

SYSTÈMES ARTIFICIELS COMPLEXES

INTELLIGENCE COLLECTIVE

Optimisation par colonie de fourmies

2

- Introduction : Natural Computing

- 1. Swarm Intelligence
 - Insectes sociaux
 - Stigmergie
 - Modélisation et simulation du fourragement

- 2. Ant Colony Optimisation
 - ACO et TSP
 - Algorithme AS
 - Variantes

Natural Computing : objectifs

3

- Dans le but de créer des systèmes autonomes, robustes et évolutifs, une nouvelle forme d'ingénierie trouve son inspiration dans les systèmes naturels complexes
- Par exemple, pour concevoir des systèmes sécurisés contre les intrusions, il est possible d'imaginer de nouveaux mécanismes inspirés des défenses immunitaires
- Ces systèmes doivent être pensé comme la « somme » d'entités distribuées, auto-organisées et adaptatives

Natural Computing : objectifs

4

- L'objectif de ce cours est de présenter un exemple de système informatique « inspirés par la Nature » utilisé en ingénierie (optimisation)
- Lien entre le domaine naturel (éthologie, sociologie, ...) et les méthodes informatiques
- La Nature doit être une source d'inspiration et non une contrainte

Bibliographie / Source

5

- Machine Nature: The Coming Age of Bio-Inspired Computing, by Moshe Sipper, McGraw-Hill, New York, 2002
- Les systèmes complexes : mathématiques & biologie, H. P. Zwirn, ed. O. Jacob
- L'optimisation par essais particulières, M. Clerc, ed. Hermes
- Outil de simulation :
<http://ccl.northwestern.edu/netlogo>
- Réseau National de Sciences de la Complexité :
<http://complexsystems.lri.fr/RNSC>
- <http://fr.wikipedia.org/...>

Bibliographie / Source

6

- E. Bonabeau, M. Dorigo and G. Theraulaz. Inspiration for optimization from social insect behaviour. Nature, Vol. 406, juillet 2000, pp. 39-42
- Bonabeau, E. & Theraulaz, G. (2000). Swarm Smarts. Scientific American, 282 (3): pp. 72-79
- E. Bonabeau, M. Dorigo and G. Theraulaz. (1999) Swarm Intelligence : From Natural to Artificial Systems. Oxford University Press

Bibliographie / Source

7

www.cnrs.fr/Cnrspresse/n386/html/n386a09.htm

L'Intelligence collective des fourmis et nouvelles techniques d'optimisation

« Les recherches sur les comportements collectifs des insectes sociaux fournissent aux informaticiens des méthodes puissantes pour la conception d'algorithmes d'optimisation combinatoire, de routage, ... »

Introduction : Intelligence Collective

- L'observation de la nature a amené des chercheurs à emprunter les principes observés chez les insectes sociaux pour en faire des algorithmes efficaces pour la résolution de certains problèmes
- La notion d'intelligence collective prend aussi une importance notable dans le cadre de l'économie de la connaissance. C'est le cas avec le développement des communautés
 - ▣ en organisation
 - ▣ management de projet
 - ▣ développements technologiques Web 2.0 / Web sémantique
 - ▣ outils tels que Wiki
 - ▣ réseaux sociaux

Intelligence collective

9

- Désigne les capacités cognitives d'une communauté résultant des interactions multiples entre ses membres (agents)
 - Un agent ne possède qu'une perception locale (et donc partielle) de l'environnement et n'a pas connaissance de la totalité des éléments qui influencent le groupe
 - Chaque agent a un comportement simple
 - Collectivement, les agents peuvent accomplir une tâche complexe
- Sous certaines conditions particulières, la synergie créée par la collaboration fait émerger des facultés de représentation, de création et d'apprentissage supérieures à celles des individus isolés

Swarm Intelligence : conditions d'émergence

10

- Une information locale et limitée
 - ▣ chaque individu ne possède qu'une connaissance partielle de l'environnement et n'a pas conscience de la totalité des éléments qui influencent le groupe
- Un ensemble de règles simples
 - ▣ chaque individu obéit à un ensemble restreint de règles simples (sans rapport direct avec le comportement global)
- Des interactions multiples
 - ▣ chaque individu est en relation avec (seulement) un ou plusieurs autres individus du groupe
- Une structure émergente utile à la collectivité
 - les individus trouvent un bénéfice à collaborer et leur performance est meilleure que s'ils avaient été seuls

Intelligence collective et animaux

« sociaux »

11

- L'intelligence collective s'observe chez :
 - ▣ insectes sociaux (fourmis, termites, abeilles, ...)
 - ▣ animaux se déplaçant en formation (meute, oiseaux migrateurs, bancs de poissons, ...)

- Points communs qui caractérisent l'intelligence collective :
 - ▣ les individus sont grégaires car ils obtiennent un avantage substantiel à chasser, se déplacer ou vivre en groupe
 - ▣ interagissent de manière locale par le moyen de signaux (grognement, phéromones, attitudes)
 - ▣ l'individu seul répond instinctivement à certains stimuli
 - ▣ la coordination du groupe est implicite et se fait au travers de règles comportementales très simples au niveau individuel

Les phénomènes de meute relèvent-ils de l'intelligence collective ?

