PRINCIPES DE BASE ET COMPARAISON DE MÉTA-HEURISTIQUES

Éric Taillard

Institut d'informatique appliquée, EIVD, Yverdon

http://ina.eivd.ch/taillard

Journées franciliennes de recherche opérationnelle, Paris, 24 janvier 2003

PLAN

INTRODUCTION

Considérations générales sur les heuristiques et les méta-heuristiques

PROGRAMMATION À MÉMOIRE ADAPTATIVE

Méta-heuristiques à mémoire

Recherche avec tabous, colonies de fourmis

Algorithmes génétiques, recherche par dispersion, construction de vocabulaire

Exemples d'applications

POPMUSIC

Méthodes basées sur l'optimisation de sous-problèmes

MIMAUSA, Grands voisinages,

Exemples d'applications

Problème de la p-médiane

Tournées de véhicules

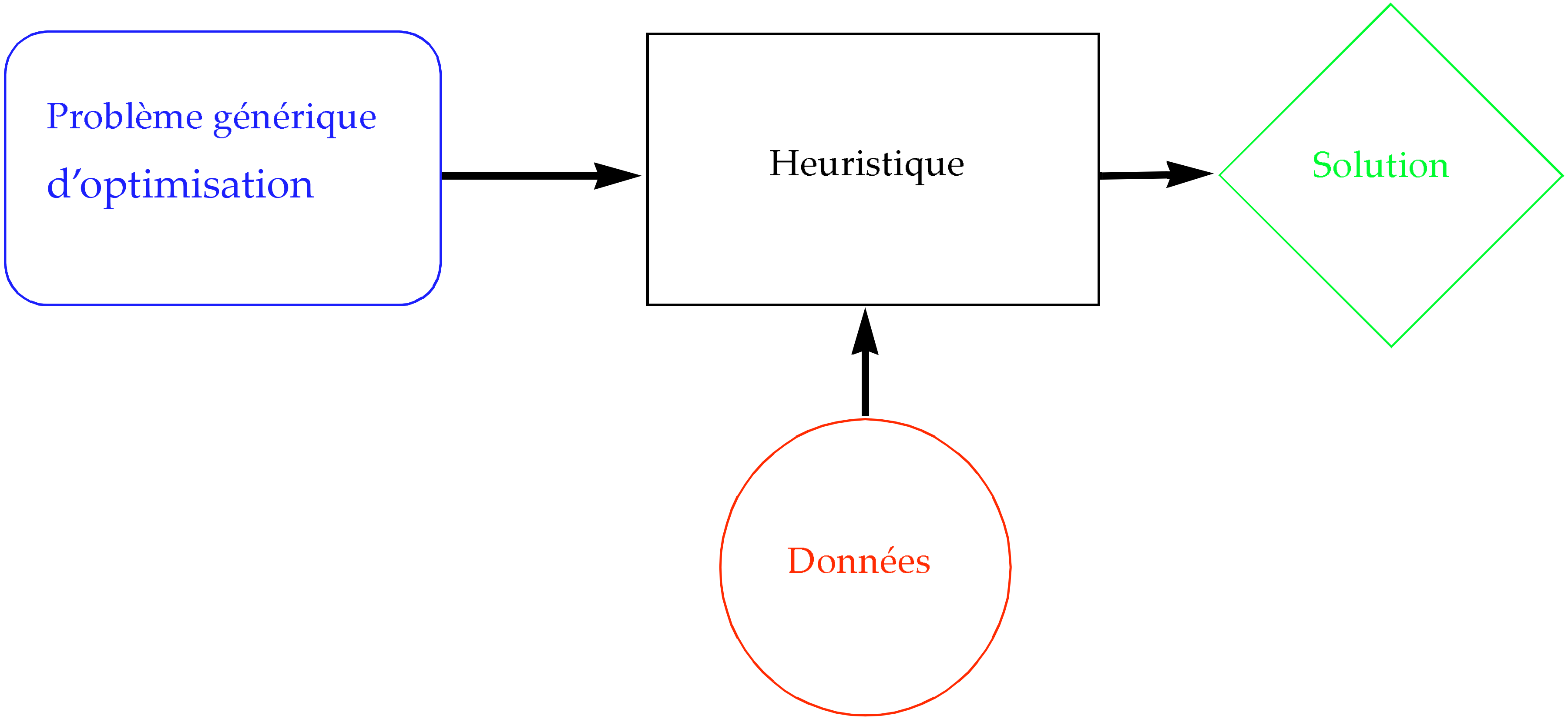
COMPARAISON DE MÉTHODES ITÉRATIVES NON DÉTERMINISTES

Problème binaire (trouvé — pas trouvé)

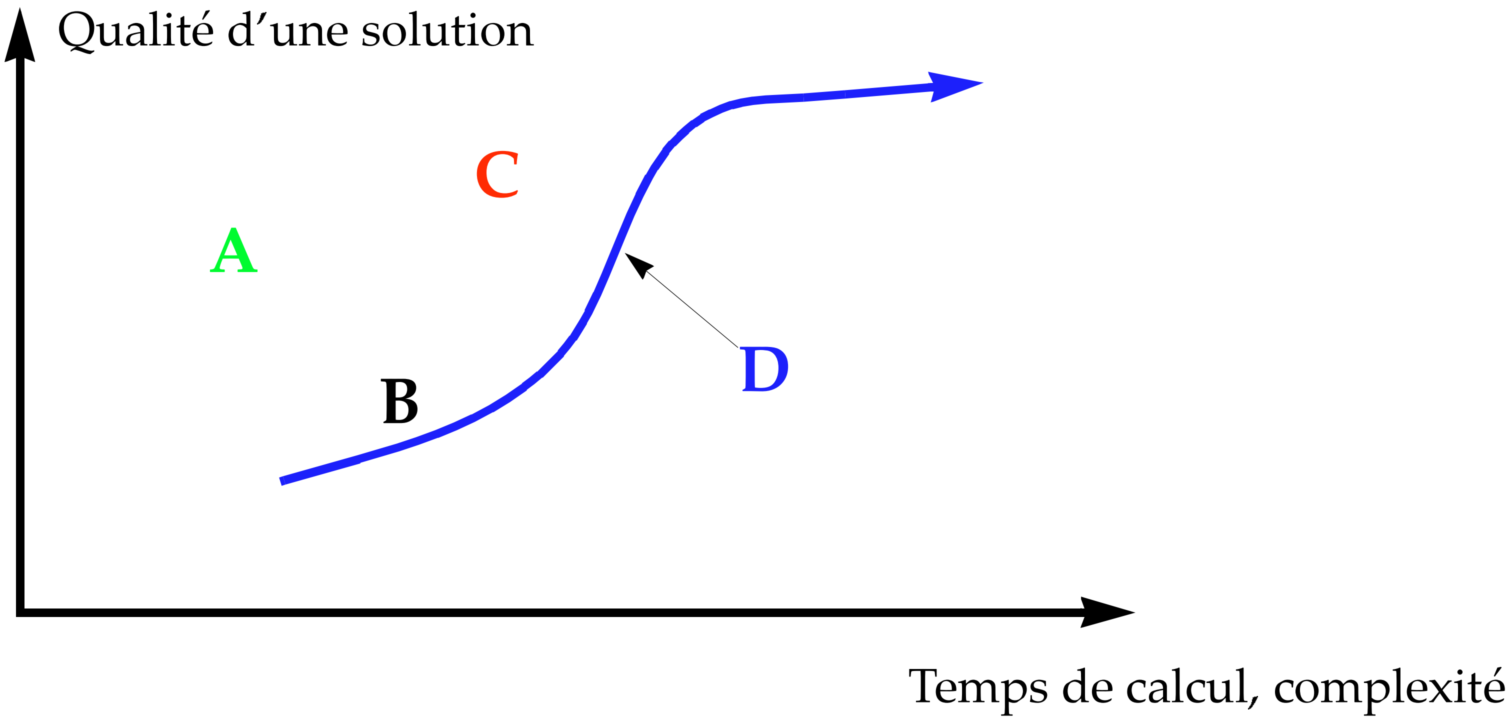
Problème d'optimisation

HEURISTIQUE :

Méthode de calcul pour un problème générique d'optimisation produisant une solution non nécessairement optimale



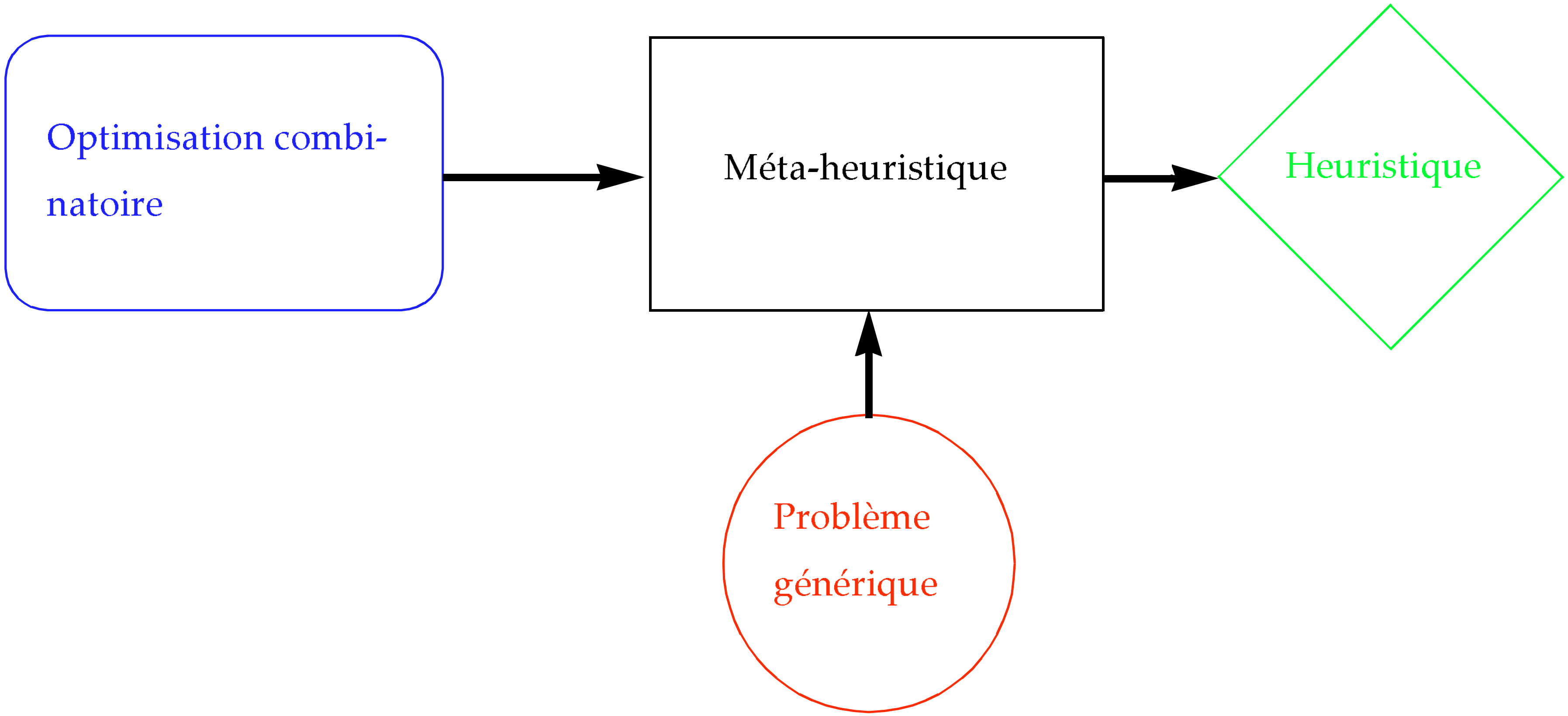
QUALITÉ D'UNE HEURISTIQUE



Autres dimensions: Données, robustesse, place en mémoire, paramètres

MÉTA-HEURISTIQUE :

Ensemble limité de concepts applicables à un large ensemble de problèmes d'optimisation combinatoire pour créer de nouvelles heuristiques



QUALITÉ D'UNE MÉTA-HEURISTIQUE

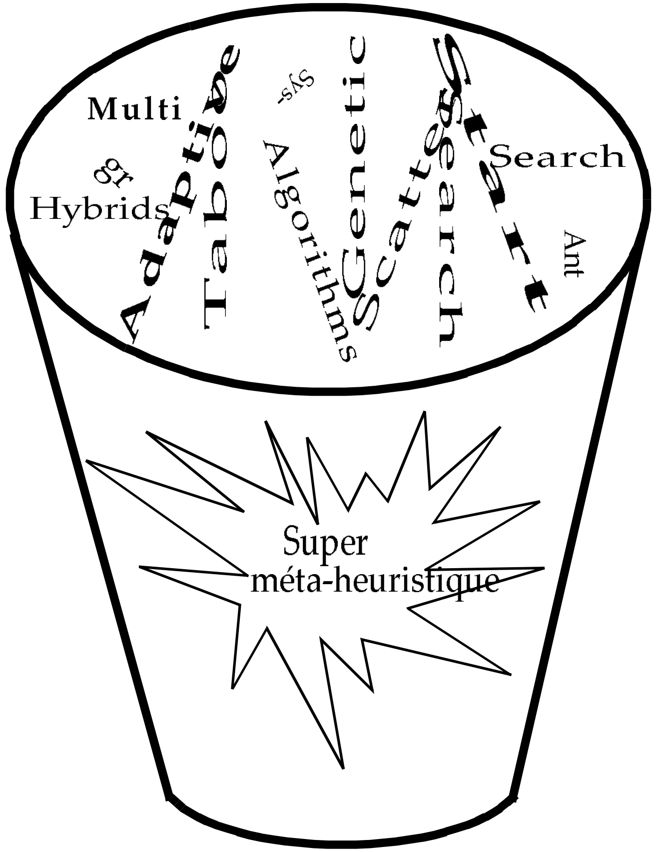
Applicabilité à un large ensemble de problèmes génériques

Puissance de suggestion de bonnes heuristiques.

