

# Optimisation par Colonies de Fourmis Artificielles

## Tour d'horizon

Nicolas Monmarché

Laboratoire d'Informatique



Université François Rabelais - Tours

Yrivals'07, Deuxième école d'été sur L'Évolution Artificielle à  
Latour de Carol, 30 mai-1er juin 2007

# Sommaire



- 1 Quelques mots sur les fourmis
- 2 Exploitation d'une source de nourriture : optimisation combinatoire
- 3 Recherche de nourriture : optimisation continue
- 4 Conclusion

# Fourmis réelles/artificielles

Tableau comparatif :

	<b>Fourmis réelles</b>	<b>Fourmis artificielles</b>
depuis quand ?	$100 \times 10^6$ années	15/20 ans
où ?	tout écosystème terrestre	ordinateurs, réseaux, robots
combien ?	$10^{18}$	↗
qui ?	entomologistes myrmécologues	informaticiens, ...
pourquoi ?	équilibres, régulations, évolution	optimisation, ...

## Peu d'espèces de fourmis...

Les insectes :

- 950 000 espèces décrites (probablement 8 000 000 au total) <sup>a</sup>
- les fourmis font partie de l'ordre des hyménoptères (120 000 espèces)
- 12 000 espèces de fourmis connues (estimation : 20 000) !

---

<sup>a</sup>source : World Conservation Monitoring Centre.



## ... mais beaucoup d'individus...

- une société de fourmis nomades d'Afrique (*dorylus*) peut comporter plus de 20 millions d'ouvrières
- il existe des super-colonies : plus de 300 millions d'ouvrières, 1 million de reines et 45 000 nids sur 2.7 km<sup>2</sup> (*Formica yessensis*, Japon)
- la fourmi d'Argentine (*Linepithema humile*) : 200 millions d'ouvrières et 130 000 reines à l'hectare...sur 6 000 km de côtes méditerranéennes.



## ... grandes prédatrices

- les fourmis rouges des Alpes italiennes ramassent en 200 jours 24 000 tonnes de nourriture dont 14 500 d'insectes
- une colonie de *Formica polyctena* capture 8 millions d'insectes annuellement (Allemagne)
- Pour les fourmis *Ectatomma tuberculatum* et *Ectatomma ruidum*, 260 millions de proies sont capturées annuellement par hectare (Mexique)



## ... et face à nous ?

- le poids sec des fourmis de la forêt amazonienne est environ 4 fois celui de vertébrés terrestres réunis
- La masse de fourmis sur Terre est équivalente à la masse de l'Humanité... (une fourmi pèse entre 1 et 10 mg)
- très bonne radiorésistance



# Les problèmes quotidiens des fourmis

- rechercher de la nourriture
- ⇒ navigation, repérage
- exploiter une source de nourriture
- ⇒ communication, recrutement, transport coopératif
- se reproduire et élever les jeunes
- ⇒ répartition des tâches, division du travail, tri, nursing
- défendre la colonie
- ⇒ communication, identification
- construire le nid
- ⇒ répartition des tâches, communication

Les fourmis abordent ces problématiques de façon collective :

→ Intelligence collective (Intelligence en essaim), auto-organisation

# La diversité des comportements

- l'habitat (dans la terre, dans les arbres (feuilles, branches, ...))
- la nourriture (trophalaxies, champignonistes, granivores, insectivores,...)
- la communication (visuelle, sonore, chimique : phéromones d'alarme, ...)

## Exemples :

- fourmis tisserandes
- fourmis processionnaires
- fourmis champignonistes
- fourmis parasites
- fourmis esclavagistes

# Modélisation biomimétique

Deux objectifs à la modélisation du comportement des fourmis :

- permettre aux biologistes de mieux comprendre et observer les mécanismes qui régissent ces sociétés (vérification de la capacité de prédiction du modèle)
- imaginer des nouvelles méthode de résolution de problèmes complexes (robotique, optimisation, ...)  $\Rightarrow$  Méta-heuristiques

# Modélisation biomimétique

Modélisations pour la résolution de problèmes complexes :

- recherche de nourriture
- ⇒ optimisation combinatoire, routage dans les réseaux
- division du travail
- ⇒ allocation de tâches (robotique)
- rangement des larves (tri du couvain) et des cadavres (organisation des cimetières)
- ⇒ Partitionnement de graphes, classification, tri collectif
- transport coopératif
- ⇒ robotique distribuée

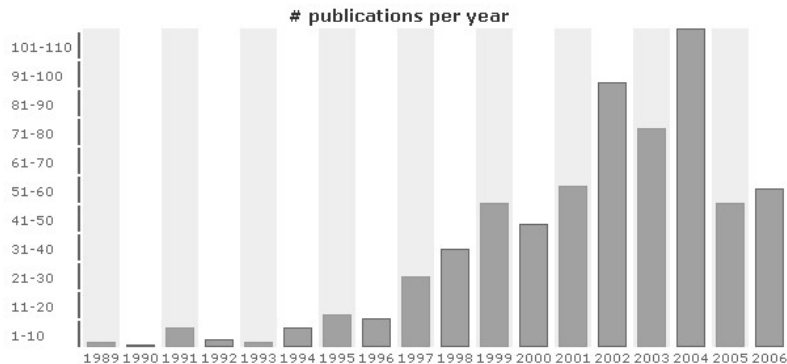
# Domaines d'origines des fourmis artificielles

- biologie théorique et comportementale
- Modélisation de systèmes complexes
- Intelligence Artificielle Distribuée (IAD)
- Systèmes Multi-Agents (SMA)
- Robotique autonome
- Optimisation stochastique par population de solutions

## Remarque

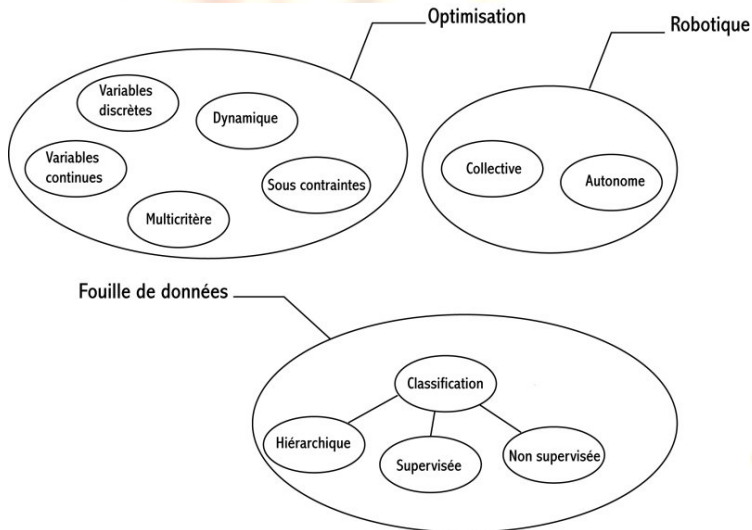
Contrairement à ce que l'on peut trop rapidement penser, une fourmi artificielle ne fait pas que de l'optimisation...

# Évolution des publications sur les fourmis artificielles



[<http://www.hant.li.univ-tours.fr/artantbib/artantbib.php>]

# Domaines d'application

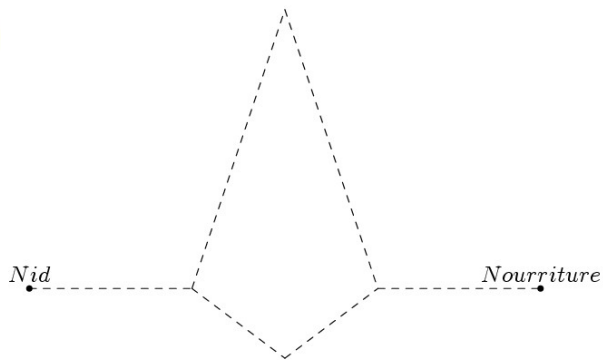


# Exemple n° 1 : exploitation d'une source de nourriture

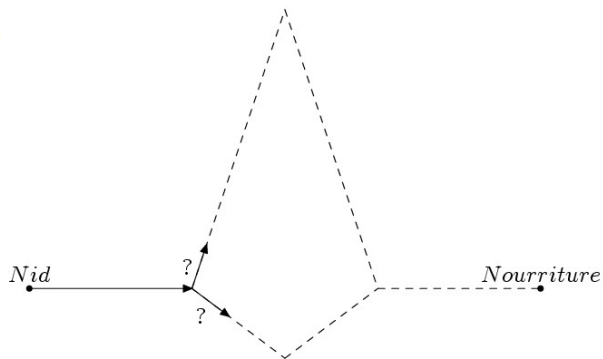
communication indirecte : marquage par phéromones  
(stimergie [43])

→ recrutement de masse

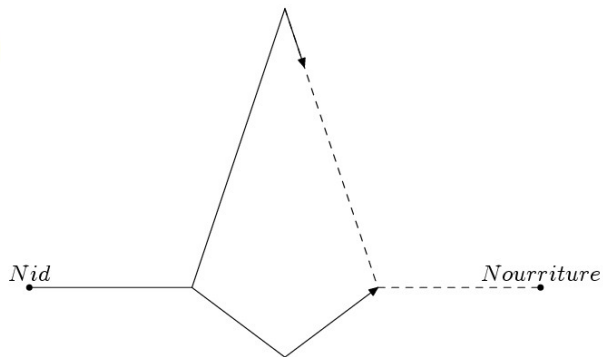
- Expériences des 2 ponts de Deneubourg et Aron sur des fourmis d'Argentine *Linepithaema humile*



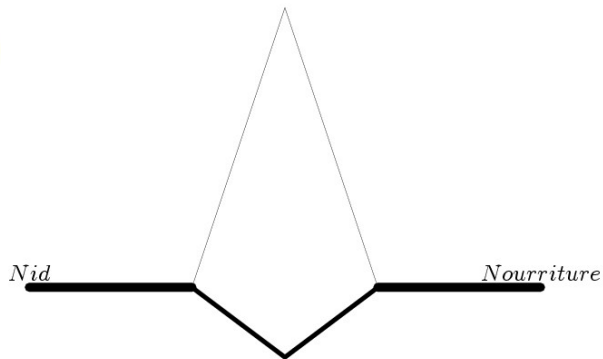
À la sortie du nid : 2 chemins possibles.



Au point de choix : les fourmis n'ont aucune vision globale.