12

Les mammifères chassant en meute (loup, hyène, lionnes) sont moins représentatifs d'une réelle intelligence collective

- ▣ chaque individu est doué d'une capacité cognitive importante
- ▣ possède une information globale assez importante
- ▣ type d'intelligence se rapproche plus de l'organisation et de la coordination qu'on retrouve dans les sports d'équipe ou le travail collaboratif
- ▣ nombre d'individus pas suffisamment élevé pour parler d'émergence d'une intelligence collective

Insectes sociaux

13

- Organisés en colonies
- Démontrent une intelligence collective qui permet de retirer un bénéfice d'un instinct grégaire

- Parmi les insectes sociaux :
 - ▣ Fourmis : Ant Colony Optimization (ACO)
 - ▣ Abeilles : Bees Optimization (BO)
 - ▣ Oiseaux-Poissons : Particule Swarm Optimization (PSO)

Insectes sociaux : Eusocialité

14

- Structure sociale de la colonie : Eusocialité
 - ▣ présence d'une caste stérile (ouvrières)
 - ▣ division du travail
 - ▣ absence de séparation franche entre les différentes générations

- Le fait que les ouvrières ne se reproduisent pas s'explique par le fait que le bénéfice n'est pas donné à leur progéniture mais à des individus « apparentés »

L'altruiste a-t-il un intérêt sélectif ?

15

Darwin (1859) observait que les insectes sociaux posaient problème dans sa théorie de l'évolution :

Comment des individus qui ne se reproduisent pas et, par conséquent, ne participant pas à l'évolution de l'espèce, peuvent-ils être sélectionnés ?

Hamilton (1964) répondit à cette question

- ▣ la fitness darwinienne (mesure du succès reproductif relatif à l'individu), concernant le passage de ses gènes aux générations suivantes *via* ses enfants, était trop étroite pour expliquer l'effet de la sélection naturelle sur la fréquence relative des allèles dans la population
 - ▣ propose que la sélection naturelle agit aussi sur des individus qui aidaient leurs apparentés et augmentaient donc leur aptitude reproductrice globale
 - ▣ introduit la valeur sélective globale (inclusive fitness)
- ▣ A lire : « Je t'aide... moi non plus » C. Clavien Ed. Vuibert (2010)

Colonie d'insectes

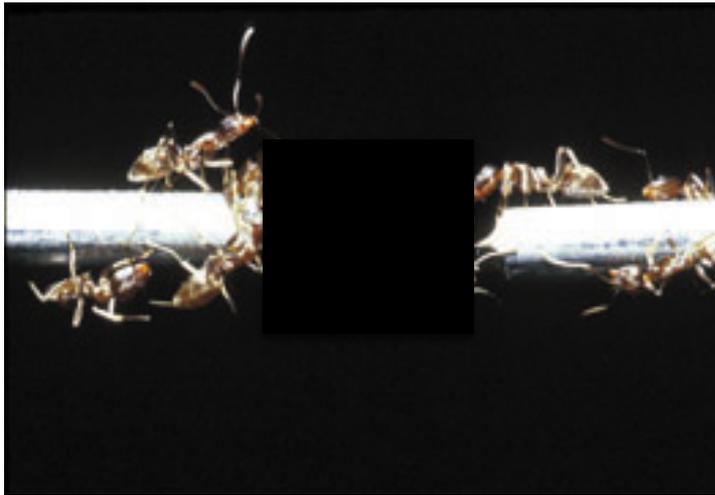
16

- En socio-biologie, une colonie est un groupe d'organismes individuels appartenant à une même espèce vivants rassemblés selon un mode de vie particulier

- La vie en colonie apporte des avantages :
 - ▣ plus fortes défenses,
 - ▣ possibilité de s'attaquer à des proies plus importantes
 - ▣ ...

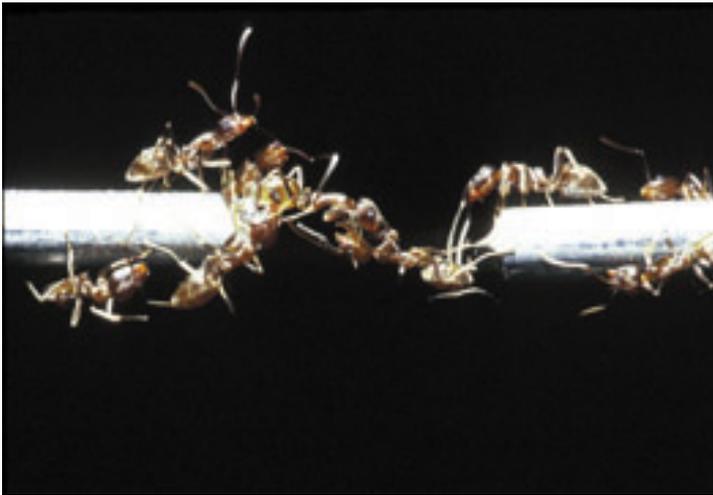
Comportements collectifs des insectes sociaux : **auto-organisation**

17



Comportements collectifs des insectes sociaux : **auto-organisation**

18



Comportements collectifs des insectes sociaux : **auto-organisation**

19



Emergence ?

Intelligence en essaim (swarm intelligence)

20

- Agents petits (puissance mécanique ?), disposant de ressources limitées (intelligence ?), ...

- Sens de la coopération rudimentaire et paradoxalement efficaces au niveau d'ensemble

- Tâches individuelles simples :
 - ▣ ramasser un œuf et l'amener vers un endroit sûr
 - ▣ prendre de la nourriture acheminée par un autre individu et la déposer en un lieu donné
 - ▣ Ramener de la nourriture au nid, ...