Simplicité de conception

Attractivité du nom ou de la métaphore

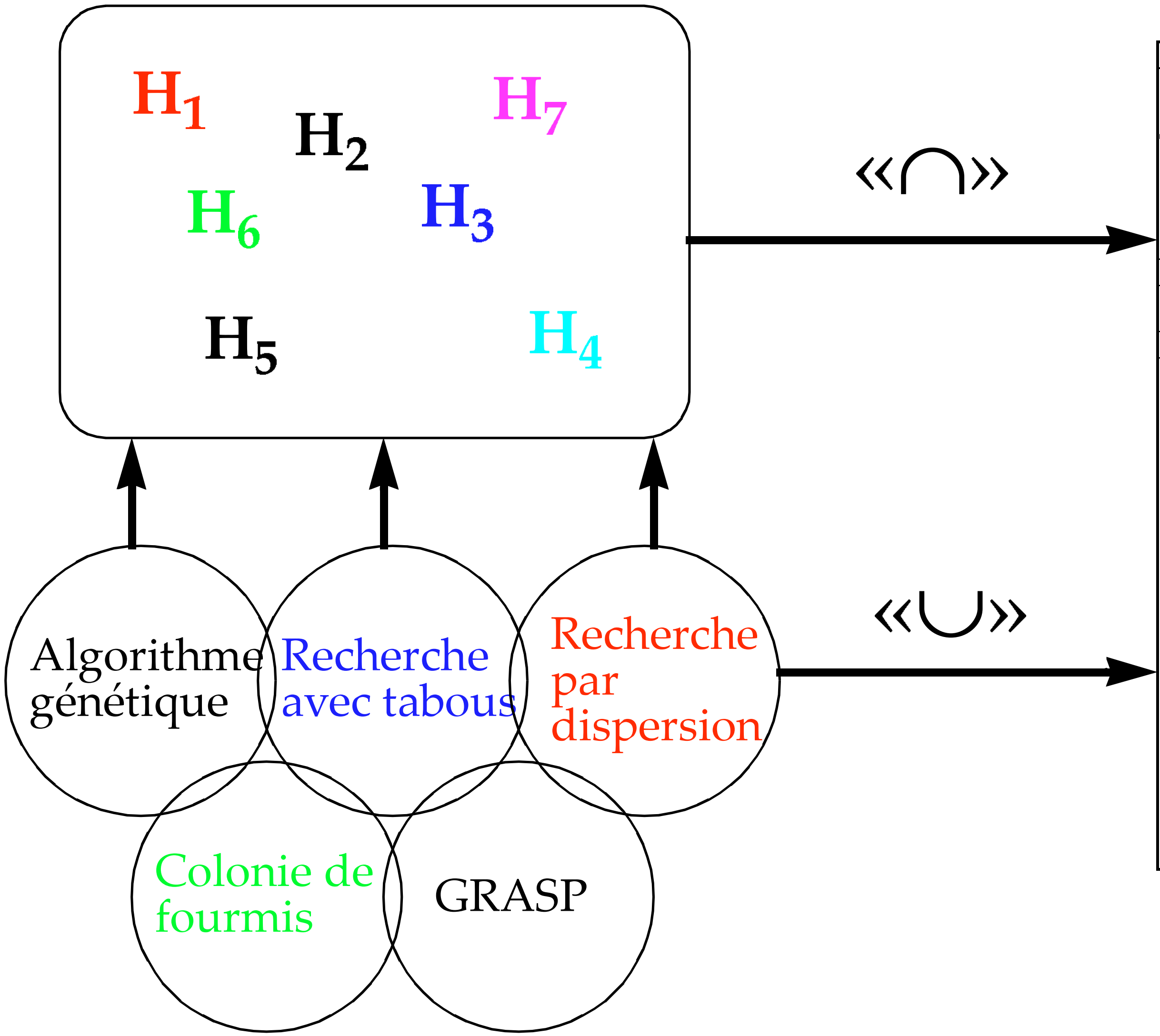
→ **Qualité d'heuristiques produites**



HEURISTIQUES — MÉTA-HEURISTIQUES

	Heuristique	Meta-heuristique
Domaine d'application	Problème d'optimisation générique	Optimisation combinatoire
Données	Exemple de problème	Problème d'optimisation générique
Coeur	Algorithme	Ensemble de principes
Résultat	Solution approchée	Algorithme

PROGRAMMATION À MÉMOIRE ADAPTATIVE



Programmation à mémoire adaptative

Initialiser la mémoire

Répéter

  Construire une solution en s'aidant de la mémoire

  Améliorer la solution avec une recherche locale

  Mettre à jour la mémoire

PAM I: RECHERCHE AVEC TABOUS

Idée de base (Glover, 1986) : recherche locale dirigée avec des mémoires.

- Mémoire à court terme (liste de tabous)
- Mémoire à long terme (fréquence, solutions élites)

Oscillation stratégiques : alterner intensification et diversification

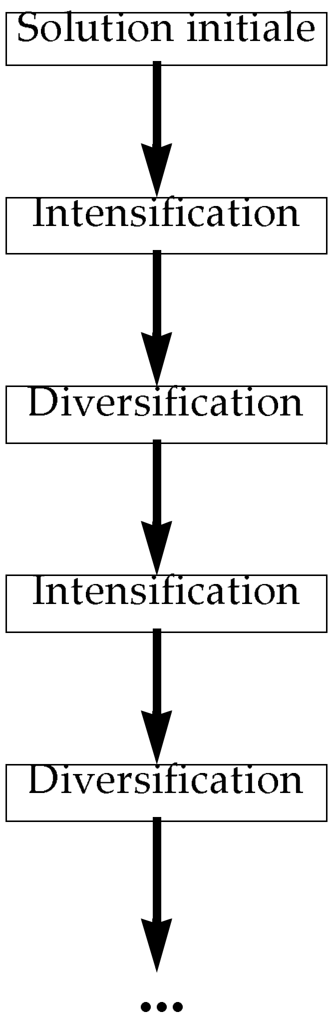
Intensification

- Recherche avec tabous de base
- Retour à une bonne solution
- Modification des paramètres de la recherche locale
- Liste de candidats

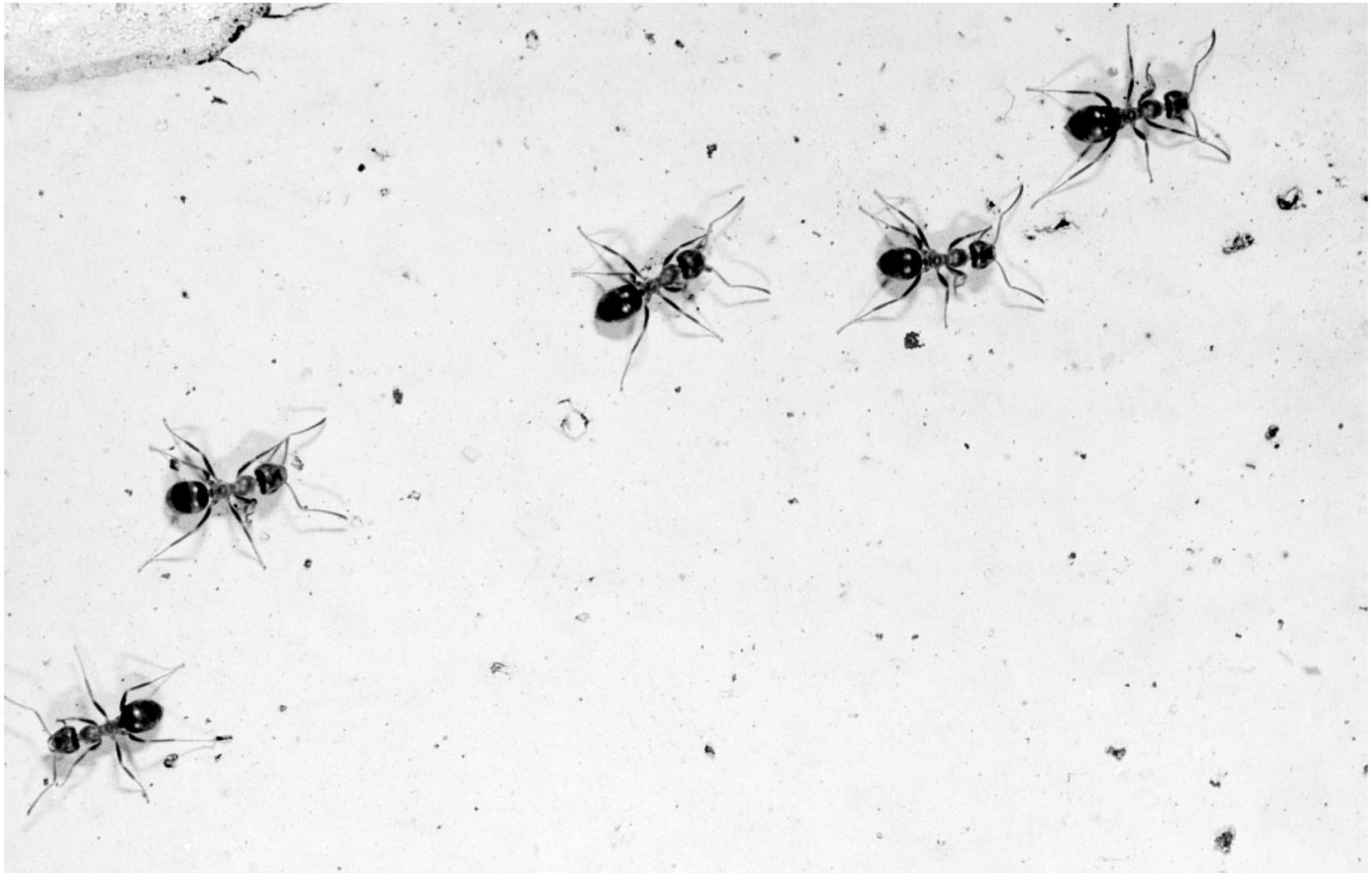
Diversification

- Modification de l'objectif
- Modification du voisinage
- Modification des paramètres de la recherche locale

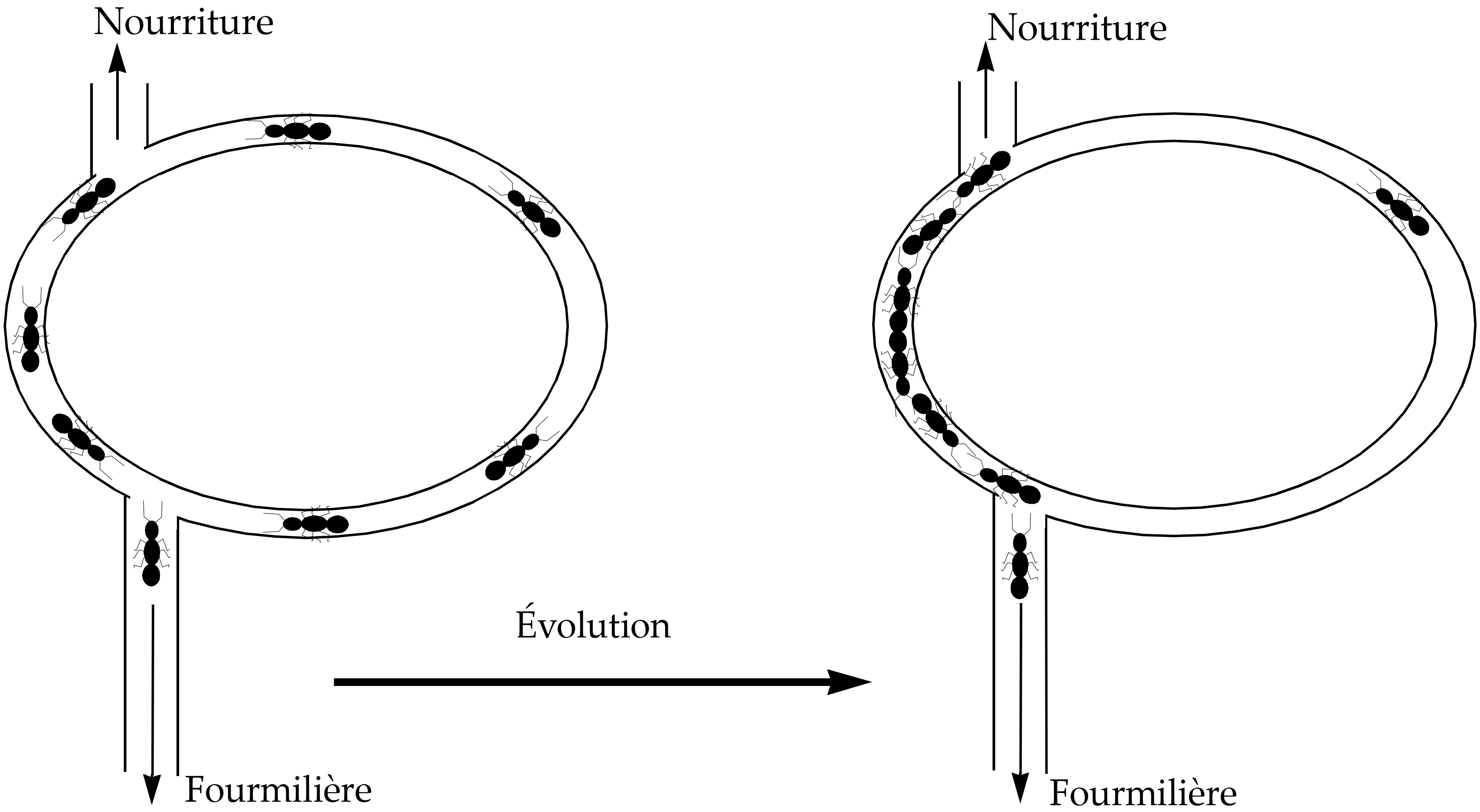
Chemins reliant deux solutions (path relinking)