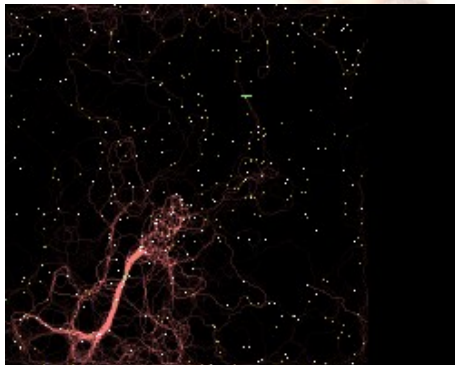


Le chemin le plus court reçoit plus de phéromones pour une durée donnée.



Le choix des fourmis est orienté par les phéromones : renforcement auto-catalytique.

# Simulation



Ant Demo by Mark Wodrich

Ants : 400

Land : 200 x 200

Land Mode : Spherical Mapping

Evaporation : 0.996

Food Strength : 100

Food Strength Decay : 0.994

Nest Strength : 100

Nest Strength Decay : 0.996

Sniff Radius : 8

Sniff Angle : 45

Time : 0:0.29

Démonstration (Mark Wodrich)

# Simulation



Démonstration (Mark Wodrich)

# Simulation



Démonstration (Mark Wodrich)

# Simulation



Démonstration (Mark Wodrich)

# Simulation



Démonstration (Mark Wodrich)

# Simulation



Démonstration (Mark Wodrich)

# Simulation



Démonstration (Mark Wodrich)

# Simulation



Démonstration (Mark Wodrich)

# Conclusion

Règles locales de déplacement :

- déposer des phéromones sur le chemin emprunté
- suivre le chemin le plus marqué (en probabilité)

Règles globales d'évolution :

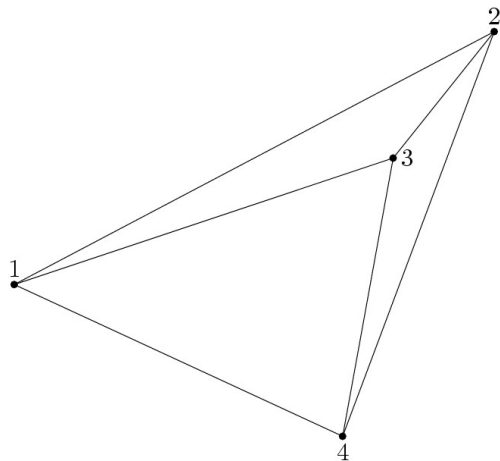
- les phéromones s'évaporent à la bonne vitesse

⇒ Découverte du plus court chemin

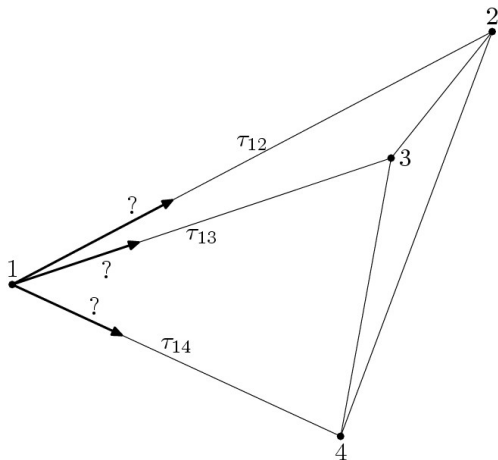
Traduction...optimisation combinatoire

# Fourmis artificielles/Optimisation combinatoire

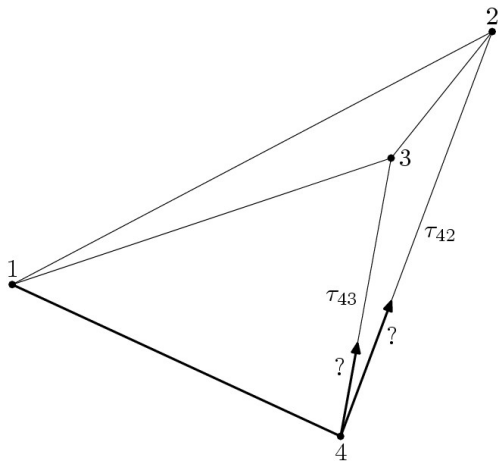
## Recherche du plus court chemin hamiltonien dans un graphe



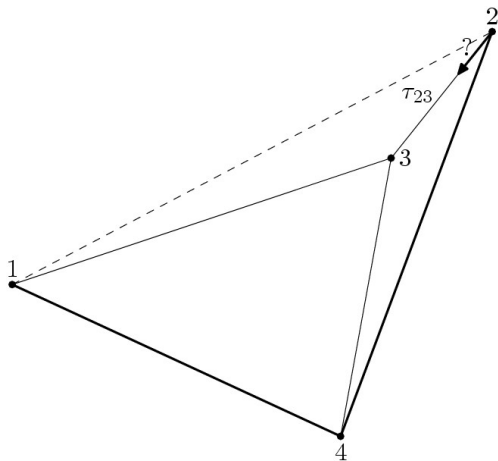
4 sommets, le graphe est complet, et la fourmi est en 1.



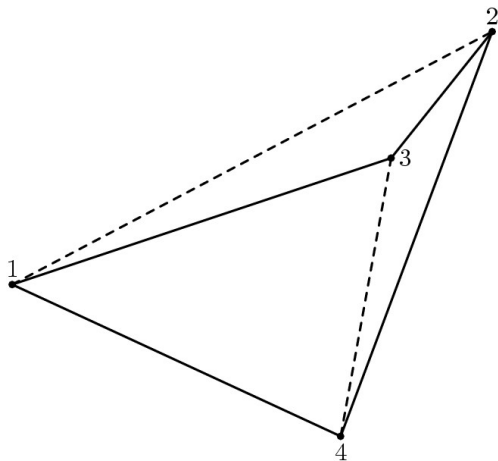
3 choix, dirigés par les valeurs  $\tau_{ij}$  = phéromones artificielles.



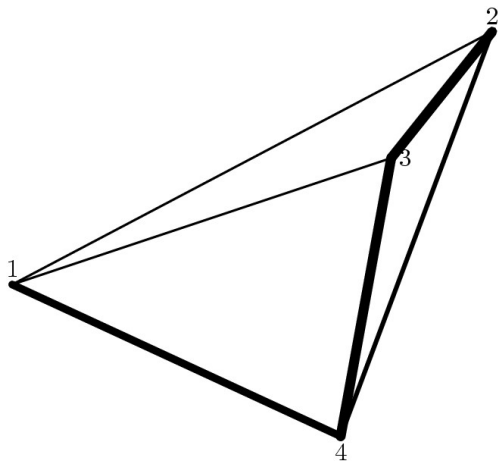
Le chemin partiel est (1,4).



Le chemin partiel est  
(1, 4, 2).



Le chemin complet est  
(1, 4, 2, 3)



Quantités de phéromones  
« numériques »

- Démonstration (A. Dutot)

# Meta heuristique ACO

**Recherche du plus court chemin hamiltonien dans un graphe**

⇔ Problème du voyageur de commerce.

**Différences par rapport aux fourmis réelles :**

- mémorisation des sommets traversés
- les phéromones sont déposées après la construction de la solution
- la vitesse de la fourmi n'est pas constante : le renforcement est alors proportionnel à la qualité de la solution construite

Nombreuses variantes :

- prise en compte de la distance entre les sommets pour le déplacement
- dans une population de  $n$  fourmis seule la meilleure dépose des phéromones
- mécanismes d'exploration/exploitation
- hybridation avec des heuristiques de recherche locale
- prise en compte de la somme des phéromones sur le chemin partiel
- ...

Initialiser les phéromones pour chaque arc à la valeur  $\tau_0$   
Placer aléatoirement les  $m$  fourmis sur les sommets du graphe  
**for**  $t = 1$  à  $t_{\max}$  **do**  
    **for**  $k = 1$  à  $m$  **do**  
        Construire le chemin  $S^k(t)$  en tenant compte des  
        phéromones ( $\tau_{ij}$ ) et de la désirabilité ( $\eta_{ij}$ ) de chaque arc  
        Calculer le coût de la solution  $S^k(t)$   
    **end for**  
    **if** une meilleure solution a été trouvée **then**  
        Mettre à jour la meilleure solution trouvé :  $S^+$   
    **end if**  
    **for all** arc  $(i, j)$  **do**  
        Mettre à jour la quantité de phéromone  $\tau_{ij}$   
    **end for**  
**end for**

Ant Colony System [28] :

- règle de transition : la fourmi placée en  $i$  choisit la ville  $j$  telle que :

$$j = \begin{cases} \arg \max_{l \in J_k(i)} \{ [\tau_{il}(t)] \cdot [\nu_{il}]^\beta \} & \text{si } q \leq q_0 \\ J & \text{sinon} \end{cases} \quad (1)$$

$q = \mathcal{U}([0, 1])$ ,  $q_0 \in [0, 1]$ ,  $\nu_{ij} = 1/d(i, j)$  (visibilité),  
 $J_k(i) = \{\text{villes restant à visiter}\}$ .  $J$  est choisie avec la probabilité :

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)] \cdot [\nu_{ij}]^\beta}{\sum_{l \in J_k(i)} [\tau_{il}(t)] \cdot [\nu_{il}]^\beta} \quad (2)$$

où  $\beta$  est un paramètre

- à chaque pas la fourmi dépose des phéromones :

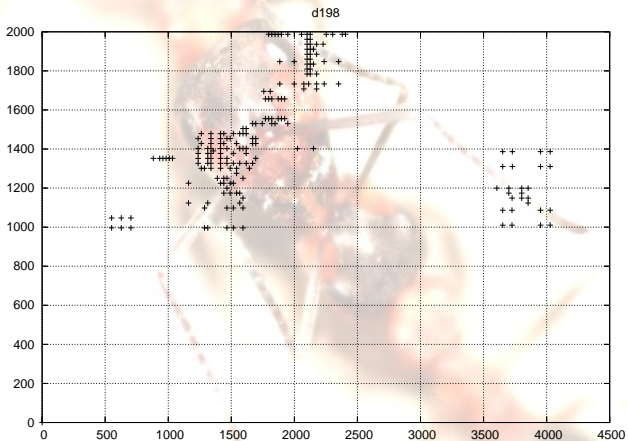
$$\tau_{ij} \leftarrow (1 - \xi)\tau_{ij} + \xi\tau_0 \quad (3)$$

- la mise à jour des phéromones n'est faite qu'à partir du meilleur chemin généré ( $C^+$ ) :

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \rho \frac{1}{L^+} \quad \forall (i, j) \in C^+ \quad (4)$$

- des listes de villes candidates sont utilisées pour accélérer le processus de construction d'un chemin ;
- une heuristique locale est utilisée pour améliorer les solutions générées par les fourmis (2-opt ou 3-opt) ;

Exemple : PVC « euclidien » à 198 villes (d198) :



# Quelques résultats

Ant Colony System [28] :

Résultats : PVC symétrique

problème	ACS+3-opt		STSP (AG)	
	best	avg.	best	avg
d198	<b>15780</b>	15781.7	<b>15780</b>	<b>15780</b>
lin318	<b>42029</b>	<b>42029</b>	<b>42029</b>	<b>42029</b>
att532	27693	27718.2	<b>27686</b>	<b>27693.7</b>
rat783	8818	8837.9	<b>8806</b>	<b>8807.3</b>

# Ant Colony Optimisation (ACO)

Problème	Références
Travelling Salesman	[14, 29, 25, 15, 16, 36, 26, 30, 37, 9, 27, 28, 74, 75, 76, 11, 73, 4, 46]
Quadratic Assignment	[56, 41, 78, 52, 53, 79, 66, 40, 55, 72]
Jobshop Scheduling	[17]
Bin Packing	[6, 32]
Vehicle Routing	[10, 12, 13, 8, 39]
Network Routing	[20, 69, 23, 21, 23, 22, 7, 50, 83, 1, 47]
Bus Driver Scheduling	[33]
Sequential Ordering	[38]
...	...