Swarm intelligence

21

- Comportement primitif dont le résultat d'ensemble est riche et cohérent
- À la différence des modèles sociaux humains, le comportement de l'individu privilégie exclusivement l'intérêt de la collectivité

Principe : plusieurs entités simples en interaction desquelles émerge à un niveau collectif une structure complexe (qualifiée d'intelligente)

Le cas des fourmis

22

Les fourmis présentent des particularités qui rendent la vie en collectivité très profitable à l'espèce :

- ▣ Un registre comportemental limité
- ▣ Des capacités cognitives limitées, ne permet pas à un seul individu d'obtenir assez d'information sur l'état de la collectivité et de l'environnement pour garantir une division des tâches efficace
- ▣ Des facultés de communication avancées par le biais des phéromones, favorisant des interactions multiples

Le cas des fourmis

23

La colonie est un système complexe stable et auto-régulé capable de :

- ▣ s'adapter aux variations environnementales les plus imprévisibles
- ▣ résoudre des problèmes, sans contrôle externe ou coordination centrale

Cela a permis aux fourmis de s'imposer sur la Terre puisque la biomasse des fourmis est sensiblement identique à celle de l'espèce humaine

Fourragement de nourriture par une colonie de fourmis

24

- Des fourmis se trouvent dans un nid et se déplacent pour récolter de la nourriture auprès de zones attractives
- Le nid et les sources diffusent des odeurs qui décroissent avec la distance
- Les phéromones s'évaporent avec le temps

Fourragement de nourriture

25

- Des entités autonomes au comportement primaire peuvent collecter de la nourriture de manière efficace
- Met en jeu des comportements locaux ; chaque fourmi
 - ▣ ne sait pas où se trouve la fourmilière ni la nourriture
 - ▣ sait uniquement analyser son voisinage direct
- L'environnement lui aussi évolue de manière locale
 - ▣ Les phéromones (molécules présentes dans l'environnement) se répandent localement
 - ▣ constitution d'un gradient d'odeur provenant de la fourmilière ou l'apparition et la disparition de phéromones indiquant la proximité de nourriture

Fourragement chez les fourmis

- Comportement collectif :
formation de chemins sans
aucune carte
- La colonie réalise une
optimisation collective de la
distance sans aucun
« leader »



White-footed ants trailing on a wall
(J. Warner, University of Florida)

Comportement local d'une fourmi : comment se déplacer ?

Chaque fourmi

1. dépose un signal chimique odorant (phéromone) pour indiquer le chemin à ses congénères
 2. sent et suit les phéromones déposés par les autres
- émetteur et récepteur
 - producteur et consommateur



Harvester ant
(Deborah Gordon, Stanford University)

Pourquoi ça marche ?

- Les fourmis déposent de la phéromone quand elle porte de la nourriture et retourne au nid
- Au départ, le choix d'un déplacement est aléatoire
- Le chemin le plus court devient vite le plus marqué car les fourmis qui l'empruntent arrivent plus vite au nid

Phéromone : moyen de communication chez les insectes

29

- Types : sexuelles, de piste, grégaires, d'alarme, ...
- Emetteur : substance chimique produite par des glandes déclenchant des réactions comportementales entre individus de la même espèce
- Recepteur : chimiorécepteurs = antennes garnies d'organes sensoriels
- Applications pratiques en agriculture
 - "pièges à phéromones " composés d'un attractif (synthèse de la phéromone naturelle de la femelle du ravageur) et d'un système assurant la capture des mâles

Piège à phéromones : mouche de l'olive

30

- Piège composé d'un toit englué et d'une capsule de phéromone suspendue au milieu
- Mode d'emploi : dans le cas général, il faut 1 piège par ha

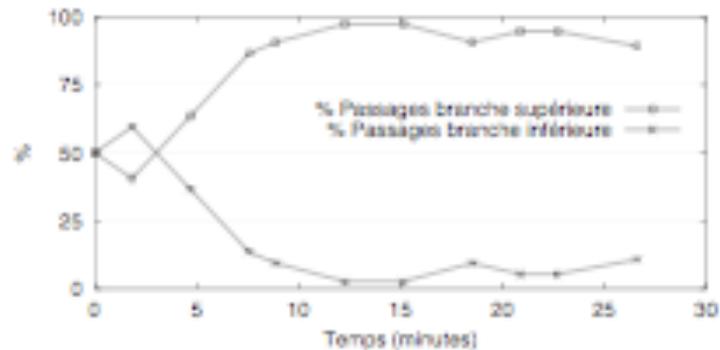


Phénomène autocatalytique

Pont binaire de Deneubourg (1999)

31

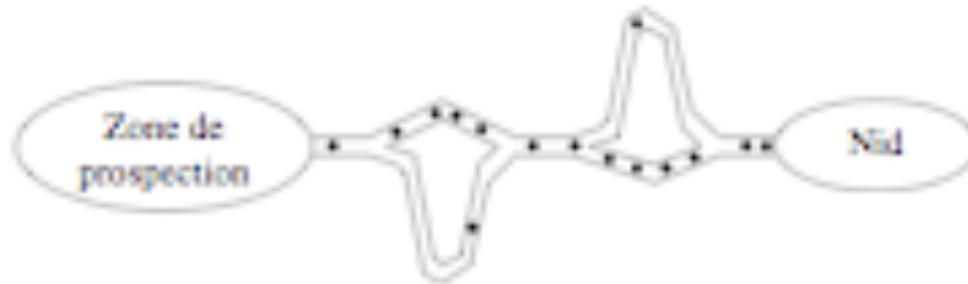
- Initialement le pont est vide
- Après une période transitoire, des fluctuations aléatoires favorisent la piste supérieure ... plus les agents suivent une piste, plus elle devient attractive
- Feedback positif : phénomène qui se renforce lui-même



Expérience du double pont binaire

32

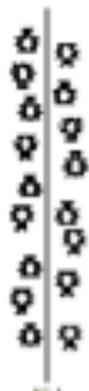
- Influence des fluctuations aléatoires initiales réduite
- Car les 4 chemins possibles sont de longueurs différentes



Effet de la coupure d'une piste

33

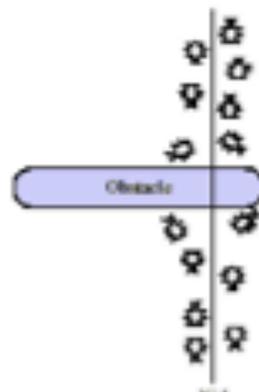
Nourriture



Nid

(a)