CHEMINS DE FOURMIS

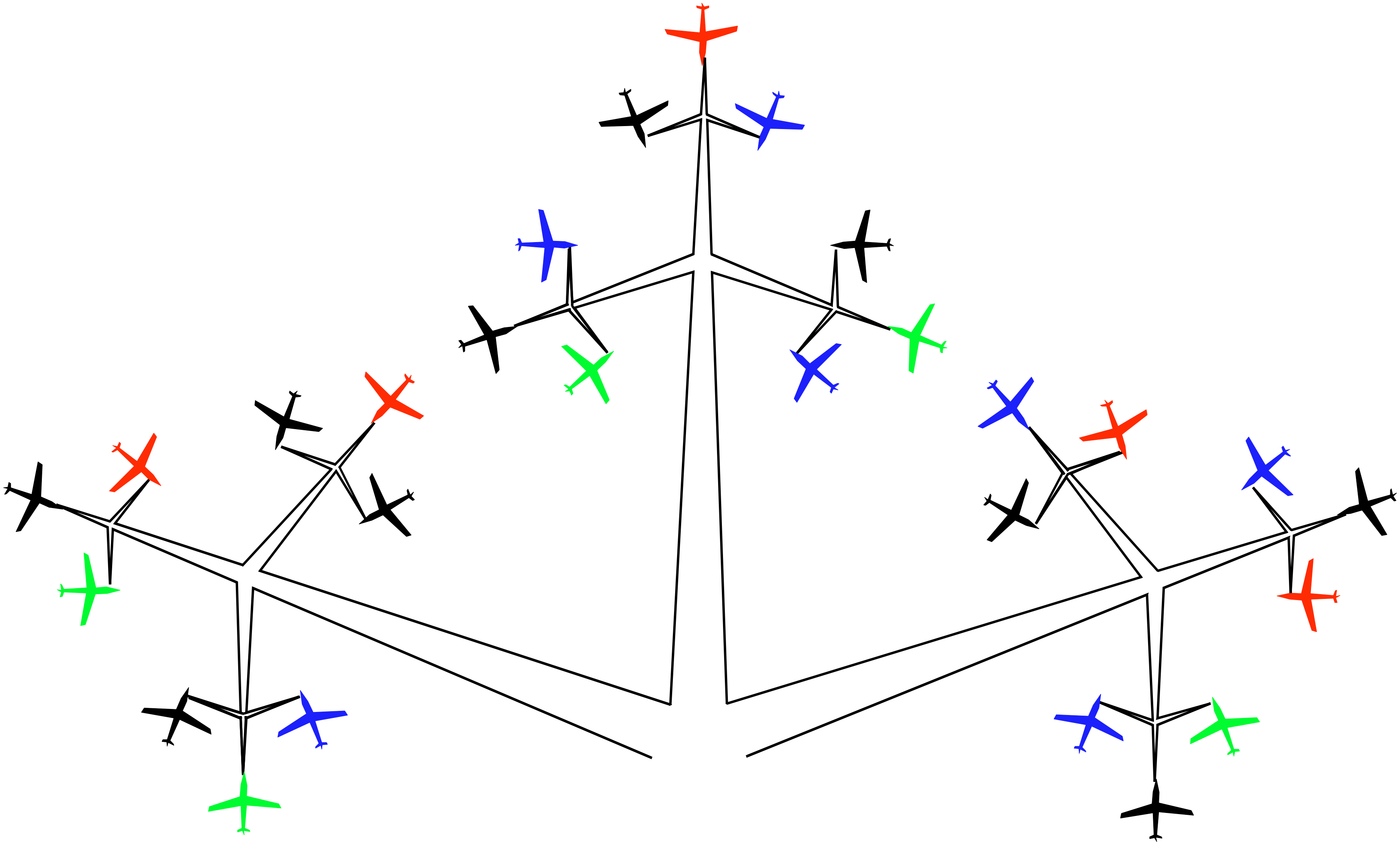


Optimisation de chemin par les fourmis



La fourmilière est séparée d'une source de nourriture par deux tubes différents. Après un certain temps, les fourmis empruntent le tube le plus court, les phéromones dans celui-ci augmentant plus rapidement.

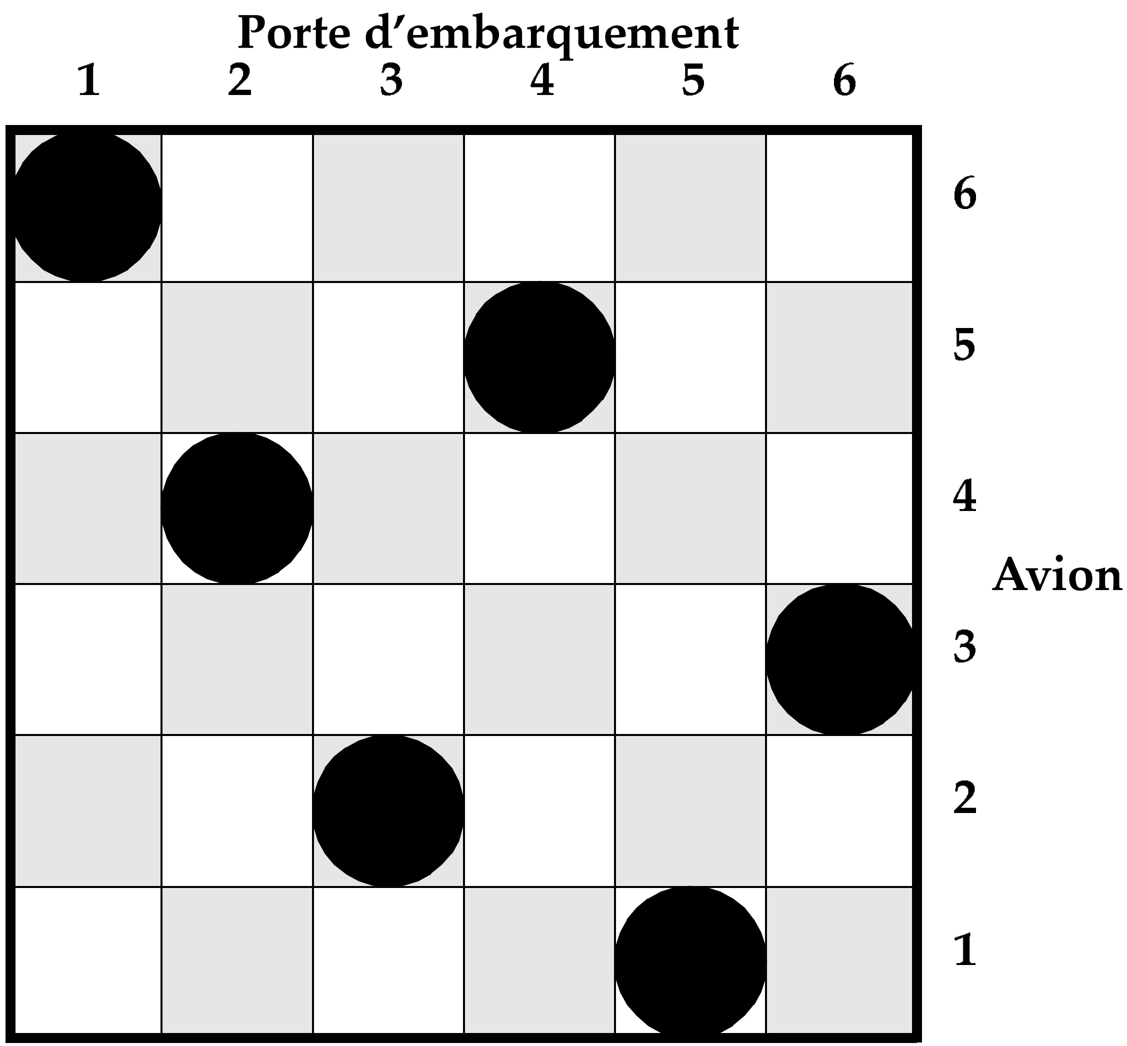
PROBLÈME D'AFFECTATION QUADRATIQUE



Quelle porte d'embarquement allouer à chaque avion ?

MODÉLISATION DU QAP:

Placer des tours sur un échiquier de telle manière qu'elles ne s'attaquent pas.

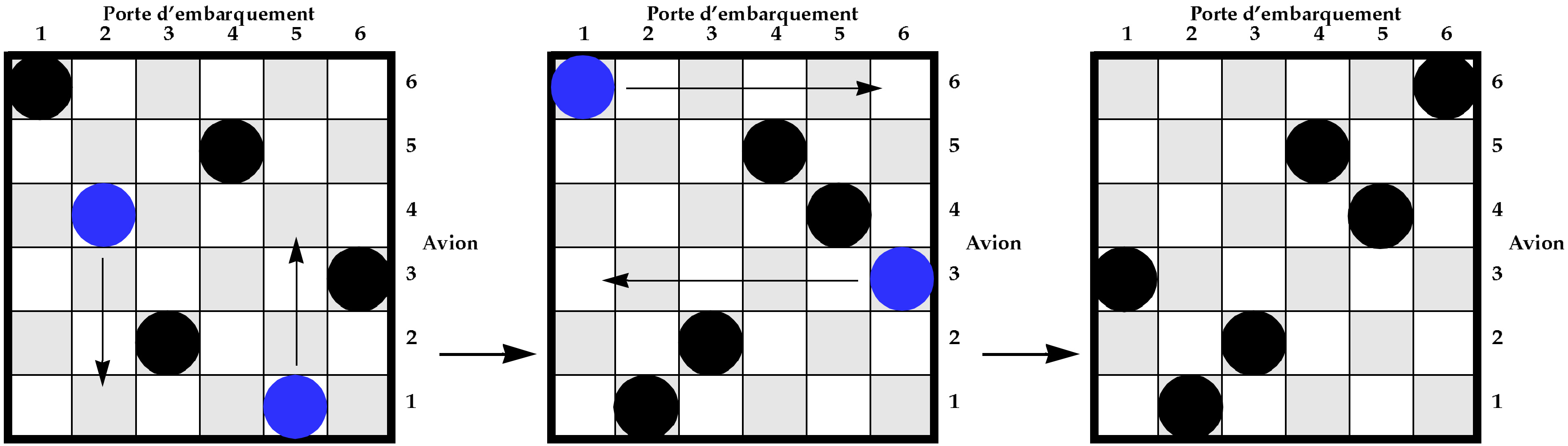


COLONIES DE FOURMI ARTIFICIELLES

Processus de construction d'une solution, à répéter un grand nombre de fois :

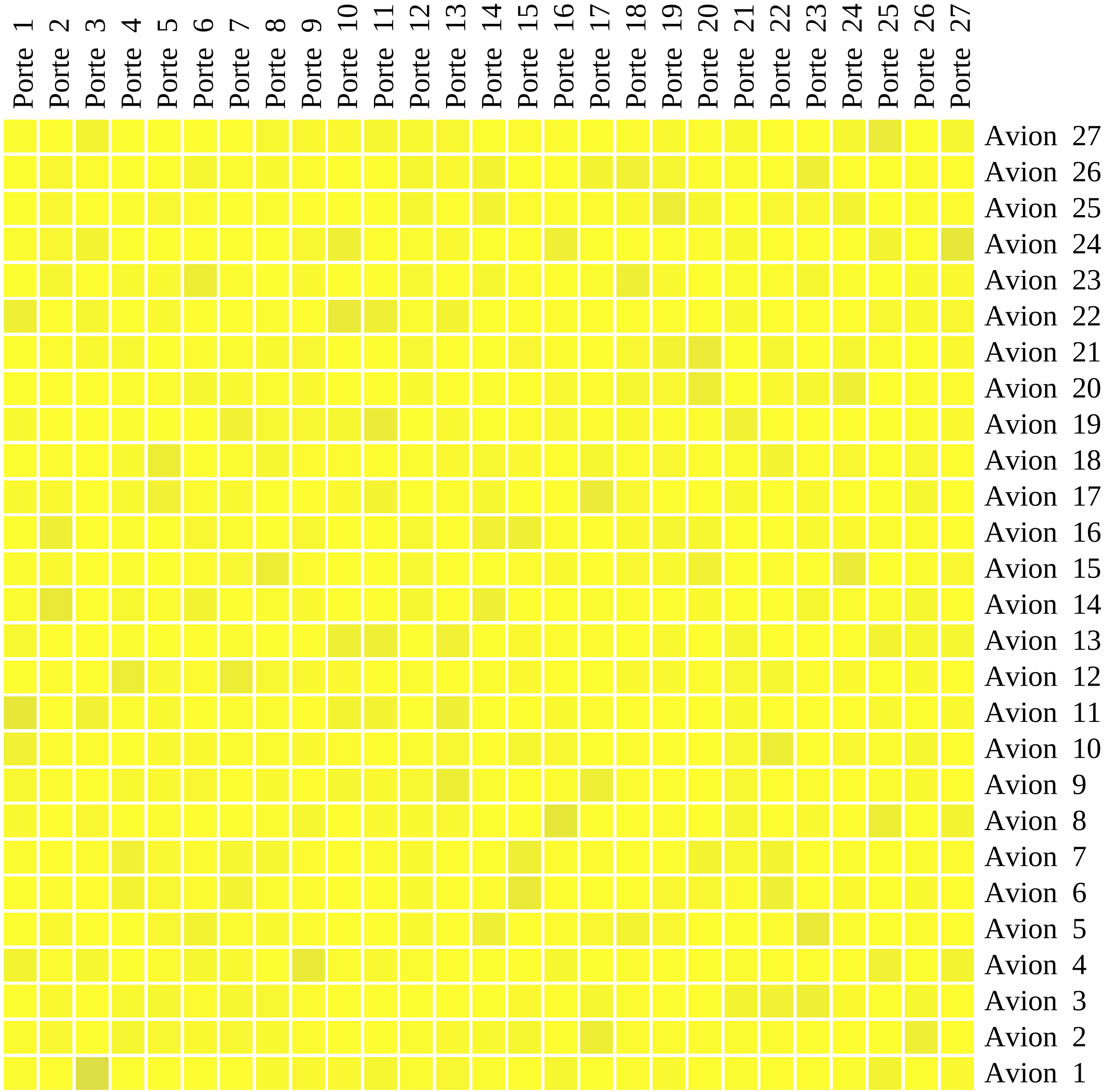
1. Répéter pour chaque tour: [Construction d'une nouvelle solution]
   Choisir une case admissible avec une probabilité proportionnelle à la quantité de phéromone artificielle déposée sur la case.