# Ant Colony Optimisation (ACO)

Problème	Références
Graph Coloring	[18]
Frequency Assignment	[53, 54]
Shortest common supersequence	[58, 59]
Flowshop Scheduling	[71, 80]
Multi-objectif	[57, 24, 44, 45]
Virtual Wave Length Path routing	[62]
Multiple Knapsack	[51]
Single Machine	[3, 34, 35]
Recognizing Hamiltonian Graphs	[81]
Total Weighted Tardiness	[19]
Constraint Satisfaction	[65, 70, 68, 67]
Dynamic graph search	[82]
Best elimination sequence in BN	[42]
Answer Set Programming	[63, 64]

# Exemple décortiqué : l'optimisation de clavier virtuel

Description du problème :



# L'optimisation de clavier virtuel

Description du problème :

- Étant donné un type de textes habituellement tapés (courriel/programmation/...) on veut obtenir le clavier virtuel qui minimisera l'effort à fournir/la fatigue (crucial dans le cas d'une personne lourdement handicapée)
- À chaque tâche, son clavier
- Les tâches peuvent évoluer dans le temps

# Hypothèses

## Hypothèses de travail :

- l'utilisateur ne tape qu'une touche à la fois (en déplaçant un pointeur)
- l'effort visuel de recherche d'un symbole est moins important que le déplacement pour l'atteindre

## Hypothèses simplificatrices :

- pour étudier le problème on peut se limiter à un sous-ensemble de symboles (par ex pas de majuscules)
- la fatigue est uniquement liée à la distance que l'utilisateur doit faire parcourir à son pointeur
- on se limite aux claviers rectangulaires  $m \times n$

# Modélisation

Clavier de taille  $12 \times 3$ .

	0	1	2	3	4	5	6	7	8	9	10	11
0	(	a	z	e	r	t	y	u	x	b	:	)
1	!	i	o	p	q	s	d	f	c	n	.	;
2	*	,	g	h	j	k	l	m	w	v	?	

- une solution (*i.e.* un clavier) est une séquence de touches de longueur  $< m \times n$

# Évaluation

Étant donné le jeu de caractères suivant :

azertyuiopmlkjhgfdsqwxcvbn

,;:!.?&"'(-\_)][\$\* 1234567890{}

# Évaluation

On veut minimiser l'effort de frappe de textes comme celui-ci :

```
#include <stdio.h>
#define dim 100
typedef int composante;
void lecture(composante *t,int *nb) {
    int i;
    puts("nombre de valeurs à entrer ? ");
    scanf("%d",nb);
    for(i=1;i<=*nb;i++) {
        printf("%dième valeur : ",i);
        scanf("%d",t++);
    }
}
```

[...]

# Évaluation

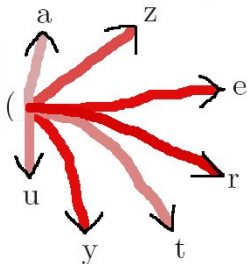
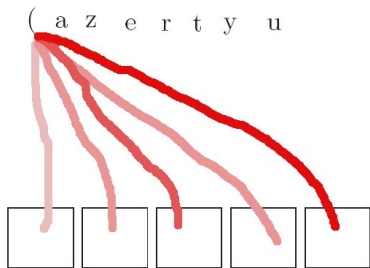
$$Q(\mathcal{K}, S) = \frac{100}{|S|} \sum_{i=1}^{|S|-1} d_{\varepsilon}(s[i], s[i+1]) \quad (5)$$

- $S = \{s[1], s[2], \dots, s[|S|]\}$  est une séquence de symbole (un texte) de longueur  $|S|$
- $d_{\varepsilon}(x, y)$  distance euclidienne entre les symboles  $x$  et  $y$  de coordonnées  $(x_1, x_2)$  et  $(y_1, y_2)$  sur le clavier  $\mathcal{K}$  :

$$d_{\varepsilon}(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

# Les phéromones

- Les phéromones sont liées à la méthode de construction d'une solution
- Deux possibilités :
  - 1 les phéromones sont déposées entre chaque symbole et chaque position du clavier
  - 2 les phéromones sont déposées entre chaque couple de symbole



# Les phéromones

- Deux méthodes de construction d'une solution :
  - ① la construction d'une solution consistera à parcourir la liste des touches et à y affecter un symbole en utilisant les phéromones
  - ② la construction d'une solution consistera à passer d'un symbole à l'autre en utilisant les phéromones et à affecter les symboles choisis dans l'ordre de parcours du clavier

# Les phéromones

Intuitivement :

- la première méthode tente de découvrir la bonne position pour chaque symbole
- la deuxième méthode tente de découvrir la bonne séquence de symboles

Il y a des avantages et inconvénients dans les deux cas...

Étudions la deuxième.

## Construction d'une solution

- soit  $\mathcal{K}_k(t)$  le clavier partiellement construit au temps  $t$  et  $i$  le dernier symbole placé

La fourmi choisit le symbole  $j$  selon la probabilité suivante :

$$P^{(t)}(i, j) = P_e^{(T)} \times \frac{\tau_{i,j}^\alpha \times \eta_{i,j}}{\sum_{l \in N(\mathcal{K}_k(t))} \tau_{i,l}^\alpha \times \eta_{i,l}} + (1 - P_e^{(T)}) \times \begin{cases} 1 & \text{si } j = \arg \max_{l \in N(\mathcal{K}_k(t))} \{\tau_{i,l}^\alpha \times \eta_{i,l}\} \\ 0 & \text{sinon} \end{cases}$$

- où  $j \in N(\mathcal{K}_k(t))$  (ensemble des symboles restant à placer dans  $\mathcal{K}_k(t)$ )
- $\alpha$  est un paramètre
- $P_e^{(T)}$  et la probabilité d'exploration/exploitation

# Exploration/exploitation

$P_e^{(T)}$  et la probabilité d'exploration/exploitation (souvent constante, mais ici nous choisissons de la faire varier au cours des itérations :  
 $P_e^{(T)} = 0.8(T/T^{\max})$  pour accroître l'exploitation au fur et à mesure des itérations)

# Mise à jour des phéromones

$$\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j} + \rho \sum_{k=1}^A \Delta_{i,j}^k \quad (6)$$

- $\rho$  coefficient d'évaporation
- $A$  est le nombre de fourmis

## Mise à jour des phéromones

- $\Delta$  est calculé pour favoriser les bons choix des fourmis :

$$\Delta_{i,j}^k = \begin{cases} \frac{\min_{l \in \{1, \dots, A\}} \{Q(\mathcal{K}_l, S)\}}{Q(\mathcal{K}_k, S) \times (\text{Rank}(\mathcal{K}_k) + 1)} & \text{si le symbole } j \text{ suit le symbole } i \text{ sur } \mathcal{K}_k \\ 0 & \text{sinon} \end{cases} \quad (7)$$

- $\Delta_{i,j}^k$  est incrémenté chaque fois qu'une fourmi a utilisé le symbole  $j$  après le symbole  $i$  dans son clavier et proportionnellement à la qualité relative de son clavier et de son rang dans un ordre décroissant de qualité ( $\text{Rank}(\mathcal{K}_{\text{best}}) = 1$ ).
- Si aucune fourmi n'est passée de  $i$  à  $j$ , la quantité de phéromone décroît (évaporation seule)
- enfin, les phéromones sont conservées dans l'intervalle  $[\tau_{\min}; \tau_{\max}]$ .

## Désirabilité $\eta_{i,l}$

- On peut utiliser l'inverse de la co-occurrence successive des symboles dans le texte qui sert de référence (plus 2 symboles se suivent fréquemment, plus la transition de l'un à l'autre par une fourmi est probable)
- on pourrait utiliser des mesures plus générales (suivant la langue...)

# Expérimentations

Que peut-on observer ?

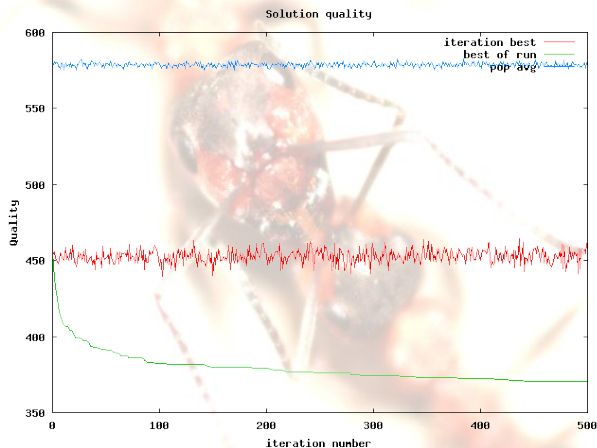
- l'évolution au cours des itérations de l'évaluation de la meilleur solution
- l'évolution au cours des itérations de l'évaluation moyenne de la colonie
- les valeur min, max et moyenne des phéromones
- la quantité d'information stockée dans la matrice de phéromone (l'entropie)
- ...