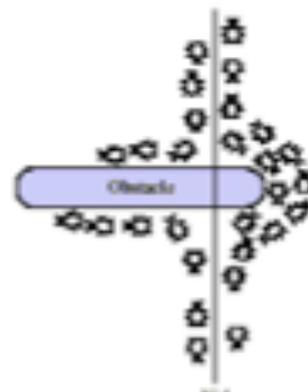
Nourriture



Nid

(b)

Nourriture



Nid

(c)

Fourmis : naturel vs. artificiel

34

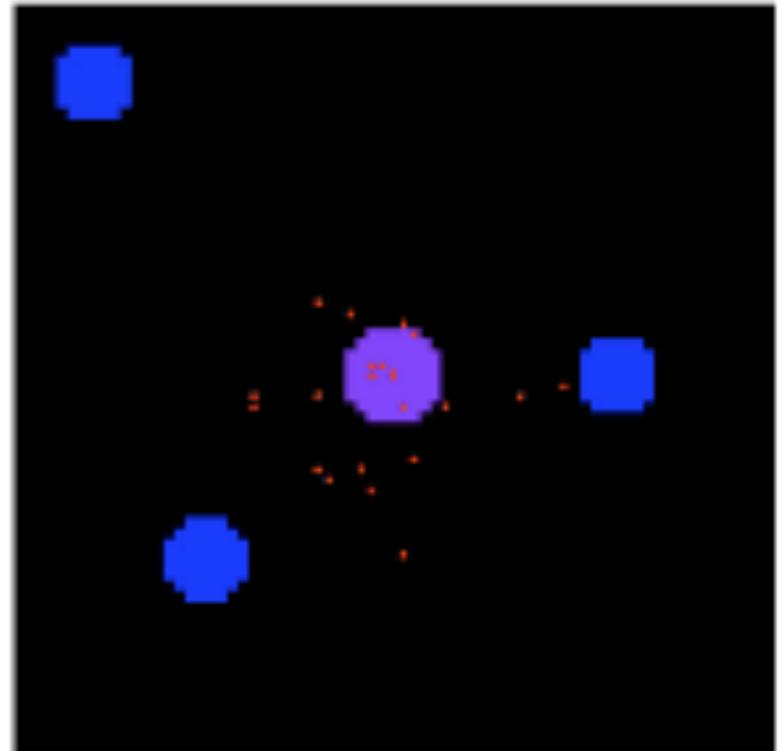
- Les fourmis choisissent la piste qui porte la plus forte concentration
- Piste chimique virtuelle

Important de comprendre les comportements naturels avant d'abstraire dans un algorithme

Mais la Nature doit être une source d'inspiration et non une contrainte et donc on va prendre des « libertés »

Modéliser & Simuler

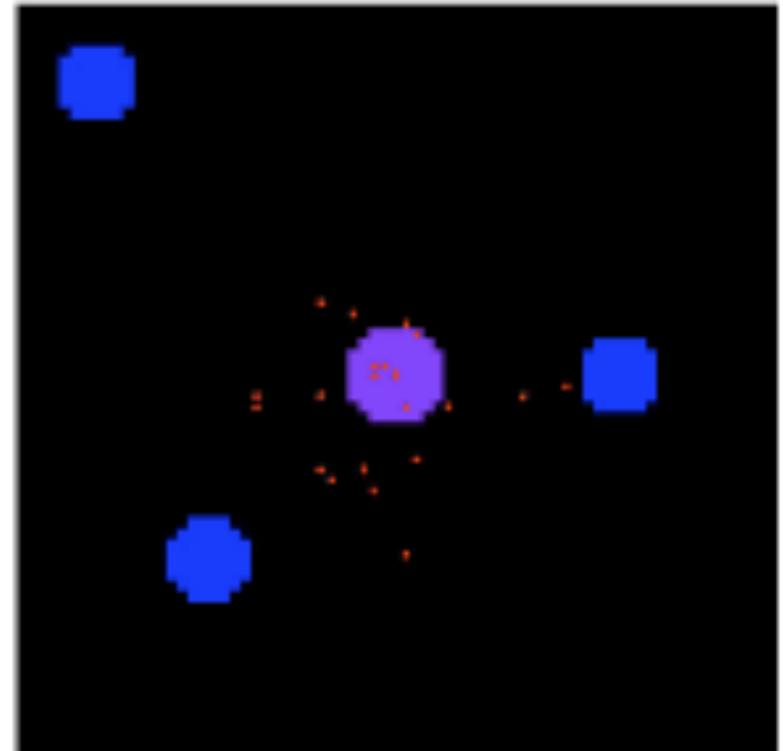
- 1 nid (violet)
- 3 sources de nourriture (bleu)
- 100 à 200 fourmis (rouge)



NetLogo : <http://ccl.northwestern.edu/netlogo>

Initialisation

- 1 nid au centre (violet)
- 3 sources de nourriture (bleu)
- 100 à 200 fourmis (rouge) situées dans le nid



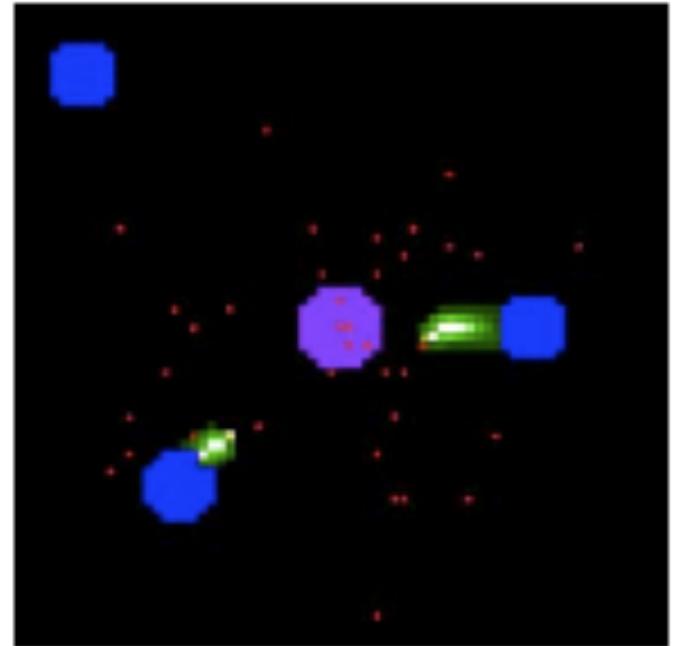
Comportement **local** des agent-fourmis