2. Améliorer la solution ainsi construite en déplaçant 2 tours à la fois jusqu'à l'obtention d'un optimum local:

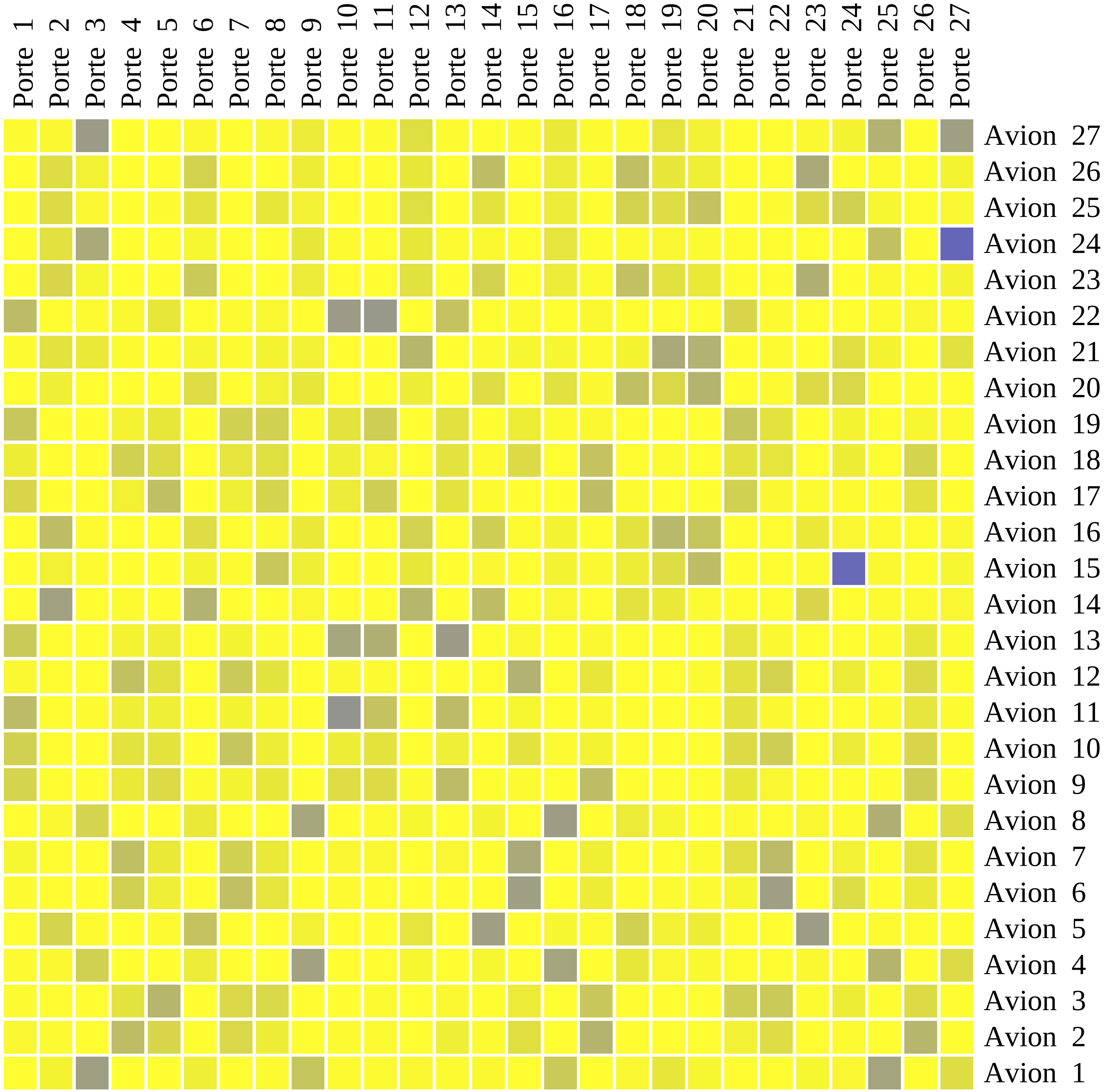


3. Déposer une certaine quantité de phéromone artificielle sur les cases occupées en dernier lieu par les tours. [Mise à jour de la mémoire]

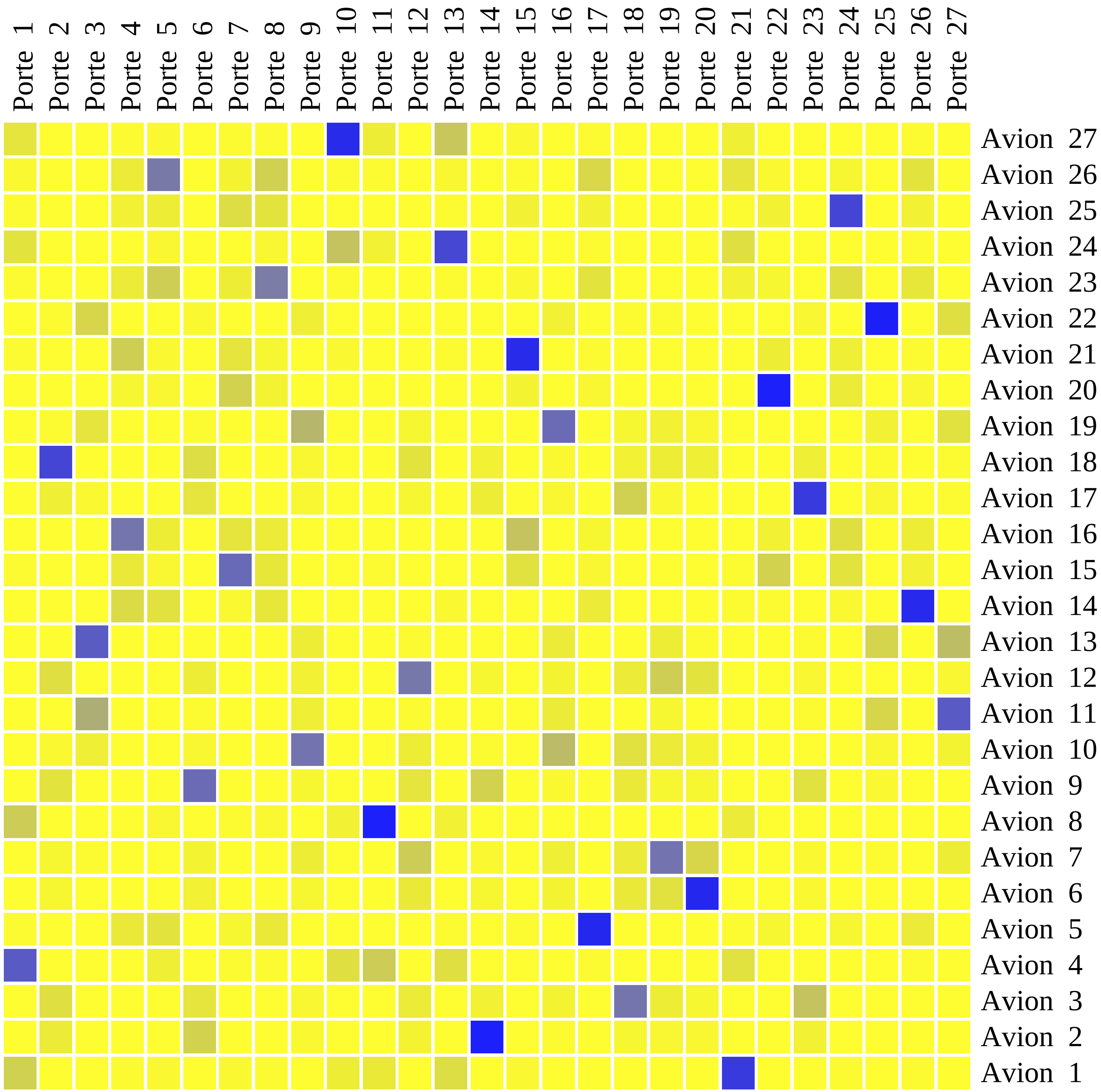
Traces de phéromones (bleues) au début



ÉVOLUTION DES TRACES AU COURS DU TEMPS



Traces en fin d'exécution du programme



PAM II: COLONIE DE FOURMIS

Mémoire:

Statistiques sur la présence de certaines composantes dans les solutions générées (traces).

Construction de nouvelles solutions

de type GRASP.

Mise à jour de la mémoire :

Statistique initialement proposée (i : composante de la solution s, de qualité q, $0 < \alpha < 1$ et β: paramètres)

$$\tau_i(t+1) = \begin{cases} \alpha \cdot \tau_i(t), & (i \notin s) \\ \alpha \cdot \tau_i(t) + \beta \cdot q, & (i \in s) \end{cases}$$

Faiblesse:

Convergence très lente, ne termine pas dans un optimum local.

Corrections:

Ajout d'une procédure de recherche locale

Processus de coordination

Renforcement sélectif des traces

Stratégies de recherche

COLONIE DE FOURMIS (VISION ALTERNATIVE)

Processus Reine

Initialiser la mémoire M

Répéter (en parallèle):

- Choisir les paramètres d'un processus Fourmi et l'activer
- Recevoir une solution d'un processus Fourmi et mettre à jour M

Retourner la meilleure solution trouvée

Processus Fourmi

Construire une solution s' (à l'aide de M et de paramètres)

Appliquer une recherche locale sur s' pour obtenir s

Retourner s au processus Reine

Remarque: vision très proche de PAM, permettant d'intégrer des composantes d'autres méta-heuristiques, ne s'éloigne pas trop de la métaphore

EXEMPLE DE PAM POUR L'AFFECTATION QUADRATIQUE

FANT

Solution :

Matrice $\boldsymbol{X} = (x_{ij})$ de variables de décision (0, 1) indiquant si l'élément i est placé en j

Mémoire :

matrice de traces $\boldsymbol{M} = (m_{ij})$, statistique sur le nombre de fois que l'élément i est placé en j

$$\boldsymbol{M} \leftarrow \boldsymbol{M} + r \cdot \boldsymbol{X}$$

$$\boldsymbol{M} \leftarrow \boldsymbol{M} + R \cdot \boldsymbol{X}^*$$

Construction d'une nouvelle solution :

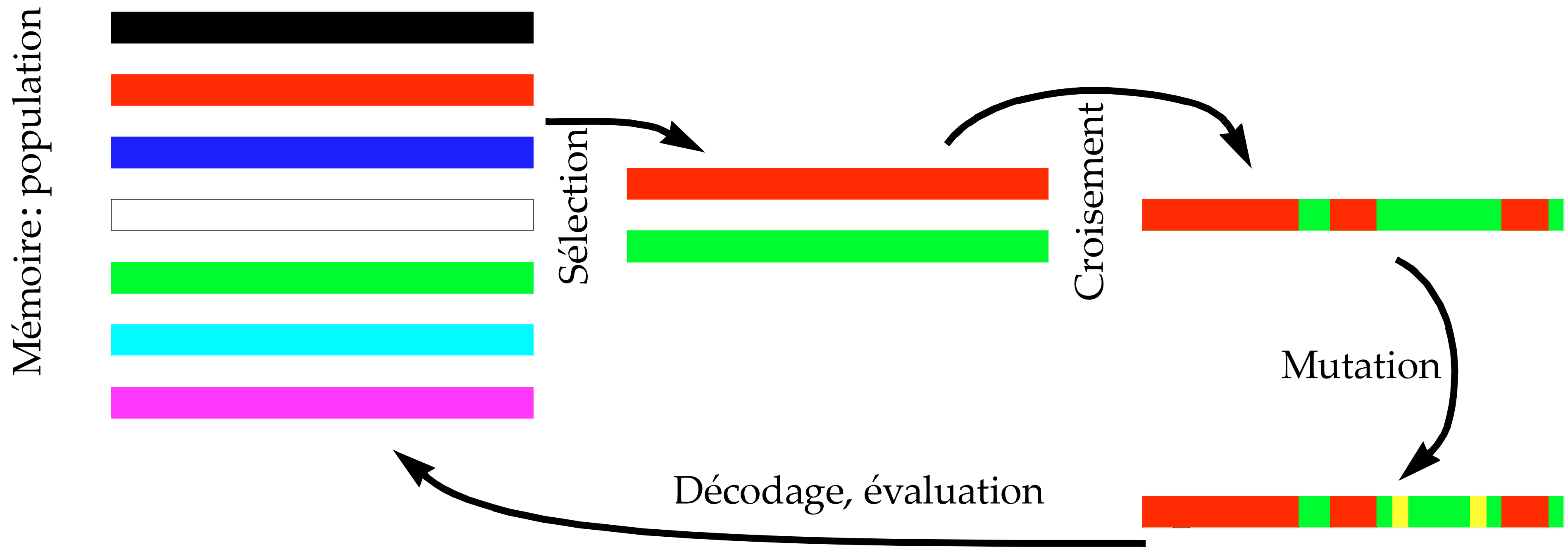
Probabilité placer i en j proportionnel à m_{ij}

Mise à jour de la mémoire :

$\boldsymbol{M} = \mathbf{1}$, $r = 1$ si on a amélioré la meilleure solution

$r \leftarrow r + 1$, $\boldsymbol{M} = r\cdot\mathbf{1}$, si on a généré à nouveau la meilleure solution

PAM III: ALGORITHME GÉNÉTIQUE



Point fort: Bonne couverture de l'espace des solutions

Faiblesses:

Très sensible au codage des solutions

→ Représentation naturelle, croisement «intelligent»

Pas de recherche locale ⇒ Convergence lente, solutions finales médiocres

→ Mutation = recherche locale

EXEMPLE DE PAM POUR L'AFFECTATION QUADRATIQUE

Algorithme génétiques hybrides (Fleurent & Ferland)

Mémoire:

Population P de solutions

Construction d'une nouvelle solution (Tate & Smith):

Sélectionner π et ρ de P, de manière probabiliste selon leur rang

Choisir l'élément σ_i de la nouvelle solution aléatoirement entre π_i et ρ_i (lorsque c'est possible)

Compléter aléatoirement les éléments de σ qui n'ont pas pu être choisis.