# Expériences

Conditions d'expérience :

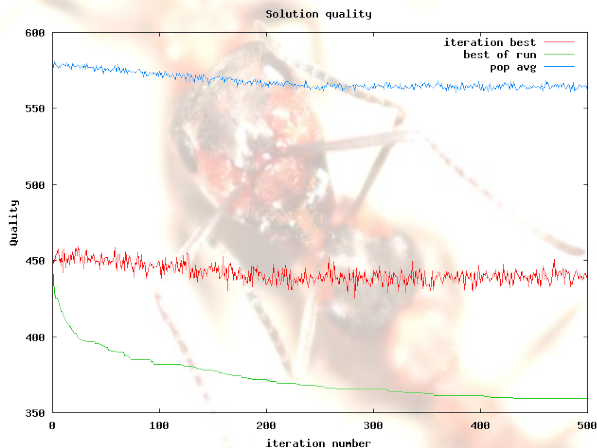
- Texte  $S$  : programmation C (longueur : 1 943 symboles)
- bornes des pheromones :  $[\tau_{\min}; \tau_{\max}] = [0.1; 0.9]$ ,
- Pheromone initiale :  $\tau_{i,j}^0 = \tau_{\max}$  si  $i \neq j$  et  $\tau_{i,i}^0 = 0.0$ ,
- taux d'évaporation :  $\rho = 0.01$ ,
- nb de fourmis = nombre de symboles à placer sur le clavier (57 ici),
- nombre d'itérations : 500 (chaque fourmi construit 500 claviers).

# Expériences



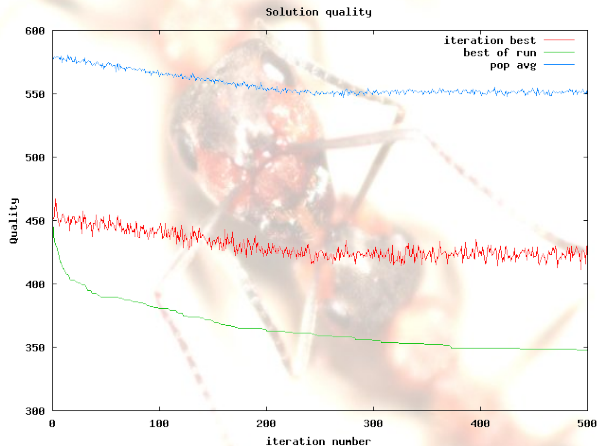
Évolution de la qualité (fitness), avec  $\alpha = 0.0$  (pas de phéromones)

# Expériences



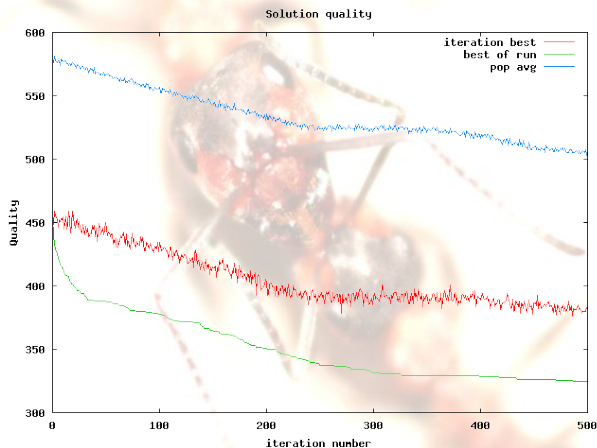
Évolution de la qualité (fitness), avec  $\alpha = 0.5$

# Expériences



Évolution de la qualité (fitness), avec  $\alpha = 1.0$

# Expériences



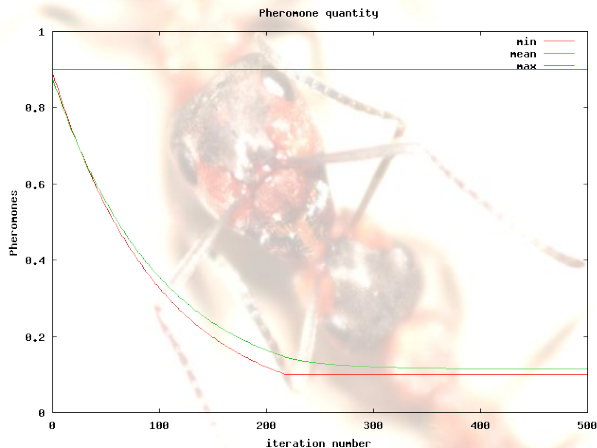
Évolution de la qualité (fitness), avec  $\alpha = 2.0$

# Expériences

Pourquoi ce fléchissement de l'amélioration de la qualité vers 250 itérations ?

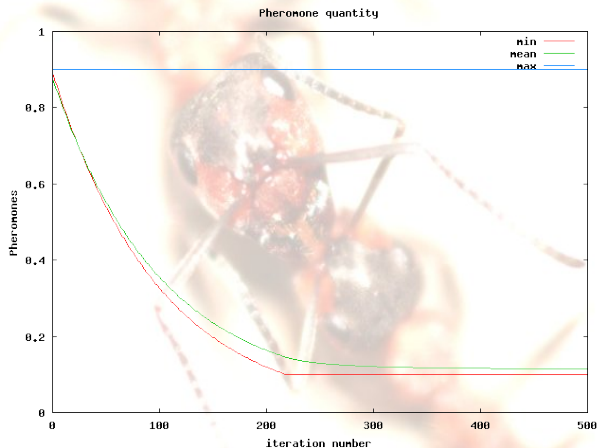
Observons d'autres mesures...

# Expériences



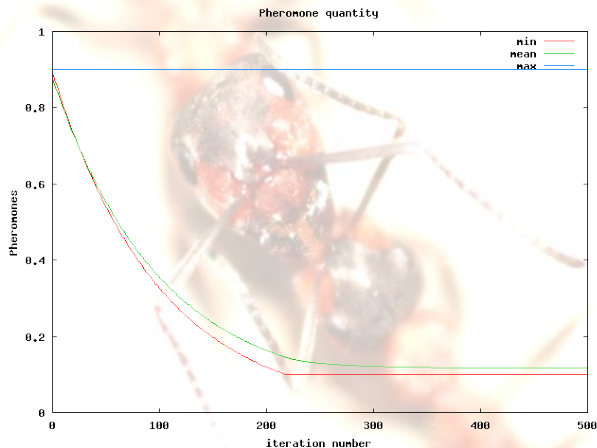
Évolution des quantités de phéromones (min, max et moyenne), avec  $\alpha = 0.0$

# Expériences



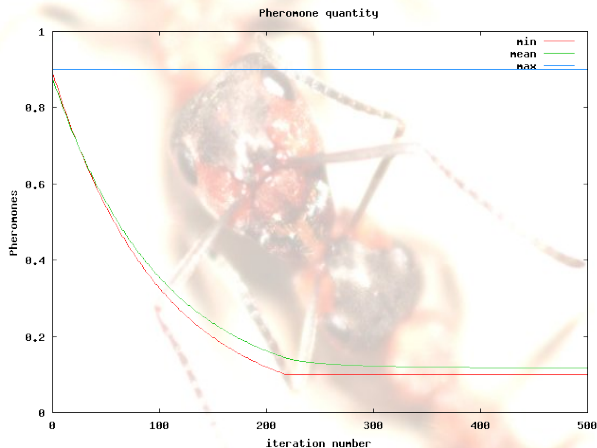
Évolution des quantités de phéromones (min, max et moyenne), avec  $\alpha = 0.5$

# Expériences



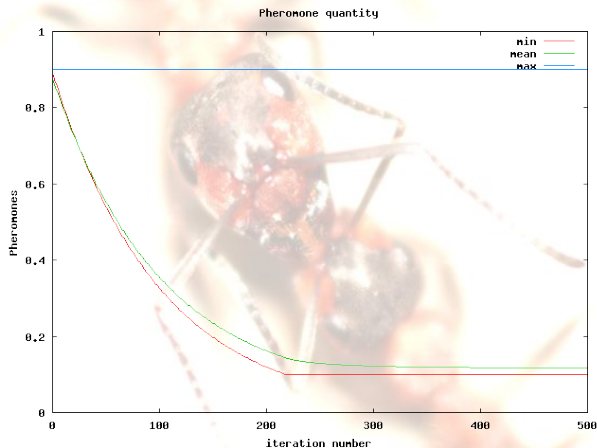
Évolution des quantités de phéromones (min, max et moyenne), avec  $\alpha = 1.0$

# Expériences



Évolution des quantités de phéromones (min, max et moyenne), avec  $\alpha = 2.0$

# Expériences



Évolution des quantités de phéromones (min, max et moyenne), avec  $\alpha = 2.0$

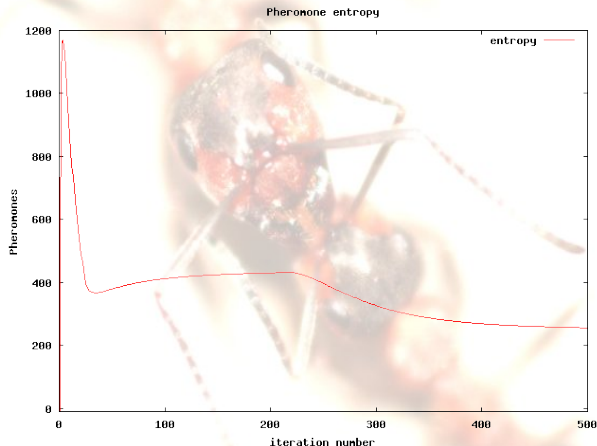
# Expériences

⇒ Quelque soit  $\alpha$ , les phéromones semblent évoluer de la même façon.