1. Si trouver nourriture,
 - ▣ la prendre
 - ▣ retourner au nid en suivant l'odeur du nid
 - ▣ déposer de la phéromone

2. Si ne transporte pas de nourriture :
 - ▣ si odeur phéromones, les suivre
 - ▣ sinon se déplacer au hasard

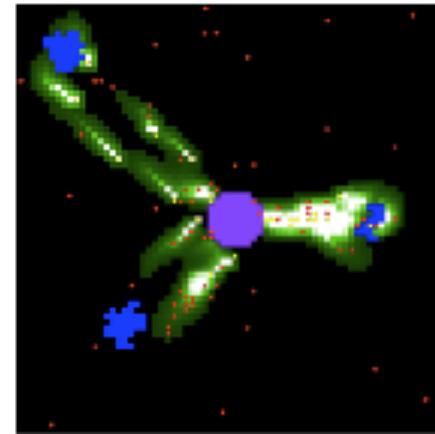
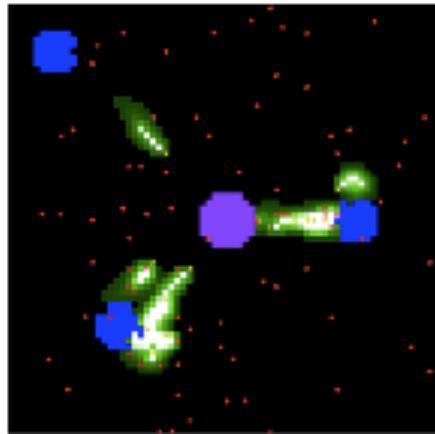
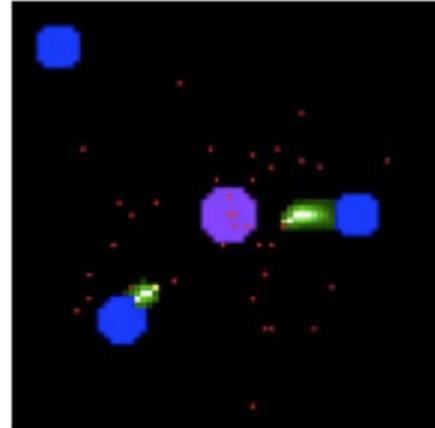
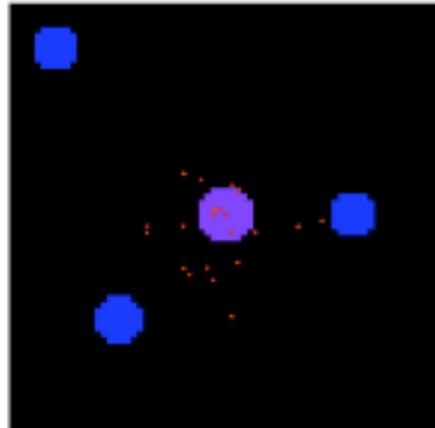
Comportement **global** observé

- Les sources de nourriture sont exploitées selon leur propre distance et attrait
- Emergence d'une décision collective (choix d'un chemin)



Modéliser & Simuler :