Recherche locale basée sur des transpositions

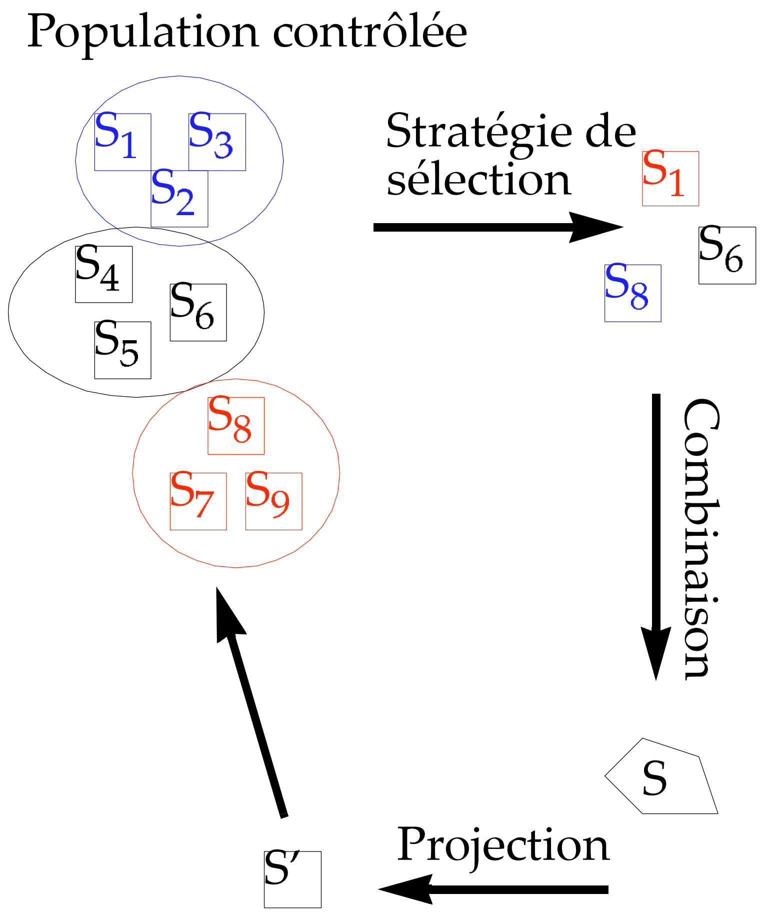
Descente rapide

Recherche avec tabous

Mise à jour de la mémoire :

Remplacer la plus mauvaise solution de la mémoire.

Recherche par dispersion



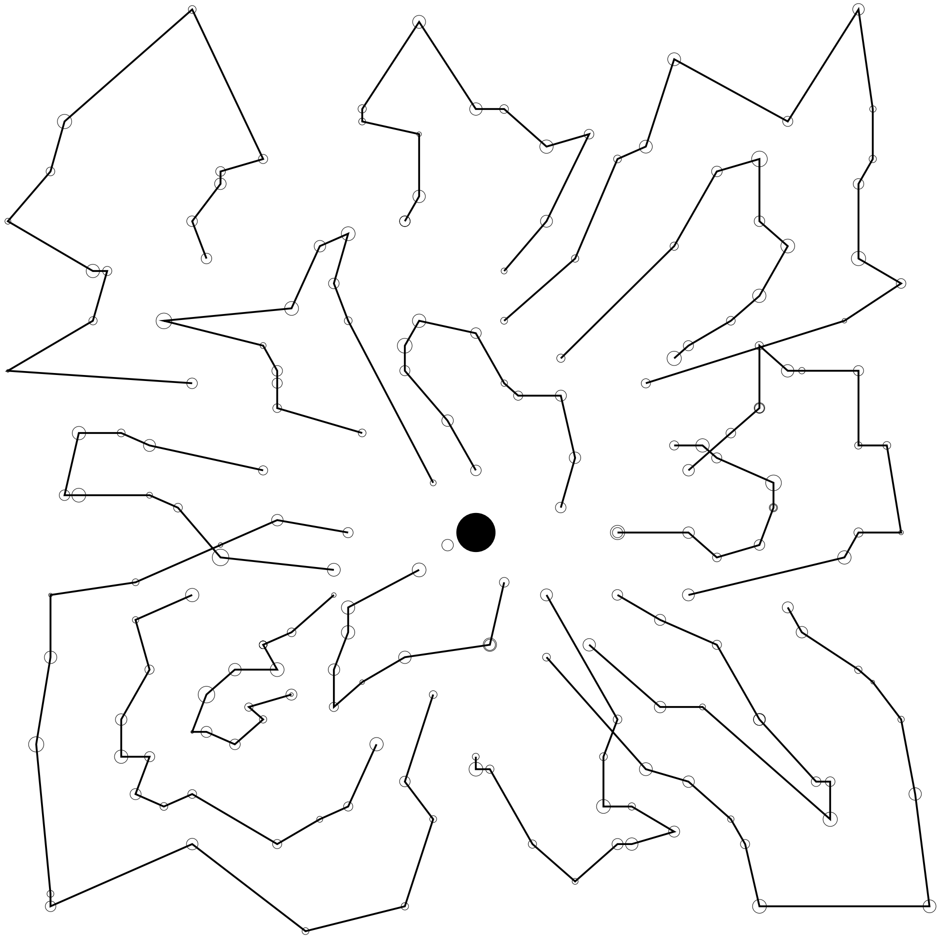
Extension des algorithmes génétiques

- Vecteurs d'entiers
- Sélection de plus de deux solutions
- Combinaison linéaire des solutions sélectionnées
- Opérateurs de réparation, de projection
- Stratégies de gestion de la population

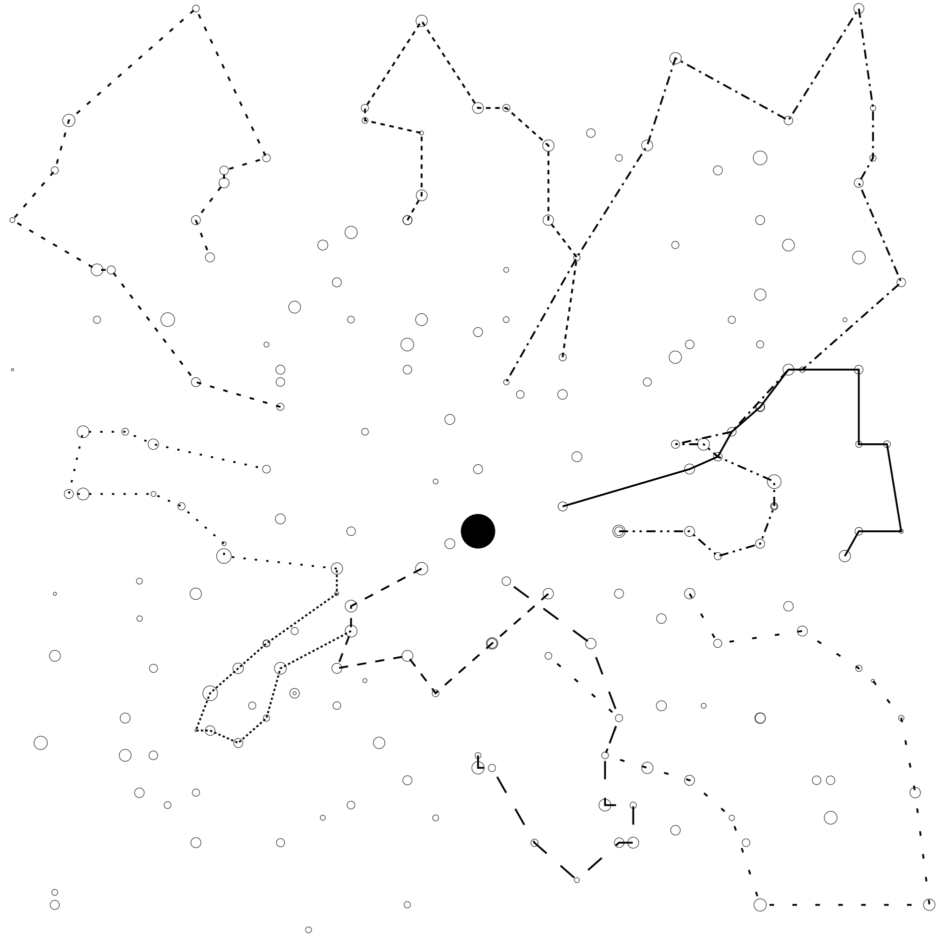
Extension de la recherche par dispersion:

- Construction de vocabulaire

CONSTRUCTION DE VOCABULAIRE



Meilleure solution connue d'un problème



Quelques tournées trouvées après recherche sommaire avec une technique non déterministe

PAM POUR L'ÉLABORATION DE TOURNÉES DE VÉHICULES

Mémoire *M*:

Population de tournées

Construction d'une nouvelle solution :

a) Donner un rang $r_T > 0 \;\forall T \in M$; $s \leftarrow \emptyset$.

b) Répéter, tant que $M \neq \emptyset$:

b1) Choisir une tournée $T \in M$, avec probabilité $r_T / \sum_{k \in M} r_k$.

b2) Poser $s \leftarrow s \cup \{T\}$.

b3) Pour toute tournée $T' \in M$ telle que $T \cap T' \neq \emptyset$

poser $M \leftarrow M \backslash \{T'\}$.

c) Compléter la solution partielle s

Question:

D'inspiration génétique, recherche par dispersion, recherche avec tabous ou fourmis ?

Le processus peut s'appliquer directement au problème de coloration de graphes.

PAM POUR D'AUTRES TYPES DE VRP

1) VRP avec tournées multiples (avec Gendreau & Laporte)

Exécuter un PAM pour le problème standard

Résoudre un problème de mise en boîte avec les tournées en mémoire

2)VRP avec objectif MinMax (avec Golden & Laporte)

Poser $L_{max} = \infty$

Exécuter un PAM avec :

Mémorisation des tournées $< L_{max}$ seulement

Recherche locale n'acceptant que les tournées $< L_{max}$

Diminuer la valeur de L_{max} lorsqu'une solution admissible est trouvée

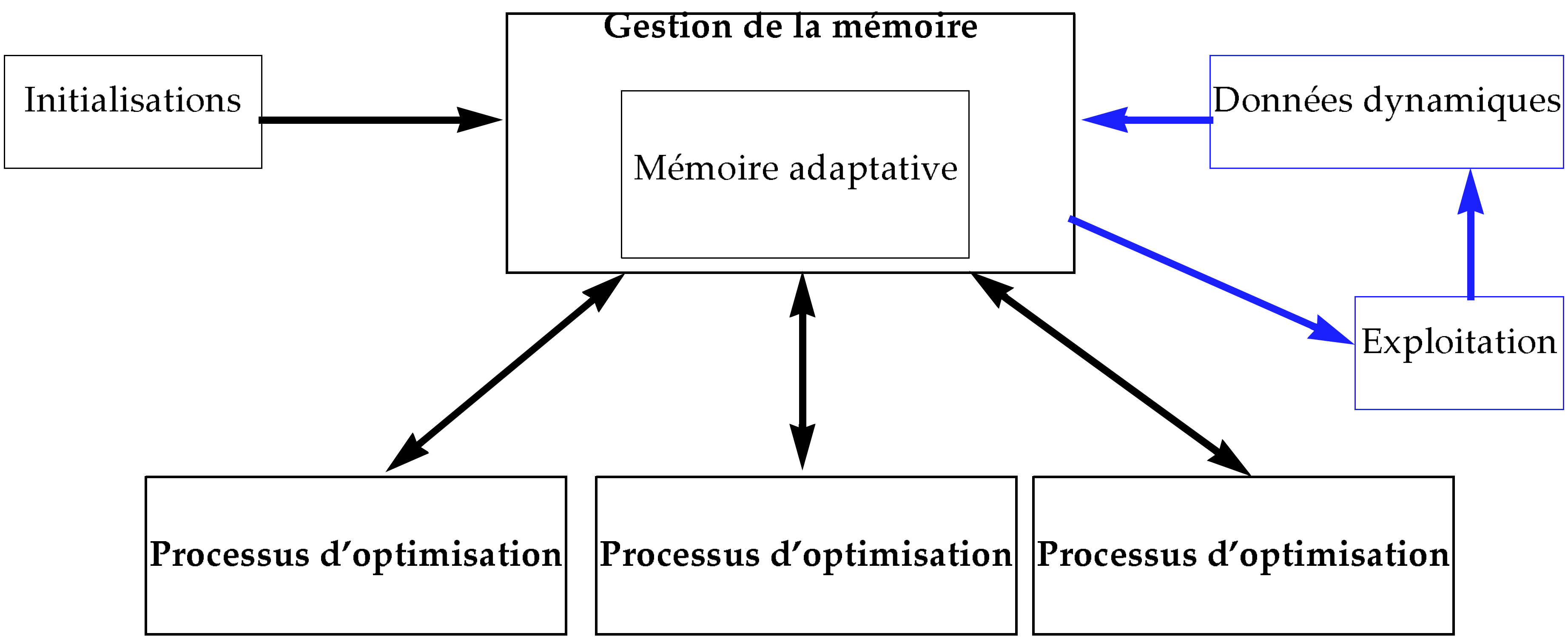
3)VRP avec flotte hétérogène

Exécuter un PAM pour chaque type $1, \ldots, k$ de véhicule

Résoudre un problème de partition d'ensemble avec des contraintes additionnelles en considérant les tournées contenues dans $M_1 \cup M_2 \cup \ldots \cup M_k$

PAM CONCURRENTE

```
Initialiser la mémoire
Répéter
   Construire une (plusieurs) solutions en s'aidant de la mémoire
   Améliorer la (les) solutions avec une recherche locale
   Mettre à jour la mémoire
```



CONCLUSIONS

Analyse d'heuristiques inspirées de méta-heuristiques à mémoire

Difficile de retrouver la métaphore

Synthèse: Programmation à mémoire adaptative

Conceptuellement simple

Peu limitative

Facilement parallélisable

Applications pratiques

Problèmes dynamiques

Recherche locale :

Composant clé des méta-heuristiques

Mémoire :

Particulièrement utile pour les longues exécutions et les problèmes structurés.