Regardons l'entropie :

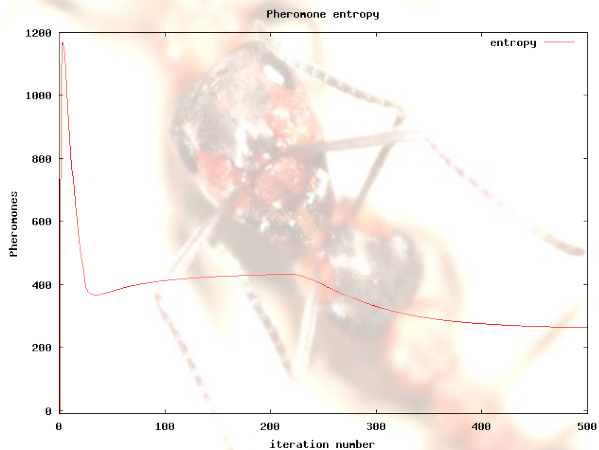
$$\text{Ent}(T) = - \sum_{i=1}^m \sum_{j=1}^n \frac{|\tau_{i,j} - \bar{\tau}|}{\max_{u,v} \{\tau_{u,v}\} - \min_{u,v} \{\tau_{u,v}\}} \log \frac{|\tau_{i,j} - \bar{\tau}|}{\max_{u,v} \{\tau_{u,v}\} - \min_{u,v} \{\tau_{u,v}\}} \quad (8)$$

# Expériences



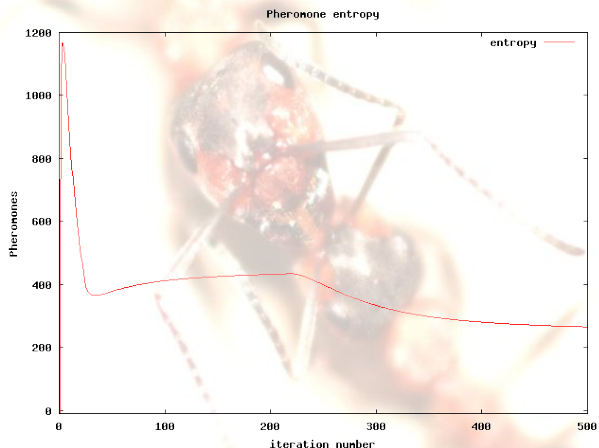
Évolution de l'entropie de phéromones, avec  $\alpha = 0.0$

# Expériences



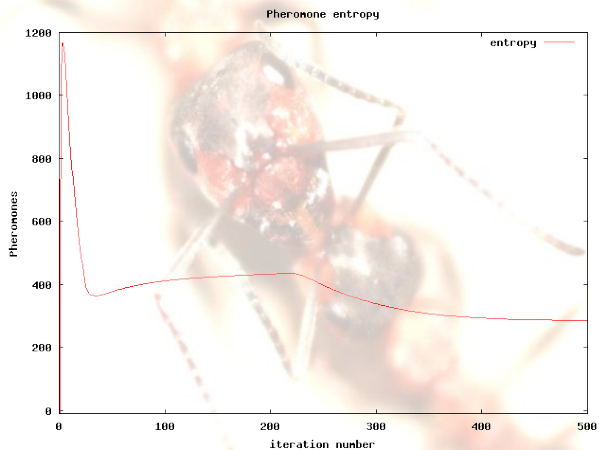
Évolution de l'entropie de phéromones, avec  $\alpha = 0.5$

# Expériences



Évolution de l'entropie de phéromones, avec  $\alpha = 1.0$

# Expériences

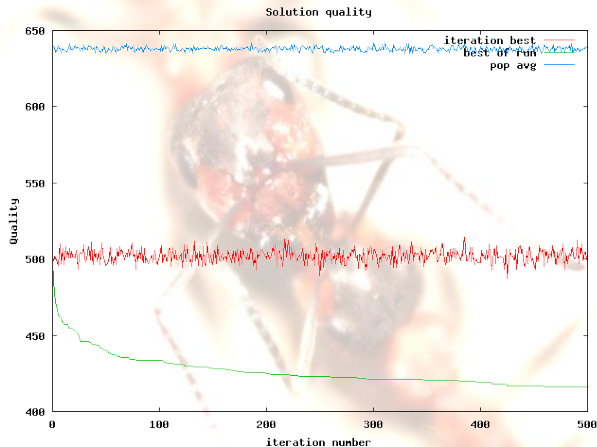


Évolution de l'entropie de phéromones, avec  $\alpha = 2.0$

# Expériences

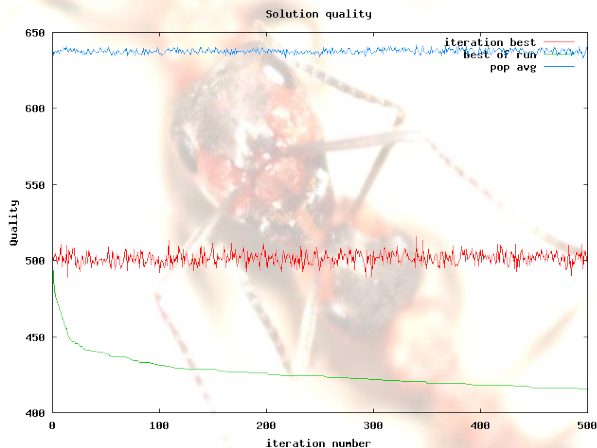
Finalement, les phéromones servent-elles à quelque chose ?  
Enlevons la désirabilité pour vérifier si les phéromones seules peuvent faire progresser la qualité.

# Expériences



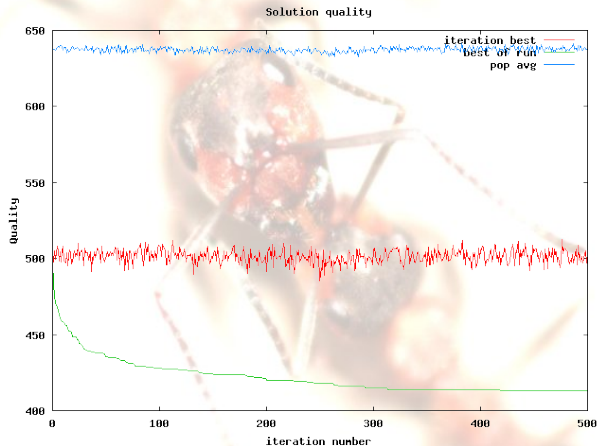
Évolution de la qualité (fitness), avec  $\alpha = 0.0$  (pas de phéromones, pas de désirabilité)

# Expériences



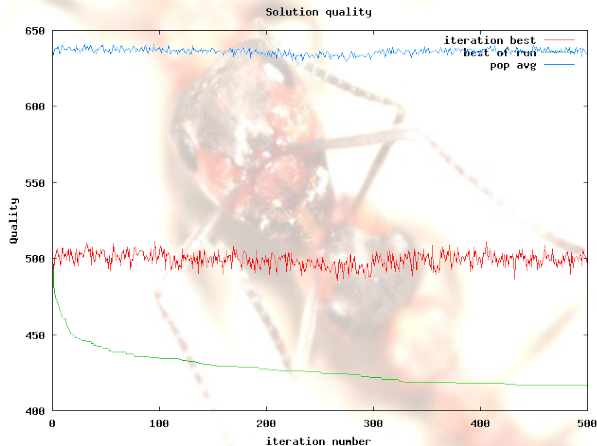
Évolution de la qualité (fitness), avec  $\alpha = 0.5$  (pas de désirabilité)

# Expériences



Évolution de la qualité (fitness), avec  $\alpha = 1.0$  (pas de désirabilité)

# Expériences



Évolution de la qualité (fitness), avec  $\alpha = 2.0$  (pas de désirabilité)

# Conclusion

Reste à faire :

- il faut contrer la stabilisation des phéromones : quand stabilité, remettre les phéromones à leur valeur initiale (restart) (MMAS)
- d'autres paramètres peuvent être étudiés ( $\rho, \dots$ )
- qualité intrinsèques des solutions produites ?
- comparaison avec d'autres méthodes.
- capacité d'intégration de changements dans les textes tapés (problème dynamique)

# Convergence de ACO

## Graph-Based Ant System [49, 48]

### Hypothèses :

- 1 il n'y a qu'une seule solution optimale (notée  $w^*$ ) pour l'instance du problème considérée
- 2 pour chaque arc  $(k, l) \in w^*$  on a :  $\nu_{kl}(u) > 0$
- 3 soit  $f^* = f^*(m)$  la meilleure évaluation trouvée au cours des cycles  $1, \dots, m - 1$   
→ seuls les chemins au moins aussi performants que  $f^*$  reçoivent un renforcement (stratégie élitiste).

## Théorème :

Soit  $P_m$ , la probabilité qu'un agent particulier parcourt le chemin optimal au cycle  $m$ , les deux assertions suivantes sont valides :

- pour tout  $\varepsilon > 0$  et en fixant les paramètres  $\rho$  et  $\beta$ , et en choisissant un nombre  $N$  de fourmis suffisamment grand, on a  $P_m \geq 1 - \varepsilon$  pour tout  $m \geq m_0$  ( $m_0$  est un entier dépendant de  $\varepsilon$ )
- pour tout  $\varepsilon > 0$  et en fixant les paramètres  $N$  et  $\beta$ , et en choisissant un facteur d'évaporation  $\rho$  suffisamment proche de 0, on a  $P_m \geq 1 - \varepsilon$  pour tout  $m \geq m_0$  ( $m_0$  est un entier dépendant de  $\varepsilon$ )

## Principe :

- état du système = {phéromones}  $\cup$  {chemins partiels}  $\cup$   $\{f^*(m)\}$
- les transitions entre états du système = processus markovien

# Rapprochements avec les AG

Matrice de phéromones  $\sim$  distribution de probabilité dans les AE  
PBIL [2] et BSC [77].

