velocity based bee colony algorithm for the car pooling problem. European Conference on Metaheuristics. Lorient, France, 2010.
  - **6.** Guo Y., Goncalves G., Hsu T. A self-adaptive genetic algorithm for the car pooling problem. In Proceeding of International Conference on Metaheuristics and Nature Inspired Computing. Djerba, Tunisia, 2010.
  - **7.** Guo Y., Goncalves G., Hsu T. Genetic algorithm with preference matrix for the car pooling problem. 12e Congrès Annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision. Saint Etienne, France, 2011.
  - **8.** Guo Y., Goncalves G., Hsu T. A guided genetic algorithm for solving the long-term car pooling problem. In Proceeding of IEEE International Conference on Computational Intelligence in Production and Logistics Systems. Paris, France, 2011.
  - **9.** Guo Y., Goncalves G., Hsu T. A clustering ant colony algorithm for the long-term car pooling problem. In Proceeding of IEEE International Conference on Swarm Intelligence. Cergy, France, 2011.
  - **10.** Guo Y., Goncalves G., Hsu T. An ant colony algorithm based hybrid approach for solving the multi-destination car pooling problem. 13e Congrès Annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision. Angers, France, 2012.

Université Lille Nord de France
Pôle de Recherche et d'Enseignement Supérieur

UNIVERSITÉ D'ARTOIS

LGI2A

Métaheuristiques pour la Résolution de Problème de Covoiturage Régulier de Grande Taille et d'une Extension

Yuhan GUO

Composition du jury

Rapporteurs:	M. Van-Dat CUNG	Professeur, Institut Polytechnique de Grenoble
	M. Jin-Kao HAO	Professeur, Université d'Angers
Examinateurs:	Mme. Marie-José HUGUET	HDR, MCF, Institut National des Sciences Appliquées de Toulouse
	M. El-Ghazali TALBI	Professeur, Université des Sciences et Technologies de Lille
Directeur de thèse:	M. Gilles GONCALVES	Professeur, Université d'Artois
Co-encadreur:	M. Tienté HSU	MCF, Université d'Artois