POPMUSIC

Partial OPtimisitation Meta-heuristic Under Special Intensification Conditions

Recherches locales indépendantes

LOPT

TRAME GÉNÉRALE DE POPMUSIC

0) Entrée : Solution s composée de parties s_1, …, s_p.

1) C = {s_1, …, s_p}

2) Tant que $C \neq \emptyset$, répéter :

   2a) Choisir $s_i \in C$; soit le sous-problème R composé de l'ensemble des r plus proches sous-problèmes de s_i ($s_i \in R$)

   2b) Optimiser R

   2c) Si R a été amélioré, poser $C \leftarrow C \cup R$;
   sinon, poser $C \leftarrow C \setminus \{s_i\}$.

La mise au point d'une heuristique basée sur POPMUSIC requiert la définition de ce qu'est :

Une partie

La distance entre deux parties

La procédure d'optimisation

CONCEPTS EN RELATION AVEC POPMUSIC

Glover

Liste de mouvements candidats

Variables consistantes et fortement déterminées

Woodruff

« Chunking »

Shaw

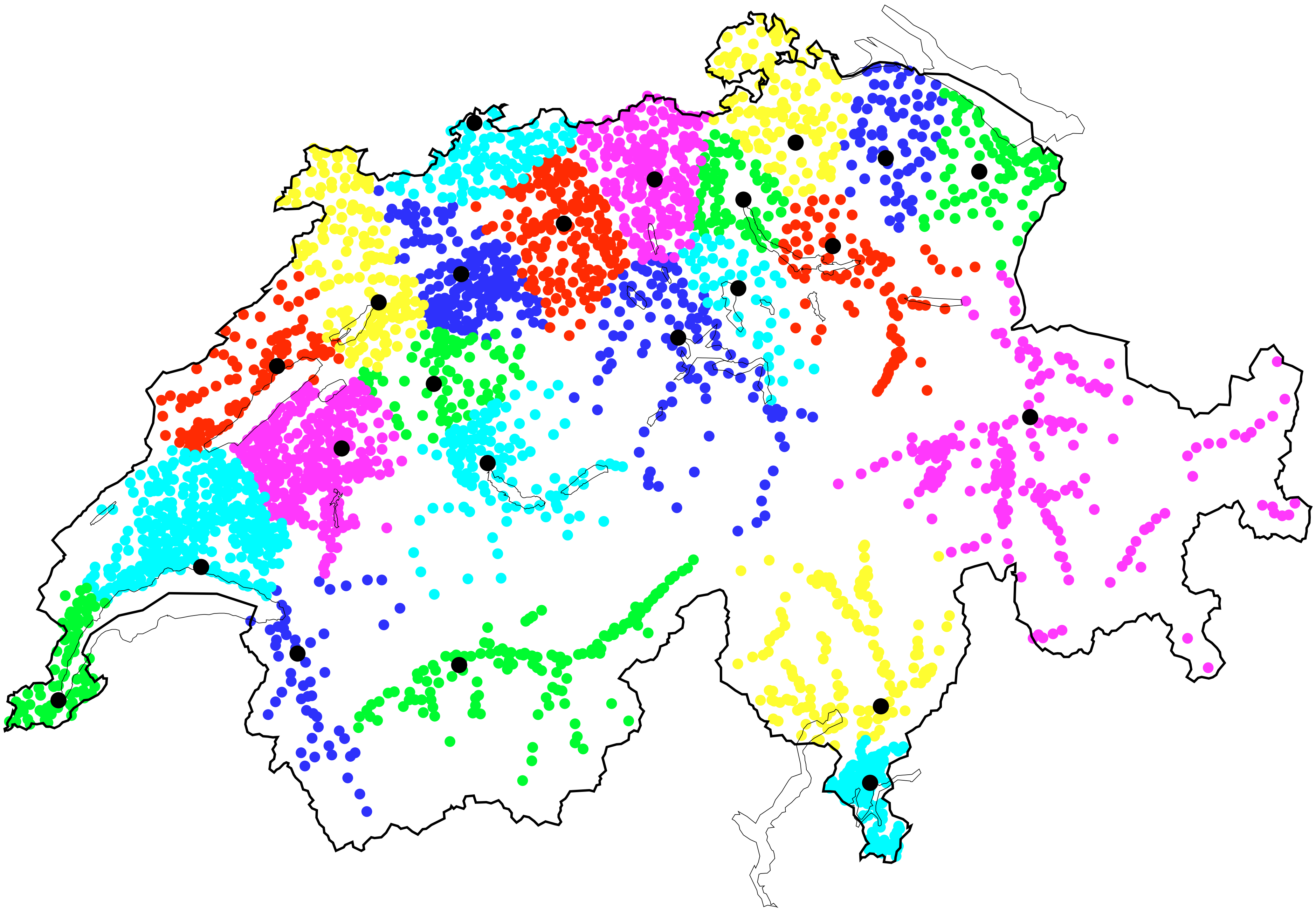
Grands voisinages

Hansen & Mladenovic

Variable neighbourhood decomposition search

En général : méthodes de décomposition

PROBLÈMES DE PLACEMENT (CLASSIFICATION GLOBALE)



POPMUSIC POUR LA CLASSIFICATION GLOBALE

Partie :

Ensemble des entités appartenant à un groupe

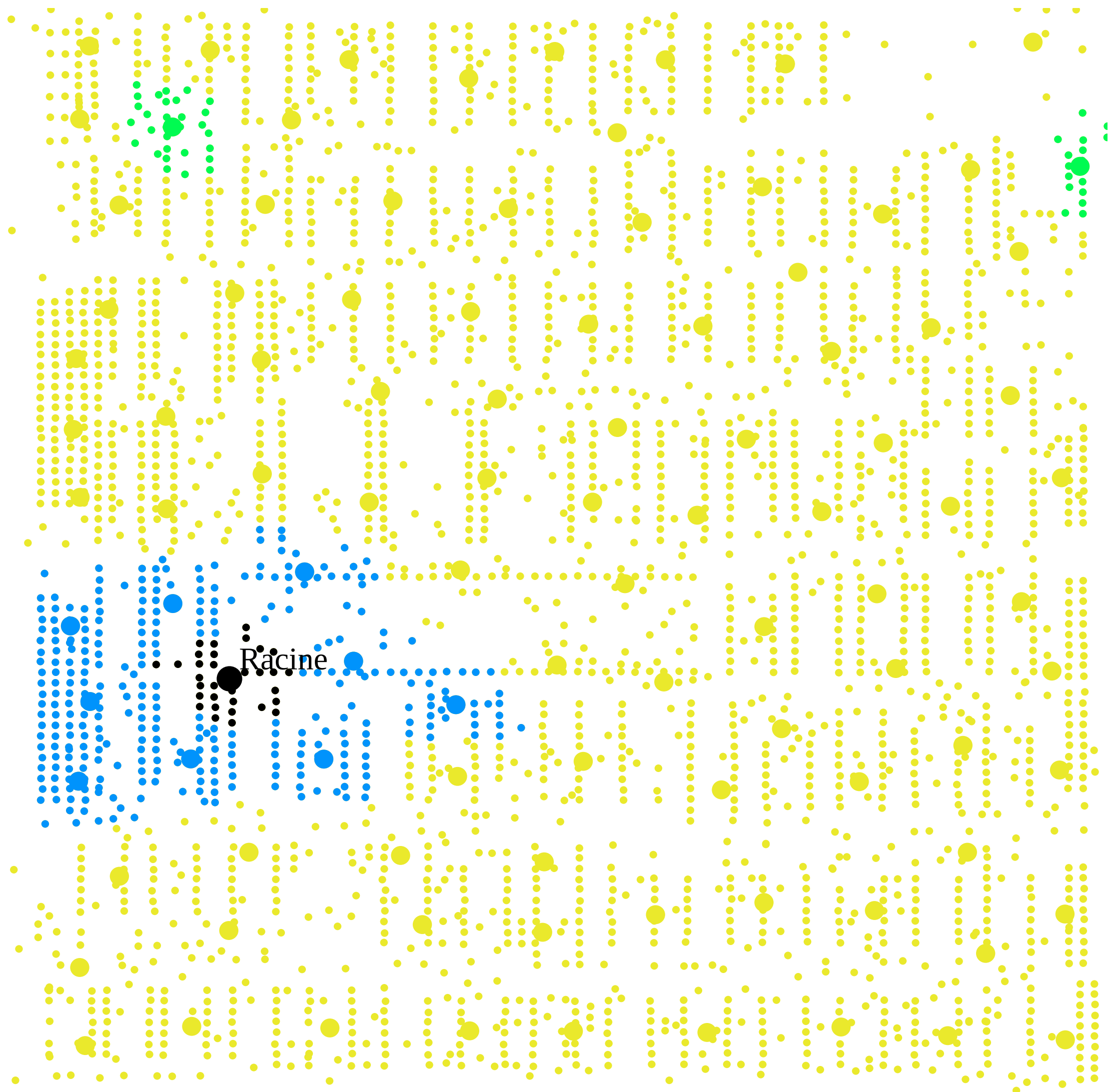
Distance :

Dissimilarité moyenne entre les éléments des groupes,

Distance entre les centres

Processus d'optimisation:

Méthode de descente basée sur une liste de candidats, re-localisation des centres, stabilisation (CLS)



Résultats numériques

Minimisation de la somme des carrés.

Exemple de problème : TSPLIB pcb3038.

Résultats de RVNS et VDNS de Hansen & Mladenovic 1999.

Problème		Qualité [% au dessus de la meilleure]				Temps de calcul [s SPARC10]			
p	Meilleure valeur connue	RVNS	VDNS	POPMUSIC (6, 40)	POPMUSIC (10, 100)	RVNS	VDNS	POPMUSIC (6, 40)	POPMUSIC (10, 100)
100	47685934.0	2.34	0.73	1.19	0.44	153	1132	145	505
150	30524769.8	3.13	1.44	1.16	0.58	153	1676	111	355
200	21875113.9	2.49	1.10	1.07	0.50	160	2124	96	262
250	16621446.4	2.56	1.34	1.35	0.76	182	2954	89	234
300	13289633.4	2.50	1.57	1.58	0.78	229	3151	82	205
350	11019171.4	2.60	1.36	1.69	0.76	231	3760	75	179
400	9362179.2	3.35	1.82	1.40	0.66	165	3446	72	170
450	8101618.7	3.47	1.71	1.61	0.80	242	4152	69	163
500	7102678.4	2.85	1.86	1.70	0.90	204	4060	68	156

Problème de la p-médiane (Wälti, Cung, Mautor, Taillard 2003)

Problème à 11849 éléments, 100-1000 centres résolus en parallèle (6x2 proc.) en moins d'une minute.

Problème à 3038 éléments, solutions à env. 0.4% obtenues en quelques secondes.