« ACO-canonique » [60] :

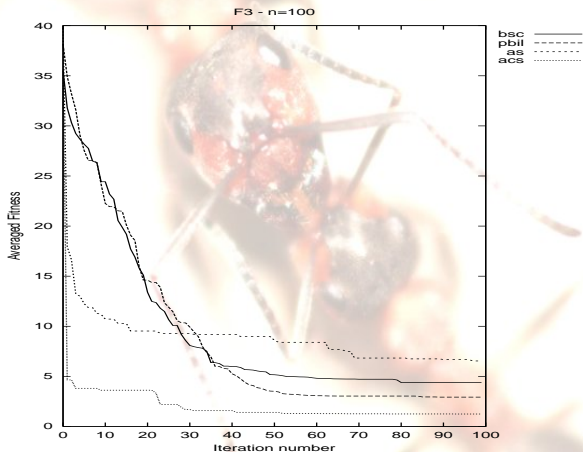
- $V = (p_1, \dots, p_l)$ , avec  $p_i \in [0, 1]$  : probabilité de générer un « 1 »,
- $P = (s_1, \dots, s_n)$ , avec  $s_i \in S$  : population de  $n$  chaînes binaires



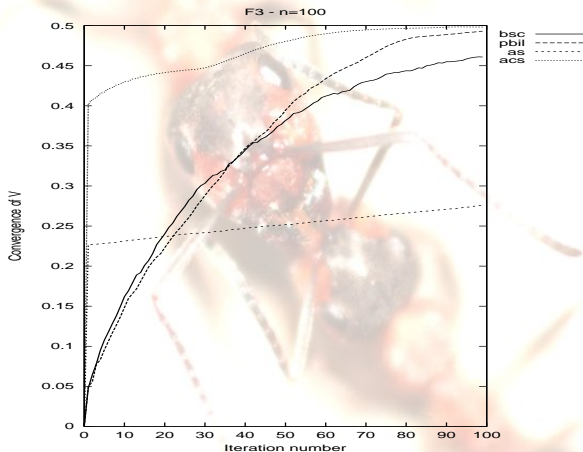
# Algorithme commun

- 1 Initialisation  $V = (p_1, \dots, p_l) \leftarrow (0.5, \dots, 0.5)$
- 2 Générer  $P = (s_1, \dots, s_n)$  suivant  $V$
- 3 Évaluer  $f(s_1), \dots, f(s_n)$
- 4 Mise à jour de  $V$  selon  $(s_1, \dots, s_n)$  et  $f(s_1), \dots, f(s_n)$
- 5 Si condition d'arrêt vérifiée **Alors** Stop **Sinon** Aller en 2

$$F_3(\mathbf{x}) = 50 + \sum_{i=1}^5 ((x_i - 1)^2 - 10 \cos(2\pi(x_i - 1))) \quad (1)$$



<sup>1</sup>avec  $x_i \in [-5.12, 5.11]$  et  $l = 10$ .

Convergence de  $V$  pour  $F_3$  :

# Les fourmis de l'espèce *Pachycondyla apicalis*



INBIOCR1002278968, La Selva. Image by J. Longino Mar'99

# recherche de nourriture par les fourmis de l'espèce *Pachycondyla apicalis*

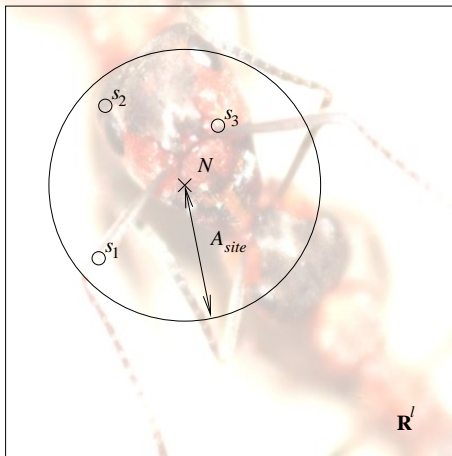


## Caractéristiques [61] :

- pas de marquage avec des phéromones
- mémorisation du site de la dernière capture
- recrutement en tandem
- déménagements du nid

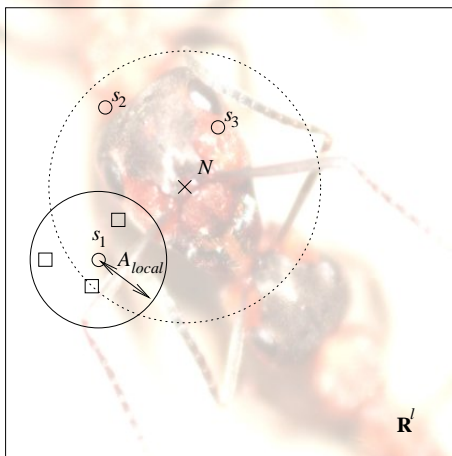
## → Algorithme API : Optimisation

## Comportement individuel :



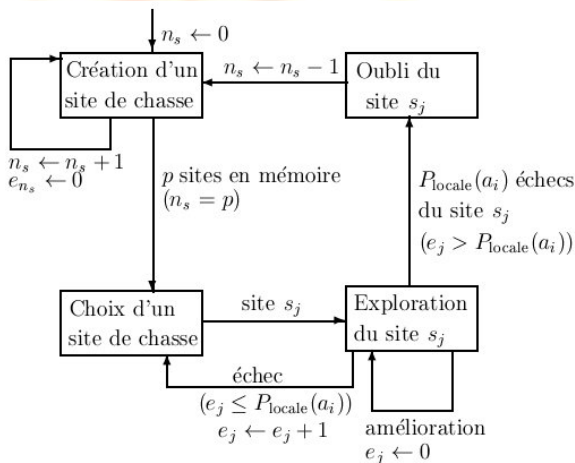
Création des sites de chasses  $s_1$ ,  $s_2$  et  $s_3$ .

## Comportement individuel :

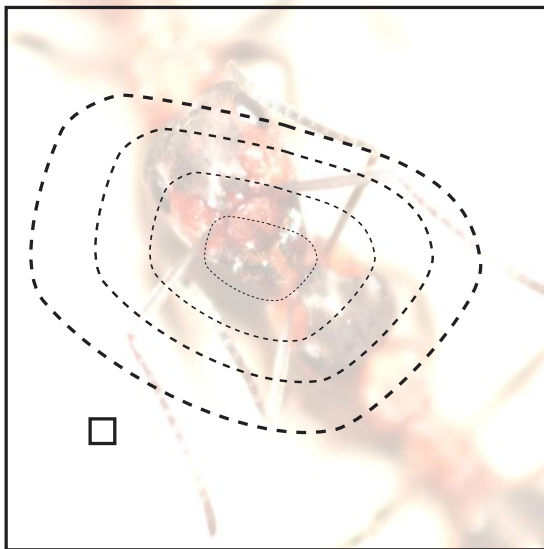


Exploration locale autour du site  $s_1$ .

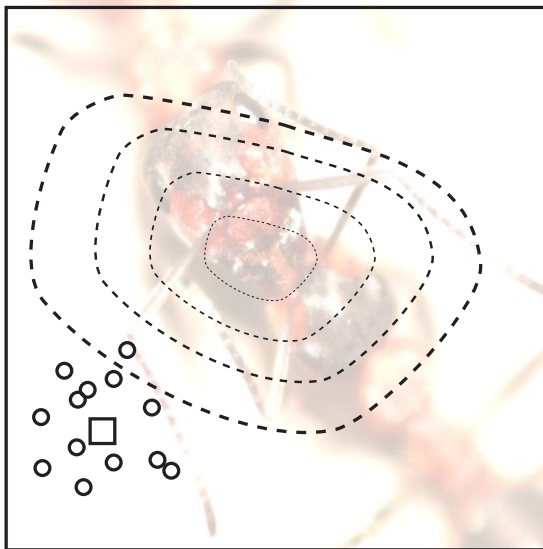
## Comportement individuel :



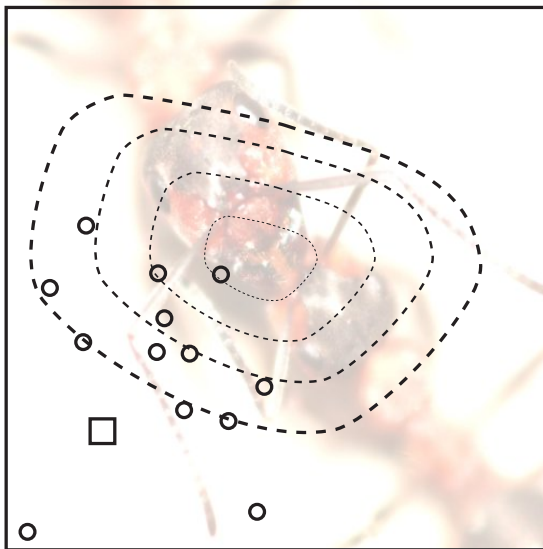
## Comportement collectif :



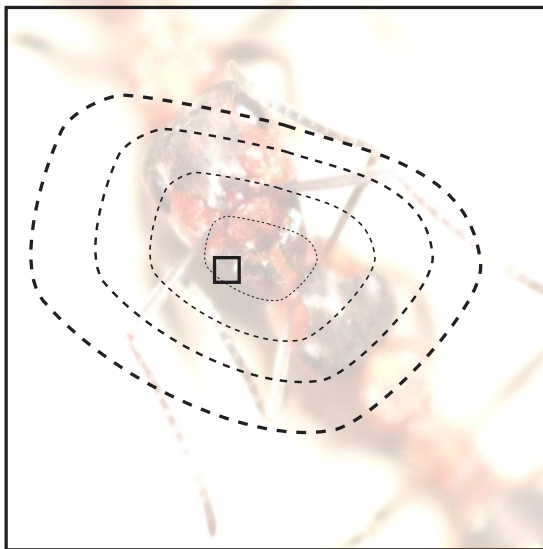
## Comportement collectif :

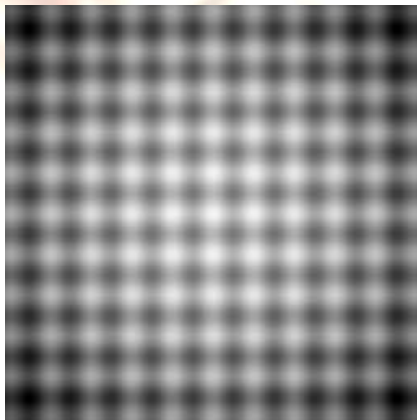


## Comportement collectif :



## Comportement collectif :

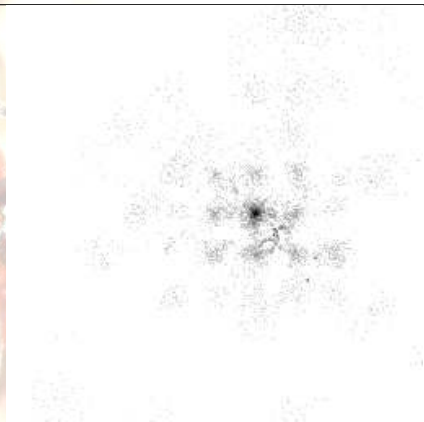




fonction de Griewank



population homogène



population hétérogène

## Applications :

- apprentissage des poids d'un RNA
- optimisation combinatoire : PVC
- recherche d'information sur le web (Thèse F. Picarougne)
- apprentissage de chaînes de Markov cachées pour la reconnaissance d'images (Thèse S. Aupetit) :
  - ▶ hybridation avec l'algorithme de Baum-Welch
  - ▶ comparaison avec les AG et une PSO

Autres méthodes de fourmis pour l'optimisation numérique : [5, 31]

# Conclusion

- bien que les fourmis artificielles soient le plus souvent des agents réactifs, des développements plus « cognitifs » sont possibles (par exemple en robotique) ou le collectif n'est pas la seule force
- de nombreux comportements des fourmis sont encore à développer sous forme de méta-heuristique, de robots...
- la réserve - potentielle - de 20 000 espèces laisse du travail...