Un nid et trois sources de nourritures



population 199

setup

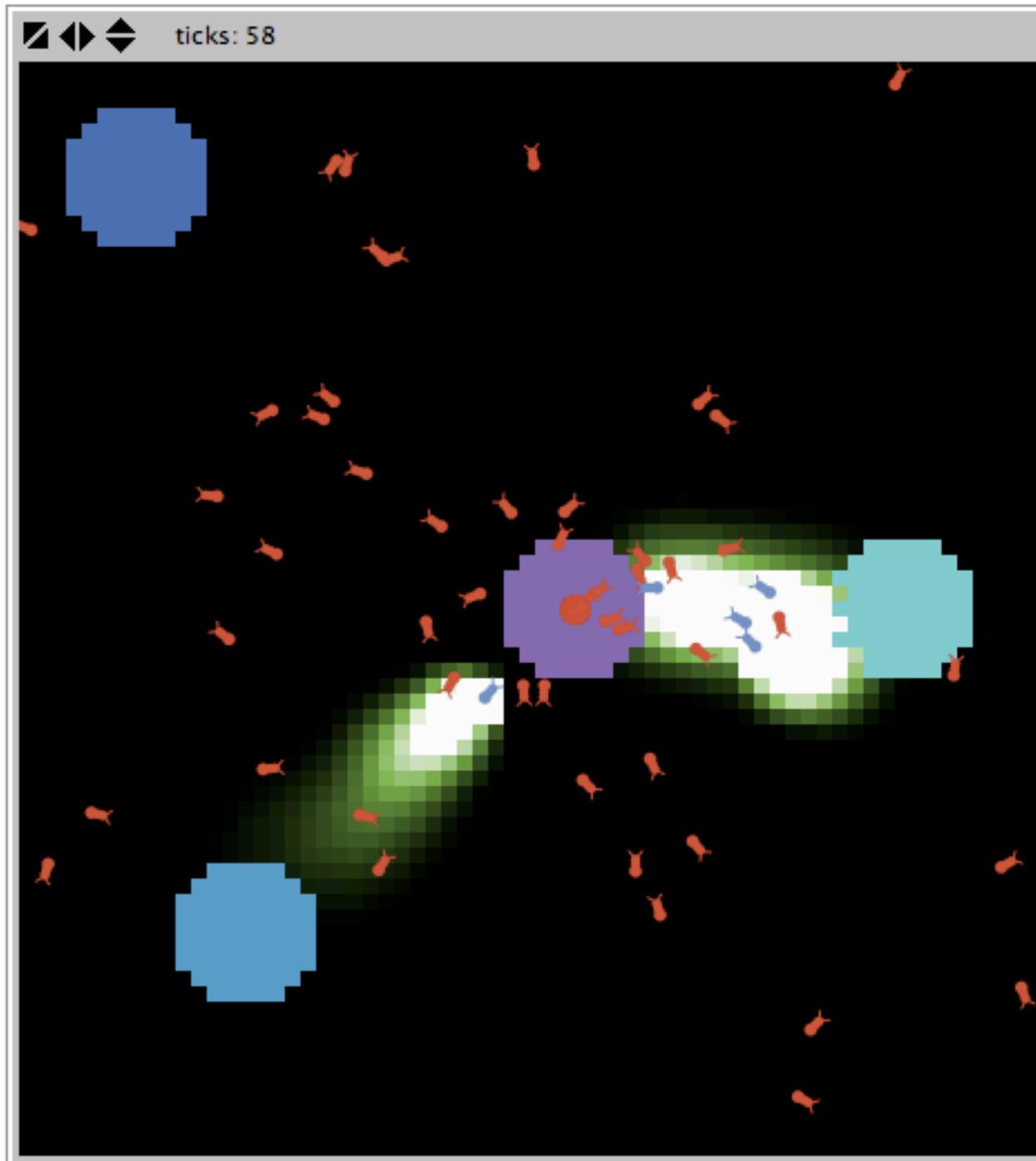
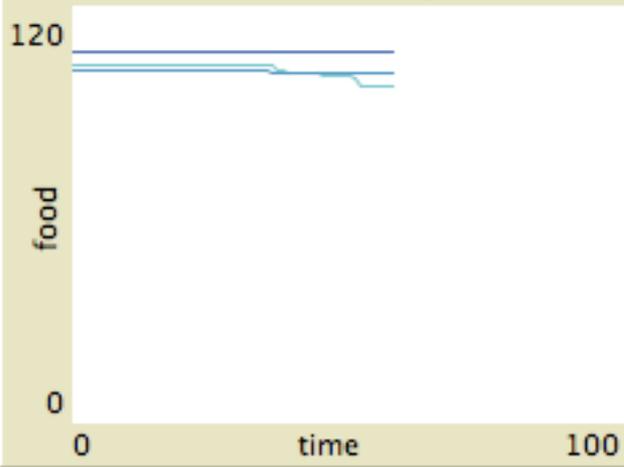
go 2

diffusion-rate 76

evaporation-rate 14

On
 Off plot?

Food in each pile



Simulation NetLogo (variables globales)

```
Globals [  
    ;taille-fourmilier  
    ;taux-diffusion-pheromone  
    ;taux-evaporation-pheromone  
]
```

Simulation NetLogo (variables propres)

```
patches-own [
```

```
  odeur-pheromone ;; quantite de pheromone sur ce patch
```

```
  nourriture      ;; quantite de nourriture sur ce patch
```

```
  nid?            ;; indique si, oui ou non, ce patch est dans le nid
```

```
  odeur-nid      ;; odeur diffusee par le nid sur ce patch
```

```
                ;; un nombre d'autant plus grand que
```

```
                ;; le patch est proche du nid
```

```
]
```

Simulation NetLogo (setup)

```
to setup
```

```
  ask patches [
```

```
    setup-nid
```

```
    setup-nourriture
```

```
  ]
```

```
  create-turtles taille-fourmiliere [
```

```
    setxy 0 0 ; position du nid
```

```
    set shape "bug"
```

```
    set size 2
```

```
    set color red ;; red = ne transporte pas de nourriture
```

```
  ]
```

```
end
```

Simulation NetLogo (go)

```
to go
  ask turtles
  [ if (who >= ticks) [ stop ] ;les fourmis partent du nid l'une apres l'autre
    ifelse (color = red)
      [ chercher-nourriture ] ;; ne transporte pas de nourriture
      [ retourner-au-nid ] ;; transporte de la nourriture et retourne au nid avec
  tortiller-avancer
]
diffuser-pheromone
evaporer-pheromone
tick
end
```

Simulation NetLogo (diffuser-evaporer)

to diffuser-pheromone

```
diffuse odeur-pheromone (taux-diffusion-pheromone / 100)
```

end

to evaporer-pheromone

```
ask patches [
```

```
  let new-odeur
```

```
  set new-odeur (odeur-pheromone * (100 - taux-evaporation-pheromone) / 100)
```

```
  set odeur-pheromone new-odeur
```

```
  if (nourriture = 0 and not nid?)
```

```
    [set pcolor scale-color blue odeur-pheromone 0.1 5]
```

```
]
```

end

Simulation NetLogo (aller-vers-odeur)

```
to aller-vers-odeur [type-odeur] ;; procedure turtle
  let odeur-devant odeur-a-angle type-odeur 0
  let odeur-droite odeur-a-angle type-odeur 45
  let odeur-gauche odeur-a-angle type-odeur -45
  if (odeur-droite > odeur-devant) or (odeur-gauche > odeur-devant)
  [ ifelse (odeur-droite > odeur-gauche)
    [ rt 45 ]
    [ lt 45 ] ]
end
```

Questions / Observations !