Resende & Werneck (2003) : GRASP 0.2% en 5 minutes, Path relinking 0.02% en 10 minutes (SGI Challenge)

POPMUSIC pour le VRP (Taillard 1993, …)

Partie :

Tournée

Distance entre parties :

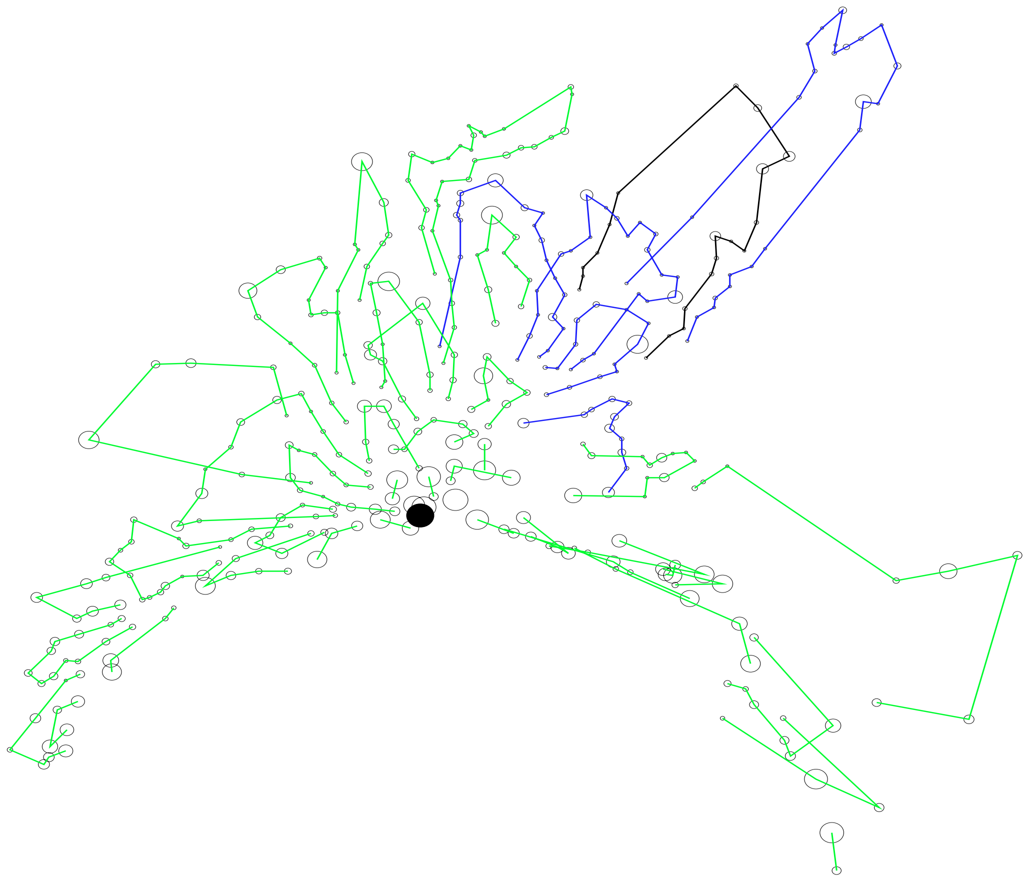
Différence d'angle entre les centres de gravité de deux tournées

Recherche locale :

Recherche avec tabous de base

Rochat & Semet 1994

Variation du paramètre r ($\equiv$VNS)



GRAND VOISINAGE POUR LE VRP (SHAW 1998)

Partie:

Client

Distance :

Distance euclidienne + composante aléatoire

Processus d'optimisation :

Réinsertion optimale (ou heuristique) des clients à l'aide de programmation par contraintes

PROBLÈME D'ÉQUILIBRAGE

Données :

n parties interchangeables (pales de turbine, conteneurs) de masse w_i ($i = 1, \ldots, n$)

n positions de placement des parties (x_i, y_i, …) ($i = 1, \ldots, n$)

Centre de gravité idéal de l'ensemble des parties : (0, 0, …)

Objectif:

$$\min_{\pi \in \Pi} = \frac{1}{\sum w_i} \cdot \left\| \left(\sum_{i=1}^{n} w_i \cdot x_{\pi_i}, \sum_{i=1}^{n} w_i \cdot y_{\pi_i}, \ldots \right) \right\|$$

Applications typiques:

Équilibrage de navires ou autre véhicule de transport

Équilibrage de turbines

Problème NP-difficile

Bipartition $\propto$ Équilibrage

Laporte & Mercure 1988, Sondergeld & Voß 1996, Mason & Rönnqvist 1997

Équilibrage $\propto$ QAP $\min_{\pi \in \Pi} \sum_{i=1}^{n} \sum_{i=1}^{n} \ldots$!?

POPMUSIC POUR L'ÉQUILIBRAGE

Partie :

Partie interchangeable

Distance :

Différence de masse

Processus d'optimisation :

Recherche avec tabous « robuste » (Ro-TS Taillard 1991)

RÉSULTATS NUMÉRIQUES (RO-TS)

REM : Reverse Elimination Method

SS : Star Shape Diversification (Sondergeld & Voß 1996)

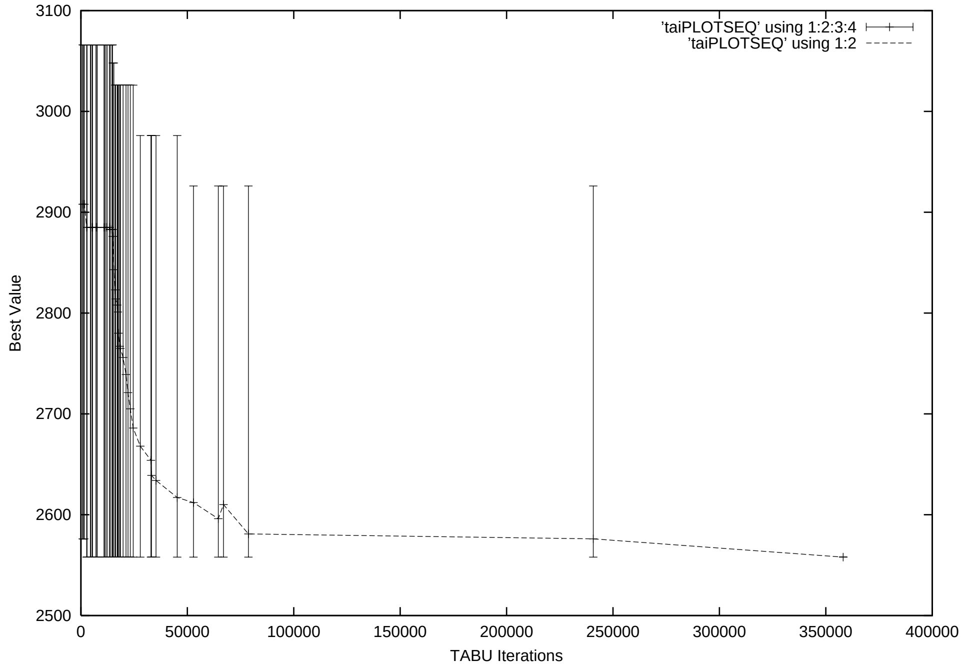
Turbines avec 5..80 pales	REM/1	REM/10	SSA1/1	SSA1/10	SSA2/1	SSA2/10	Ro-TS
$\sum$ excentricités	37.28685	37.55715	37.32012	37.47627	37.30784	37.30163	37.26166
$\sum$ Temps CPU [s] (10000 itérations)	> 5000	> 5000	> 5000	> 5000	> 11000	> 11000	109

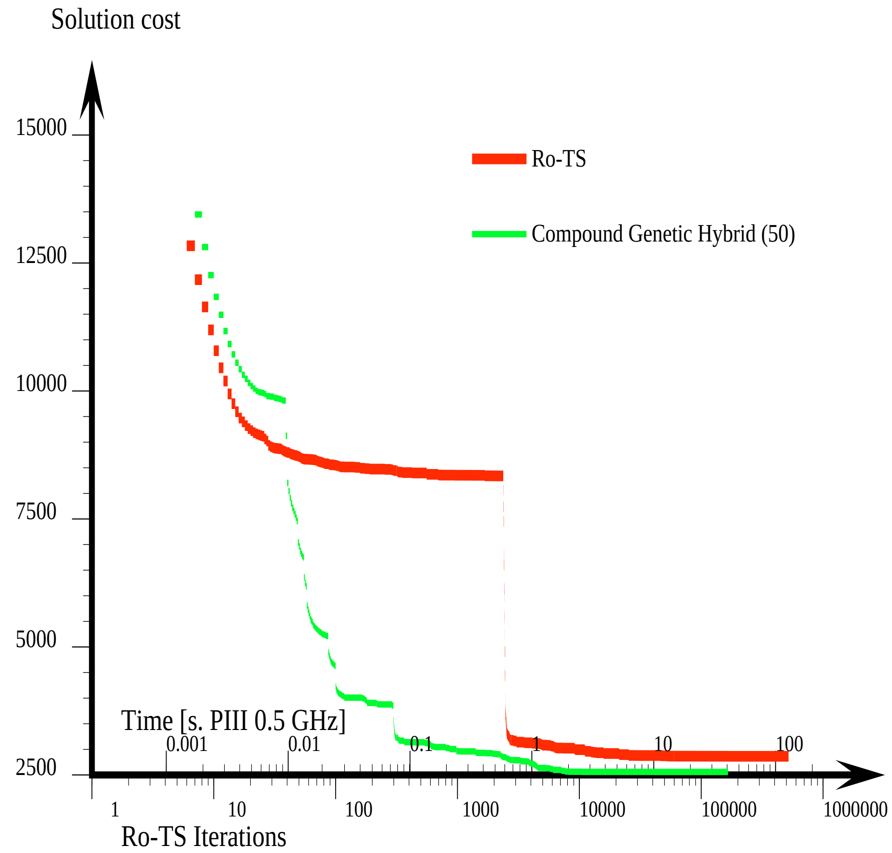
RÉSULTATS NUMÉRIQUES (POPMUSIC)

Problème		Qualité	Temps de calcul [s SPARC5]		Seuil de confiance pour 300 (10'000) exécutions
Nom	Taille	TS/POP	POPMUSIC (22, 1070)	TS(10'000)	POPMUSIC «meilleur» que TS
linear30	30	0.991	3.26	4.06	98.2
linear40	40	1.169	6.92	6.75	99.996
linear50	50	1.300	10.21	10.26	99.999
linear60	60	1.437	14.11	15.13	99.999
linear70	70	1.419	18.44	20.53	99.999
linear80	80	1.494	21.88	26.71	99.999
mas38	38	0.878	4.67	5.92	0.000
mas40	40	1.037	6.24	6.30	99.999
mas100-1	100	1.201	36.29	43.02	99.999
mas100-2	100	1.234	44.08	42.85	99.999
mas100-3	100	1.096	51.08	42.12	97
mas100-4	100	1.280	39.04	42.03	99.999
mas100-5	100	1.239	42.86	43.01	99.999
mas100-6	100	1.145	35.50	42.81	99.98
mas100-7	100	1.300	38.47	42.74	99.999
mas100-8	100	1.305	38.91	42.50	99.999
mas100-9	100	1.324	42.78	42.16	99.999
mas100-10	100	1.149	37.70	43.51	98.3

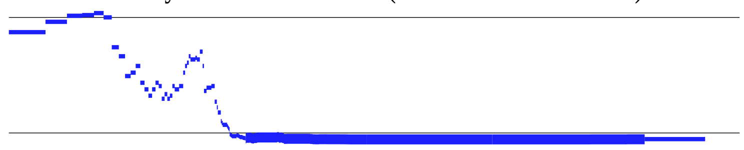
COMPARAISON DE MÉTHODES ITÉRATIVES NON DÉTERMINISTES

PROBLÉMATIQUE





Probability TS better than CGH (Confidence = 5% - 95%)



COMPARAISON BINAIRE (TROUVÉ - PAS TROUVÉ)

	FANT			MMas	
Itération	Trouvé	Temps CPU	Itération	Trouvé	Temps CPU
25000	Non	83.0	1335	Oui	4.4
778	Oui	2.7	26104	Non	85.0
25000	Non	83.0	26104	Non	85.0
24658	Oui	81.9	26104	Non	85.0
1052	Oui	3.5	173	Oui	0.6
19787	Oui	65.6	1389	Oui	4.5
784	Oui	2.61	26104	Non	85.0
1009	Oui	3.35	26104	Non	85.0
253	Oui	0.85	26104	Non	85.0
5010	Oui	16.6	26104	Non	85.0

Conditions de l'expérience : Peu d'exécutions (le théorème central limite ne s'applique pas), nombre d'itérations (ou temps d'exécution, nombre de noeuds de l'arbre de recherche) limitées.

Questions :

Est-ce que FANT est significativement meilleur que MMas (i. e. est-ce qu'un taux de succès de 8/10 est significativement meilleur qu'un taux de 3/10) ?

Est-ce que FANT est significativement meilleur que MMas après 10 secondes ?

Et, en général, après t seconds ($0.6 < t < 83$) ?

TEST STATISTIQUE POUR LA COMPARAISON BINAIRE

Test classique du « U » (Gauss) :

Hypothèse : grands échantillons (> 14)

Problème : Comparer les fréquences relatives f_a et f_b de réponses « Oui » dans les deux échantillons

Hypothèse nulle : $f_a = f_b$

Hypothèse alternative : $f_a > f_b$

Échantillons : (n_a exécutions, a succès pour A), (Respectivement : n_b, b)

Calculer : $$U = \frac{a/n_a - b/n_b}{\sqrt{\dfrac{a/n_a \cdot (1 - a/n_a)}{n_a} + \dfrac{b/n_b \cdot (1 - b/n_b)}{n_b}}}$$

Rejeter l'hypothèse nulle au seuil α si $U > u_{1-\alpha}$

En d'autres termes, rejeter l'hypothèse que

l'heuristique A est aussi bonne que B si la différence entre les fréquences de succès est trop grande.

Test non standard pour la comparaison binaire

Problème : Comparer les fréquences f_a et f_b de succès dans deux échantillons

Hypothèse nulle : $f_a = f_b = p$

Hypothèse alternative : $f_a > f_b$

Échantillons : (n_a exécutions, a succès), (Respectivement : n_b, b)

Méthode :

En supposant que $f_a = f_b = p$, la probabilité $P(p)$ d'observer

a succès ou plus pour la méthode A

b succès ou moins pour la méthode B

est donnée par : $$P(p) = \sum_{i=a}^{n_a} \sum_{j=0}^{b} \binom{n_a}{i} \cdot p^i \cdot (1-p)^{n_a - i} \cdot \binom{n_b}{j} \cdot p^j \cdot (1-p)^{n_b - j}$$

L'hypothèse nulle sera rejetée au seuil de confiance α si $\max_p P(p) < 1 - \alpha$

Pour une méthode itérative, le test doit être répété pour chaque effort de calcul t

COMPARER LA QUALITÉ DE MÉTHODES HEURISTIQUES

Question : quelle est la meilleure des deux méthodes S et T (problème de minimisation) ?

Heuristique S		Heuristique T		Réponse	
Moyenne : 100		Moyenne : 101			S
Moyenne : 100	Écart-type : 6	Moyenne : 100	Écart-type : 12	Maintenant	S?
Moyenne : 100	Écart-type : 6	Moyenne : 100	Écart-type : 12	Demain	T?
Moyenne : 100	Écart-type : 6	Moyenne : 101	Écart-type : 12	Demain	T?

La réponse correcte dépend de la distribution des valeurs de solution, qui est inconnue a priori.