# Conclusion

Pour aller plus loin :

- Bibliographie sur les fourmis artificielles :  
[<http://www.hant.li.univ-tours.fr/artantbib/artantbib.php>]
- Exposition « Les insectes artificiels » (muséum d'histoire naturelle de la ville de Tours) : du 1er octobre 2005 au 20 août 2006 : **Trop tard!!**
- Evolution Artificielle 2007 : Tours, du 29 au 31 octobre 2007  
<http://ea07.hant.li.univ-tours.fr>



A. Adamatzky and O. Holland.

Reaction-diffusion and ant-based load balancing of communication networks.  
*Kybernetes*, 31(5-6) :667–681, 2002.



S. Baluja and R. Caruana.

Removing the genetics from the standard genetic algorithm.

In A. Prieditis and S. Russell, editors, *Proceedings of the Twelfth International Conference on Machine Learning (ICML)*, pages 38–46. Morgan Kaufmann, San Mateo, California, 1995.



A. Bauer, B. Bullnheimer, R. Hartl, and C. Strauss.

An Ant Colony Approach for the Single Machine Total Tardiness Problem.

POM Working Paper 5/99, Department of Management Science, University of Vienna, 1999.



Leonora Bianchi, Luca Maria Gambardella, and Marco. Dorigo.

Solving the Homogeneous Probabilistic Traveling Salesman Problem by the ACO Metaheuristic.

In M. Dorigo, G. Di Caro, and M. Sampels, editors, *Proceedings of the Third International Workshop on Ant Algorithms (ANTS'2002)*, volume 2463 of *Lecture Notes in Computer Science*, pages 176–187, Brussels, Belgium, September 12-14 2002. Springer Verlag.



George Bilchev and Ian C. Parmee.

The Ant Colony Metaphor for Searching Continuons Design Spaces.

In T. Fogarty, editor, *Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature (PPSN VII)*, volume 993 of *Lecture Notes in Computer Science*, pages 24–39. Springer Verlag, 1995.



George Bilchev and Ian C. Parmee.

Evolutionary Metaphors for the Bin Packing Problem.

In *Fifth Annual Conference on Evolutionary Programming*, San Diego, California, USA, February 29-March 2 1996.



Eric Bonabeau, Florian Henaux, Sylvain Guerin, Dominique Snyers, Pascale Kuntz, and Guy Theraulaz.

Routing in Telecommunications Networks with “Smart” Ant-Like Agents.

Working Paper 98-01-003, Santa Fe Institute, 1998.



Bernd Bullnheimer.

*Ant Colony Optimization in Vehicle Routing.*  
Doctoral thesis, University of Vienna, January 1999.



Bernd Bullnheimer, Richard F. Hartl, and Christine Strauss.

A New Rank Based Version of the Ant System - A Computational Study.  
Working Paper 3/97, Institute of Management Science, University of Vienna, Austria, april 1997.



Bernd Bullnheimer, Richard F. Hartl, and Christine Strauss.

Applying the Ant System to the Vehicle Routing Problem.  
In *Proceedings of the second Metaheuristics International Conference (MIC'97)*, Sophia-Antipolis, France, 1997.



Bernd Bullnheimer, Richard F. Hartl, and Christine Strauss.

A New Rank Based Version of the Ant System : A Computational Study.  
*Central European Journal for Operations Research and Economics*, 7(1) :25–38, 1999.



Bernd Bullnheimer, Richard F. Hartl, and Christine Strauss.

An Improved Ant system Algorithm for the Vehicle Routing Problem.  
In Dawid, Feichtinger, and Hartl, editors, *Annals of Operations Research*, volume 89 of *Nonlinear Economic Dynamics and Control*, pages 319–328. 1999.



Bernd Bullnheimer, Richard F. Hartl, and Christine Strauss.

Applying the Ant System to the Vehicle Routing Problem.  
In S. Voss, S. Martello, I.H. Osman, and C. Roucairol, editors, *Meta-Heuristics : Advances and Trends in Local Search Paradigms for Optimization*. Kluwer, Boston, 1999.



A. Colorni, M. Dorigo, and V. Maniezzo.

Distributed Optimization by Ant Colonies.  
In F. Varela and P. Bourguine, editors, *Proceedings of the First European Conference on Artificial Life (ECAL)*, pages 134–142. MIT Press, Cambridge, Massachusetts, 1991.



A. Colorni, M. Dorigo, and V. Maniezzo.

An investigation of some properties of an “Ant algorithm”.

In R. Männer and B. Manderick, editors, *Proceedings of the Second International Conference on Parallel Problem Solving from Nature (PPSN II)*, pages 509–520. Elsevier Science, 1992.



**A. Coloni, M. Dorigo, and V. Maniezzo.**

**New Results of an Ant System Approach Applied to the Asymmetric TSP.**

In I.H. Osman and J.P. Kelly, editors, *Proceedings of the first Metaheuristics International Conference (MIC'95)*, pages 356–360, Hilton Breckenridge, Colorado, 1995. Kluwer Academic Publishers.



**A. Coloni, M. Dorigo, V. Maniezzo, and M. Trubian.**

**Ant System for Job-shop Scheduling.**

*JORBEL - Belgian Journal of Operations Research, Statistics and Computer Science*, 34(1) :39–53, 1994.



**D. Costa and A. Hertz.**

**Ants Can Colour Graphs.**

*Journal of the Operational Research Society*, 48 :295–305, 1997.



**M. den Besten, T. Stützle, and M. Dorigo.**

**Ant Colony Optimization for the Total Weighted Tardiness Problem.**

In Marc Schoenauer, Kalyanmoy Deb, Guenter Rudolph, Xin Yao, Evelynne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI)*, volume 1917 of *Lecture Notes in Computer Science*, pages 611–?? Springer Verlag, 2000.



**Gianni Di Caro and Marco Dorigo.**

**AntNet : A Mobile Agents Approach to Adaptive Routing.**

Technical Report 97-12, IRIDIA, Université Libre de Bruxelles, Belgium, 1997.



**Gianni Di Caro and Marco Dorigo.**

**Ant Colonies for Adaptive Routing in Packet-switched Communications Networks.**

In A.E. Eiben, T. Bäck, H.-P. Schwefel, and M. Schoenauer, editors, *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature (PPSN V)*. Springer-Verlag, New York, 1998.



**Gianni Di Caro and Marco Dorigo.**

**AntNet : Distributed Stigmergetic Control for Communications Networks.**

*Journal of Artificial Intelligence Research*, 9 :317–365, 1998.



Gianni Di Caro and Marco Dorigo.

Mobile Agents for Adaptive Routing.

In *31st Hawaii International Conference on System*, Big Island of Hawaii, January 6-9 1998.



K. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, and C. Stummer.

Ant Colony Optimization in Multiobjective Portfolio Selection.

In J.P. Sousa, editor, *Proceedings of the fourth Metaheuristics International Conference (MIC'01)*, pages 243–248, Porto, Portugal, July 16-20 2001.



Marco Dorigo.

*Optimization, Learning and Natural Algorithms (in Italian)*.

PhD thesis, Politecnico di Milano, Italy, 1992.



Marco Dorigo and Luca Maria Gambardella.

A Study of some properties of Ant-Q.

In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Proceedings of the Fourth International Conference on Parallel Problem Solving from Nature (PPSN IV)*, pages 656–665. Springer-Verlag, Berlin, September 22-27 1996.



Marco Dorigo and Luca Maria Gambardella.

Ant Colonies for the Traveling Salesman Problem.

*BioSystems*, 43 :73–81, 1997.



Marco Dorigo and Luca Maria Gambardella.

Ant Colony Sytem : A Cooperative Learning Approach to the Travelling Salesman Problem.

*IEEE Transactions on Evolutionary Computation*, 1(1) :53–66, 1997.



Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni.

Positive feedback as a search strategy.

Technical Report 91-016, Politecnico di Milano, Italy, 1991.



Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni.

The Ant System : Optimization by a Colony of Cooperating Agents.  
*IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26(1) :29–41, 1996.



Johann Dréo.

*Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues. Application en génie biomédical.*

PhD thesis, Université Paris12, 2005.



Frederick Ducatelle and John Levine.

Ant Colony Optimisation for Bin Packing and Cutting Stock.

In *Proceedings of the UK Workshop on Computational Intelligence (UKCI-01)*, Edinburgh, September 2001.



P. Forsyth and A. Wren.

An Ant System for Bus Driver Scheduling.

In *7th International Workshop on Computer-Aided Scheduling of Public Transport*, Boston, August 1997.



C. Gagné, W.L. Price, and M. Gravel.

Scheduling a Single Machine with Sequence Dependent Setup Time Using Ant Colony Optimization.

Technical Report 2001-003, Faculté des Sciences de L'Administration, Université Laval, Québec, Canada, April 2001.



C. Gagné, W.L. Price, and M. Gravel.

Comparing an ACO algorithm with other heuristics for the single machine scheduling problem with sequence-dependent setup times.

*Journal of the Operational Research Society*, 53(8) :895–906, 2002.



Luca Maria Gambardella and Marco Dorigo.

Ant-Q : A Reinforcement Learning Approach to the Travelling Salesman Problem.

In A. Prieditis and S. Russell, editors, *Proceedings of the Twelfth International Conference on Machine Learning (ICML)*, pages 252–260. Morgan Kaufmann, San Mateo, California, 1995.



Luca Maria Gambardella and Marco Dorigo.

Solving Symmetric and Asymmetric TSPs by Ant Colonies.

In IEEE, editor, *Proceedings of the third IEEE International Conference on Evolutionary Computation (ICEC)*, pages 622–627, Nagoya, Japan, 1996. IEEE Press.



Luca Maria Gambardella and Marco Dorigo.

HAS-SOP : An Hybrid Ant System for the Sequential Ordering Problem.  
Technical Report 97-11, IDSIA, Lugano, Switzerland, 1997.



Luca Maria Gambardella, E. Taillard, and G. Agazzi.

MACS-VRPTW : A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows.  
In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimisation*, pages 63–76. McGraw-Hill, London, UK, 1999.



Luca Maria Gambardella, E. Taillard, and M. Dorigo.