47

- Les fourmis exploitent-elles les sources de nourritures en série ou en parallèle ?
- La source la plus proche est-elle exploitée en premier ?
- Y-a-t-il émergence d'un chemin stable (i.e. ininterrompu) entre le nid et une source de nourriture ?
- Existe-t-il une taille critique pour la colonie ?

Questions / Observations !

48

- Déterminer le taux de fourmis « utiles » : ratio entre le nombre de fourmis qui ramènent de la nourriture et le nombre total de fourmis

- Quelle est l'influence sur la dynamique
 - ▣ du taux de diffusion des phéromones
 - ▣ du taux d'évaporation des phéromones

Pour aller plus loin ...

49

- Essayer différents placements pour les sources de nourriture
 - ▣ Que se passe-t-il si deux sources sont à une même distance du nid ?
 - Dilemme exploitation/exploration : série ou //
 - ▣ Que se passe-t-il avec un obstacle entre le nid et une source ?

- Une fourmi utilise une « astuce » pour revenir au nid en suivant une odeur diffusée par le nid. Les vrais fourmis ne font pas ici ! Essayer d'implanter d'autres solutions !

Concepts

- Règles locales simples (parallélisme)
- Emergence de « computation » collective (globale)
- Pas de « leader », pas de carte
 - ▣ actions décentralisées
- Interactions locales
 - ▣ ant \leftrightarrow ant (via l'environnement)
- Transition de phase (?seuil)
 - ▣ masse critique = # minimal de fourmis

Stigmergie

51

Stimulation d'agents par la performance de ce qu'ils ont accompli [Grassé 1999]

□ Communication :

- Décentralisée *via* des interactions locales
- indirecte *via* l'espace

- Coïncidence spatiale et dé-synchronisation temporelle
- Pas de contact physique entre agents

□ Optimisation = Propriété émergente

Ant Colony Optimisation

52

- Introduit par M. Dorigo en 1991
- Historiquement appliqué sur le TSP

Idée : représenter le problème à résoudre sous la forme de la recherche d'un meilleur chemin dans un graphe

Ant Colony Optimisation

53

Méta-problème : comment gérer le compromis exploitation/exploration ?

- ▣ Intensification (via les phéromones) des « bons » chemins
- ▣ Diversification par la nature aléatoire des décisions

A.C.O et T.S.P

54

- Le problème du voyageur de commerce (Traveller Salesman Problem) consiste à trouver le chemin le plus court en passant une seule fois par un nombre donné de villes
- En utilisant des fourmis artificielles conçues pour déposer des pistes de phéromone dont la concentration varie en fonction de la distance totale qu'elles ont parcourue, on peut obtenir des chemins quasi-optimaux

Traveller Salesman Problem

- Etant donné n points (des « villes ») et les distances séparant chaque point, trouver un chemin de longueur totale minimale qui passe une seule fois par chaque point et revienne au point de départ
- Les informaticiens ont utilisé des modèles du comportement collectif des insectes sociaux définis par les éthologistes pour traiter ce type de problèmes d'optimisation

Problème du Voyageur de Commerce (TSP)

56

- Complexité : # de parcours possibles, pour n villes est de $(n-1)!$
 - ▣ 6 villes, 120 parcours possibles
 - ▣ 10 villes plus de 362 000
 - ▣ 60 villes plus de 10^{80} (# atomes dans l'univers)

- Ceci peut expliquer pourquoi ce problème :
 - ▣ n'a pas été étudié sérieusement avant l'arrivée des ordinateurs
 - ▣ est devenu aujourd'hui un benchmark classique

AntSystem (AS)

57

- Le premier ACO (1992)
 - Performances moyennes
 - Étendu à de nouvelles versions : par exemple, MaxMinAS
- Méthode robuste
 - la colonie continue de fonctionner lorsque certains individus échouent à accomplir leur tâche
- Le problème de la représentation !
 - une solution potentielle = un tour du « voyageur »
 - un tour = un chemin suivi par une fourmi artificielle

Algorithme AS

58

- **Initialiser**
- **repeat**
 - Chaque ant construit une solution (i.e. un tour complet)
 - Mettre à jour les phéromones
 - Ajouter (déposer) de la phéromone afin de récompenser (renforcer) les meilleures solutions
 - Retirer (évaporer) les traces de phéromone
- **until (maxCycles ou bonneSolution)**

AntSystem : initialiser...

59

Les n^2 distances inter-villes sont connues

1. Répartir les m ants aléatoirement sur les n villes
2. Pour chaque ant, initialiser aléatoirement sa `liste-tabou` avec sa ville de départ
3. Initialiser les pistes de phéromones : $\tau_{ij} = C$, où C constant positive non nulle

Algorithme AS

60

- **Initialiser**
- **repeat**
 - Chaque ant construit une solution (ie. un tour complet)
 - Mettre à jour les phéromones
 - Ajouter (déposer) de la phéromone afin de récompenser (renforcer) les meilleures solutions
 - Retirer (évaporer) les traces de phéromone
- **until (maxCycles ou bonneSolution)**

AntSystem :

Chaque ant construit un tour complet

61

```
for(k=0 ; k < m ; k++){  
  for(l=0 ; l < n ; l++) {  
    la fourmi k choisit une ville j et s'y déplace }}
```

- La fourmi k placée sur une ville i à l'instant t choisit une ville de destination j en fonction de :
 - ▣ sa liste tabou (pour exclure les villes déjà visitées)
 - ▣ la quantité de phéromone $\tau_{ij}(t)$ sur l'arc reliant les deux villes
 - ▣ la visibilité de la ville j : η_{ij} ($1/\text{distance}$ entre i et j)

- Importance relative des phéromones et de la visibilité ?
 - ▣ choix aléatoire selon deux paramètres α et β