Exemple : 10 exécutions des heuristiques A, B, C donnent :

A : 98 98 98 98 98 98 98 98 98 118 Moyenne : 100, écart-type : 6

B : 96 96 96 96 96 96 96 96 96 136 Moyenne : 100, écart-type : 12

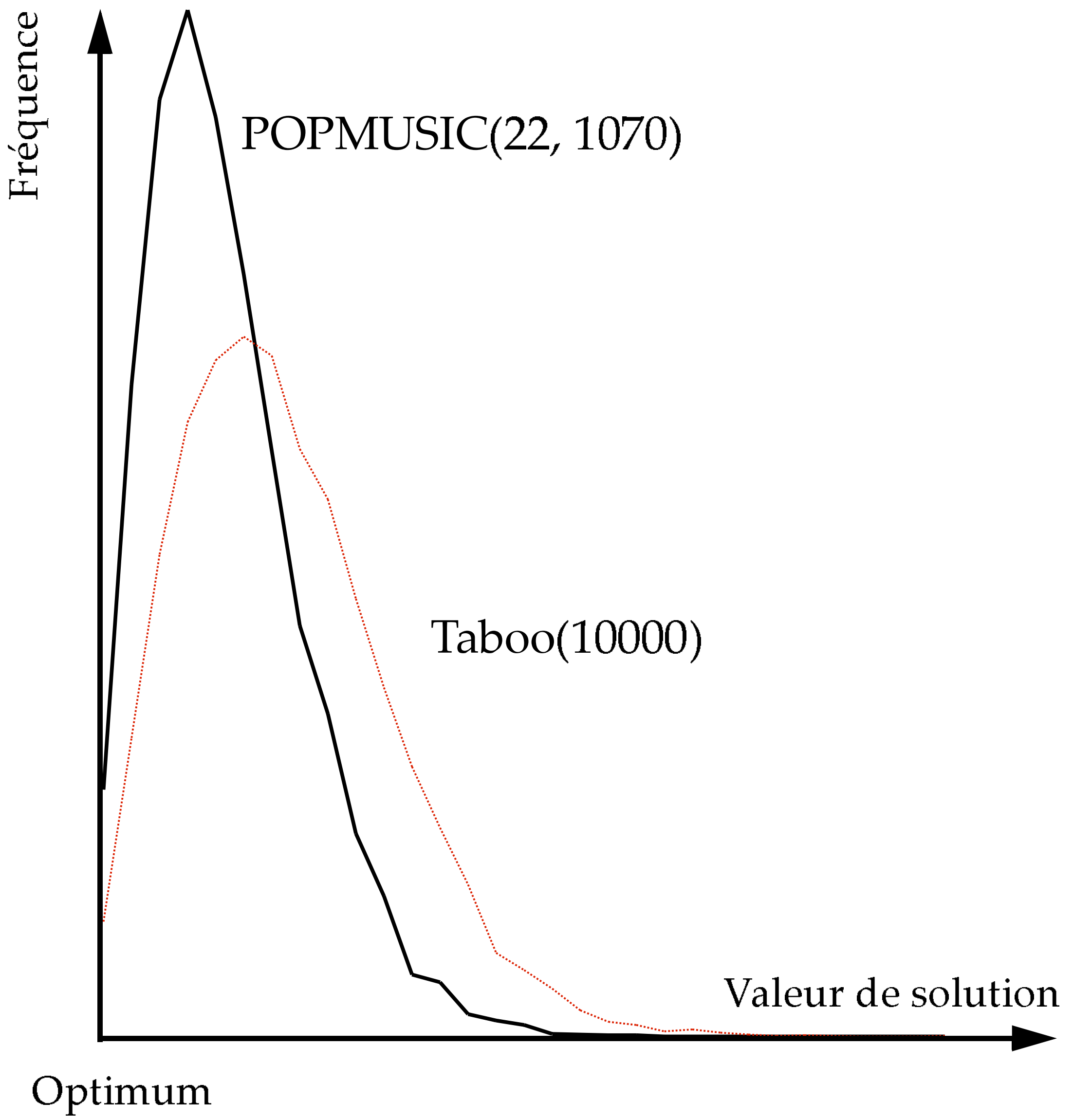
C : 97 97 97 97 97 97 97 97 97 137 Moyenne : 101, écart-type : 12

P(une exécution de A meilleure qu'une exécution de B) = P(A < B) : 0.1

Deux exécutions indépendantes : P(min(A, A) < min (C, C)) : 0.01

DISTRIBUTIONS DES VALEURS DE SOLUTIONS

Pas les mêmes, pas gaussiennes ! Généralement inconnues et non déterminables raisonnablement.



TEST STATISTIQUE STANDARD

Test du T :

Hypothèse : mêmes distributions ou grands échantillons.

:-(

Test statistique non paramétrique

Expérience : Exécuter les heuristiques A et B n_a (respectivement n_b) fois sur le même exemple de problème

Question : Est-ce que l'heuristique B est significativement moins bonne que l'heuristique A pour un effort de calcul donné ?

Méthode :

Hypothèse nulle :

Supposons que les deux méthodes A et B fournissent des solutions selon la même fonction de répartition

Trier les $n_a + n_b$ exécutions des deux méthodes A et B par qualité décroissante (pour un effort de calcul t)

Calculer $T = \sum$ rangs des exécutions de A

La probabilité P d'observer une $\sum$ rangs pour l'heuristique $A < T$ est donnée par

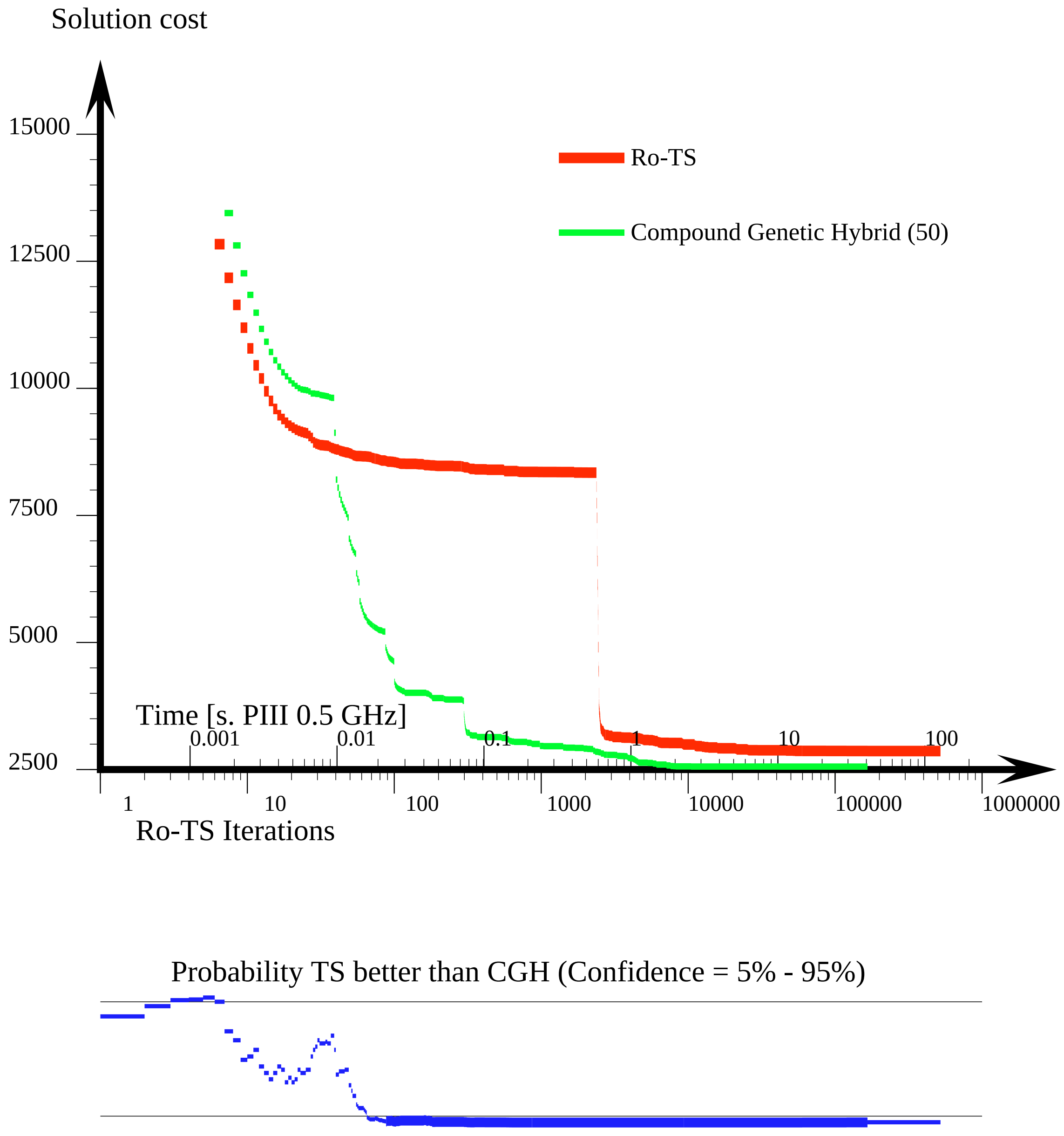
$$P = f(T, n_a, n_b), \text{ une fonction qui prend quelques pages de code}$$

Rejeter l'hypothèse nulle au seuil de confiance α si $P < 1-\alpha$

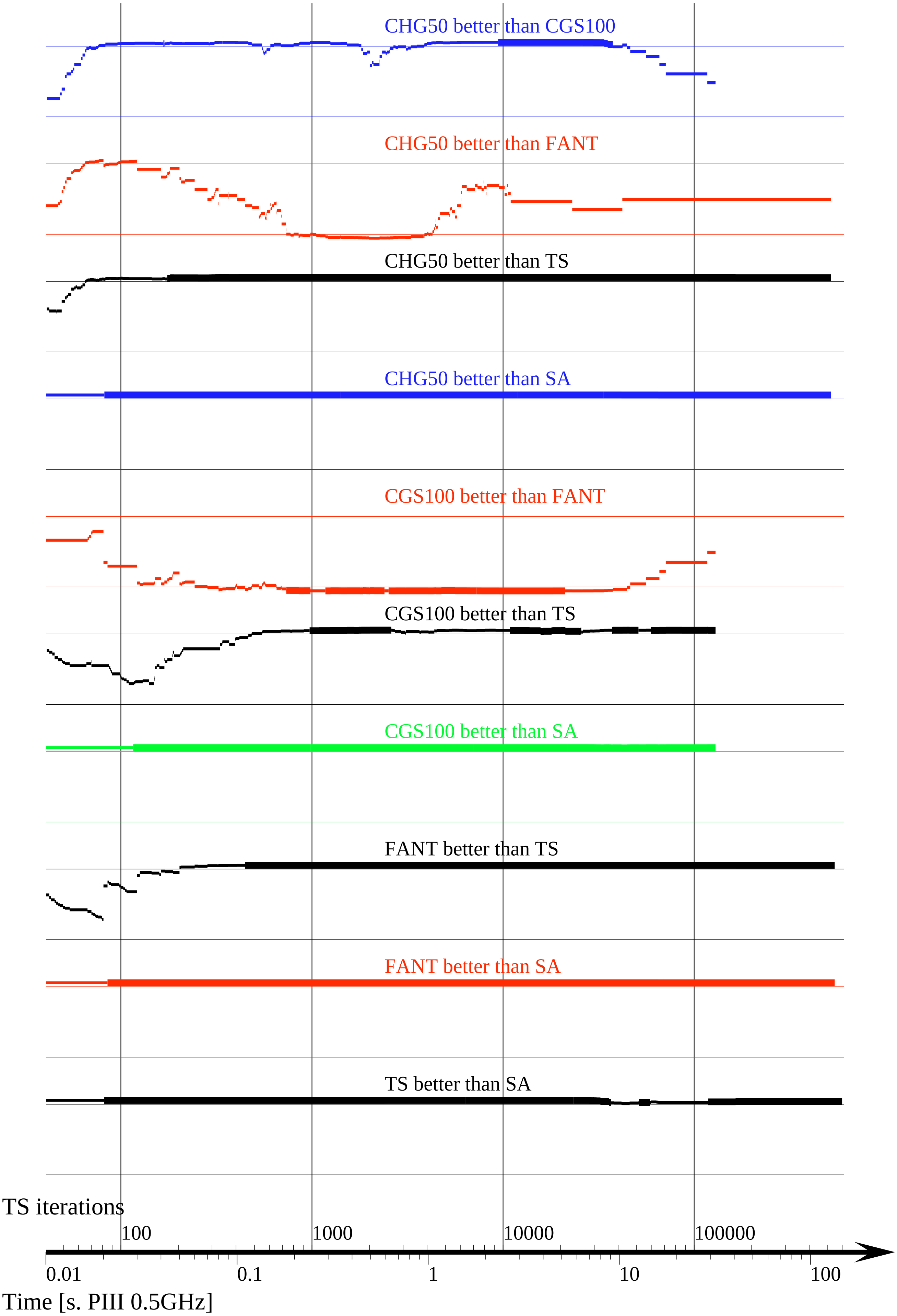
Pour une heuristique itérative, ce test doit être répété pour chaque effort de calcul t

Note : Une variante de ce test est connu sous le nom de « Test de Mann-Whitney »

Exemple de résultat du logiciel « STAMP »:



Probability diagrams (confidence : 5%-95%) on instance Tai27e01



CONCLUSIONS

Raconter toute l'histoire

Confrontation avec différentes autres méthodes.

Considérer un large ensemble d'exemples de problèmes.

Laisser tourner les méthodes itératives longtemps et utiliser des échelles logarithmiques.

Rapporter également les expériences négatives

Comprendre ce qui fait qu'une méthode marche ou ne marche pas.

Analyse graphique des résultats

Distribuer les codes des méthodes ou des résultats numériques complets.

S'assurer de la validité des résultats

Tirer des conclusions supportées par les expériences.

Aide pour le calibrage de paramètres