Ant Colonies for the QAP.  
*Journal of the Operational Research Society*, 50 :167–176, 1999.



Luca Maria Gambardella, É.D. Taillard, and M. Dorigo.

Ant Colonies for the QAP.  
Technical Report 97-4, IDSIA, Lugano, Switzerland, 1997.



J.A. Gámez and J.M. Puerta.

Searching for the best elimination sequence in Bayesian networks by using ant colony optimization.  
*Pattern Recognition Letters*, 23(1-3) :261–277, 2002.



P.P. Grassé.

La reconstruction du nid et les coordinations inter-individuelles chez *bellicositermes natalensis* et *cubitermes* sp. la théorie de la stigmergie : essai d'interprétation du comportement des termites constructeurs.  
*Insectes Sociaux*, 6 :41–80, 1959.



M. Gravel, W.L. Price, and C. Gagné.

Scheduling Continuous Casting of Aluminum Using a Multiple-Objective Ant Colony Optimization Metaheuristic.  
Technical Report 2001-004, Faculté des Sciences de L'Administration, Université Laval, Québec, Canada, April 2001.



Marc Gravel, Wilson L. Price, and Caroline Gagné.

Scheduling Continuous Casting of Aluminum Using a Multiple-Objective Ant Colony Optimization Metaheuristic.

*European Journal of Operational Research*, 143(1) :218–229, November 2002.



Michael Guntsch, Martin Middendorf, and Schmeck H.

An Ant Colony Optimization Approach To Dynamic TSP.

In L. Spector and E.D. Goodman, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, San Francisco, California, July 8 - 11 2001. Morgan Kaufmann, San Francisco, CA.



L. Guoying, L. Zemin, and Z. Zheng.

Multicast routing based on ant algorithm for delay-bounded and load-balancing traffic.

In *Conference on Local Computer Networks*, pages 362–368, 2000.



Walter J. Gutjahr.

ACO algorithms with guaranteed convergence to the optimal solution.

*Information Processing Letters*, 82(3) :145–153, 16 May 2002.



W.J. Gutjahr.

A Graph-based Ant System and its convergence.

*Future Generation Computer Systems*, 16(8) :873–888, 2000.



M. Heusse, D. Snyers, S. Guérin, and P. Kuntz.

Adaptive Agent-Driven Routing and Load Balancing in Communication Networks.

Technical Report RR-98001-IASC, ENST de Bretagne, 1998.



G. Leguizamón and Z. Michalewicz.

A New Version of Ant System for Subset Problems.

In P.J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzalá, editors, *Proceedings of Congress on Evolutionary Computation (CEC99)*, Washington DC, July 6-9 1999. IEEE Press.



Vittorio Maniezzo.

Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem.

Technical Report CSR 98-1, C. L. in Scienze dell'Informazione, Università di Bologna, Sede di Cesena, Italy, 1998.



Vittorio Maniezzo and Antonella Carbonaro.

An ANTS heuristic for the Frequency Assignment Problem.

Technical Report CSR 98-4, C. L. in Scienze dell'Informazione, Università di Bologna, Sede di Cesena, Italy, 1998.



Vittorio Maniezzo and Antonella Carbonaro.

An ANTS heuristic for the frequency assignment problem.

*Future Generation Computer Systems*, 16(8) :927–935, 2000.



Vittorio Maniezzo and Alberto Coloni.

The Ant System Applied to the Quadratic Assignment Problem.

*IEEE Transactions on Data and Knowledge Engineering*, 11(5) :769–778, 1999.



Vittorio Maniezzo, Alberto Coloni, and Marco Dorigo.

The Ant System Applied to the Quadratic Assignment Problem.

Technical Report 94-28, IRIDIA, Université Libre de Bruxelles, Belgium, 1994.



Carlos E. Mariano and Eduardo F. Morales.

MOAQ an Ant-Q Algorithm for Multiple Objective Optimization Problems.

In W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. Jakiela, and R.E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 894–901, Orlando, Florida USA, July 13-17 1999. Morgan Kaufmann, San Francisco, CA.



R. Michel and M. Middendorf.

An Island Model Based Ant System with Lookahead for the Shortest Supersequence Problem.

In A.E. Eiben, T. Bäck, H.-P. Schwefel, and M. Schoenauer, editors, *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature (PPSN V)*, pages 692–701. Springer-Verlag, New York, 1998.



R. Michel and M. Middendorf.

An ACO Algorithm for the Shortest Common Supersequence Problem.

In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimisation*, pages 51–61. McGraw-Hill, London, UK, 1999.

 Nicolas Monmarché, Eric Ramat, Laëtitia Desbarats, and Gilles Venturini.

Probabilistic Search with Genetic Algorithms and Ant Colonies.

In A.S. Wu, editor, *Proceedings of the Optimization by Building and Using Probabilistic Models workshop, Genetic and Evolutionary Computation Conference*, pages 209–211, Las Vegas, Nevada, July 8-12 2000.

 Nicolas Monmarché, Gilles Venturini, and Mohamed Slimane.

On how *Pachycondyla apicalis* ants suggest a new search algorithm.

*Future Generation Computer Systems*, 16(8) :937–946, 2000.

 G. Navarro Varela and M.C. Sinclair.

Ant Colony Optimisation for Virtual-Wavelength-Path Routing and Wavelength Allocation.

In P.J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzalá, editors, *Proceedings of Congress on Evolutionary Computation (CEC99)*, Washington DC, July 6-9 1999. IEEE Press.

 P. Nicolas, F. Saubion, and I. Stéphan.

Genes and Ants for Default Logic.

In *Proceedings of AAAI Spring 2001 Symposium : Answer Set Programming : Towards Efficient and Scalable Knowledge Representation and Reasoning*, Stanford, March 26-28 2001.

 P. Nicolas, F. Saubion, and I. Stéphan.

Answer Set Programming by Ant Colony Optimization.

In *proceedings of the 8th European Conference on Logics in Artificial Intelligence (JELIA'02)*, Cosenza, Italy, September 23-26 2002.

 S. Pimon and C. Solnon.

A Generic Ant Algorithm for Solving Constraint Satisfaction Problems.

Technical report, Brussels, Belgium, September 8-9 2000.

 O. Roux, C. Fonlupt, E.-G. Talbi, and D. Robillard.

ANTabu - enhanced version.

Technical Report LIL-99-1, Laboratoire d'Informatique du Littoral, 1999.



L. Schoofs and B. Naudts.

**Ant Colonies are Good at Solving Constraint Satisfaction Problems.**

In *Proceedings of the IEEE Conference on Evolutionary Computation (CEC)*, volume 2, pages 1190–1195, 2000.



L. Schoofs and B. Naudts.

**Solving CSPs with Ant Colonies.**

In M. Dorigo, M. Middendoff, and T. Stützle, editors, *From Ant Colonies to Artificial Ants : Proceedings of the Second International Workshop on Ant Algorithms (ANTS'2000)*, Brussels, Belgium, September 8-9 2000.



R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz.

**Ant-based Load Balancing in Telecommunications Networks.**

*Adaptive Behavior*, 5(2) :169–207, 1997.



Christine Solnon.

**Solving Permutation Constraint Satisfaction Problems with Artificial Ants.**

In *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000)*, pages 118–122. IOS Press, August 2000.



Thomas Stützle.

**An Ant Approach to the Flow Shop Problem.**

In *EUFIT'98*, pages 1560–1564, Aachen, 1998.



Thomas Stützle and Marco Dorigo.

**ACO Algorithms for the Quadratic Assignment Problem.**

In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimisation*, pages 33–50. McGraw-Hill, London, UK, 1999.



Thomas Stützle and Marco Dorigo.

**ACO Algorithms for the Traveling Salesman Problem.**

In K. Miettinen, M. Mäkelä, P. Neittaanmäki, and J. Periaux, editors, *Evolutionary Algorithms in Engineering and Computer Science : Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming and Industrial Applications*. John Wiley & Sons, 1999.



Thomas Stützle and H. Hoos.

$MAX - MIN$  Ant System and Local Search for the Traveling Salesman Problem.

In IEEE, editor, *Proceedings of the fourth International Conference on Evolutionary Computation (ICEC)*, pages 308–313. IEEE Press, 1997.



Thomas Stützle and H. Hoos.

Improvements on the Ant System : Introducing the  $MAX - MIN$  Ant System.

In *Third International Conference on Artificial Neural Networks and Genetic Algorithms*, University of East Anglia, Norwich, UK, 1997. Springer Verlag.



Thomas Stützle and H. Hoos.

The  $MAX - MIN$  Ant System and local Search for Combinatorial Optimization Problems : Towards Adaptive Tools for Global Optimization.

In *Proceedings of the second Metaheuristics International Conference (MIC'97)*, Sophia-Antipolis, France, 1997.



G. Syswerda.

Simulated crossover in genetic algorithms.

In L.D. Whitley, editor, *Second workshop on Foundations of Genetic Algorithms*, pages 239–255, San Mateo, California, 1993. Morgan Kaufmann.



E. Taillard and L. M. Gambardella.

An Ant Approach for Structured Quadratic Assignment Problems.

In *Proceedings of the second Metaheuristics International Conference (MIC'97)*, Sophia-Antipolis, France, July 21-24 1997.



É.D. Taillard.

FANT : Fast Ant System.

Technical Report 46-98, IDSIA, IDSIA, Lugano, 1998.



Vincent T'Kindt, Nicolas Monmarché, Daniel Laügt, and Fabrice Tercinet.

Combining Ant Colony Optimization and Simulated Annealing to solve a 2-machine flowshop bicriteria scheduling problem.

In *Proceedings of the European Chapter on Combinatorial Optimization*, pages 129–130, Capri, Italy, May 18-20 2000.



I.A. Wagner and A.M. Bruckstein.

**Hamiltonian(t)** - An Ant-Inspired Heuristic for Recognizing Hamiltonian Graphs.

In P.J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, editors, *Proceedings of Congress on Evolutionary Computation (CEC99)*, Washington DC, July 6-9 1999. IEEE Press.



I.A. Wagner, M. Lindenbaum, and A.M. Bruckstein.

**ANTS** : Agent, Networks, Trees, and Subgraphs.

*Future Generation Computer Systems*, 16(8) :915–926, 2000.



T. White and B. Pagurek.

**Towards Multi-Swarm Problem Solving in Networks.**

In *Proceedings of the third International Conference on Multi-Agent Systems (ICMAS'98)*, 1998.