AntSystem : phéromone vs. visibilité

62

- **Visibilité** : information statique qui ne dépend que du problème
- **Piste chimique** : information dynamique gérée par chaque agent
 - ▣ mémoire distribuée qui représente l'état de la recherche de la solution
 - ▣ chaque agent modifie la façon dont le problème va être représenté et donc perçu par les autres agents
 - ▣ chaque agent est à la fois producteur et consommateur
 - ▣ mémoire (indirectement) partagée via l'environnement (stigmergie)

AntSystem : phéromone vs. visibilité

63

Choix aléatoire selon deux paramètres α et β qui contrôlent l'importance relative des **phéromones** et de la **visibilité**

La probabilité d'une transition entre deux villes est un compromis entre visibilité et pistes chimiques

- ▣ $\alpha=0$: les villes les + proches ont + de chance d'être sélectionnées (algorithme glouton)
- ▣ $\beta=0$: seule l'amplification des phéromones agit : convergence prématurée

Algorithme AS

64

- **Initialiser**
- **repeat**
 - Chaque ant construit une solution (ie. un tour complet)
 - **Mettre à jour les phéromones**
 - Ajouter (déposer) de la phéromone afin de récompenser (renforcer) les meilleures solutions
 - Retirer (évaporer) les traces de phéromone
- **until (maxCycles ou bonneSolution)**

Fin d'un cycle de base

Toutes les ants ont terminé un tour en revenant à leur ville de départ

65

- Pour chaque ant k :
 - calculer la longueur de son tour $L_k(t)$
 - vider sa liste-tabou
- Mettre à jour les phéromones τ_{ij}
- Rechercher le plus petit tour et le mémoriser s'il est meilleur que les précédents
- ✓ Chaque ant recommence un nouveau tour à partir de sa propre ville de départ

Comment mettre à jour les phéromones ?

66

Attention : c'est une m.a.j « retardée » à l'instant $t+n$ pour la période $[t .. t+n]$

$$\tau_{ij}(t+n) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t)$$

ρ dans $[0, 1[$

- évaporation de la piste
- Éviter le piège des minima locaux
- Permettre d'oublier les solutions sous-optimales

$\Delta\tau_{ij}(t)$

- quantité de phéromone déposée par les ants entre t et $t+n$

Évaporation des phéromones

67

- Si $\rho=0$, pas d'évaporation donc pas de limitation du phénomène autocatalytique
- Si $\rho=1$, les ants prennent seulement en compte les dépôts du dernier cycle, donc pas de mémoire

ρ représente la persistance de la piste (ie. l'effet mémoire)

Quantité de phéromone déposée par les ants lors d'un tour (cycle)

68

$\Delta\tau_{ij}(t)$ = quantité de phéromone déposée par les ants sur l'arc reliant la ville i à la ville j entre les instants t et $t+n$

Pour chaque ant k passant par l'arc (i, j) ,

$$\Delta\tau_{ij}(t) += Q/L_k(t)$$

Où

- ▣ Q représente un « quota » de phéromones attribué à chaque ant (souvent $Q=100$)
- ▣ $L_k(t)$ est la longueur du tour réalisé par l'agent k

Idée : + un tour est court, + les arcs qui le composent sont approvisionnés en phéromone

Remarque : c'est bien une m.a.j « retardée »

Intensification vs. diversification

69

- Intensification : **exploitation** de l'information disponible à un instant donnée durant un run
- Diversification : **exploration** de régions de l'espace de recherche pas ou insuffisamment prises en compte

- Méta-problème : comment gérer à chaque instant le compromis?
 - ▣ choisir *a priori* un bon réglage des paramètres (α , β , ρ)
 - ▣ contrôler la dynamique en fonction de l'état du système

Hybridation : ACO/LocalSearch

70

- Objectif : améliorer la performance globale de l'algorithme
- Comment : optimiser chaque « tour » par des modifications locales

- Initialiser
- **repeat**
 - Chaque ant construit un tour complet
 - **Optimiser chaque tour par une recherche locale**
 - Mettre à jour les phéromones (déposer+évaporer)
- **until** (maxCycles ou bonneSolution)

Optimisation

par colonie de fourmis et TSP

71

	Oliver30 (30-city)	Eil50 (50-city)	Eil75 (75-city)
ACO	420 (830)	425 (1830)	535 (3480)
Genetic algorithm	421 (3200)	428 (25000)	545 (80000)
Evolutionary programming	420 (40000)	426 (100000)	542 (325000)
Simulated annealing	424 (24617)	443 (68512)	580 (173250)
Optimal solution	420	425	535

Cette optimisation est une conséquence de l'interaction subtile entre renforcement et évaporation de la phéromone, qui fait que seules les meilleures liaisons subsistent

Variante : algorithme Max-MinAS

72

- Fournir des résultats compétitifs
- Imposer des bornes τ_{\min} et τ_{\max} aux traces de phéromones
- Les traces sont initialisés avec τ_{\max}

Quel est l'effet de ces choix sur le fonctionnement de l'algorithme ?

A voir en Travaux dirigés ...

Algorithme Max-MinAS

73

- Initialiser
- **repeat**
 - Chaque ant construit une solution
 - Améliorer chaque solution par recherche locale
 - Mettre à jour les phéromones (déposer+évaporer)
 - **Si une tracePhéromone** $<\tau_{\min}$ **alors la mettre à** τ_{\min}
 - **Si une tracePhéromone** $>\tau_{\max}$ **alors la mettre à** τ_{\max}
- **Until** (maxCycles ou bonneSolution)

Software package : ACOTSP

74

- Auteur : Thomas Stütze
- ACOs appliqués aux TSP symétriques
 - Ant System (AS)
 - Elitist Ant System (EAS)
 - MAX-MIN Ant System (MMAS)
 - Rank-based version of Ant System (RAS)
 - Best-Worst Ant System (BWAS)
 - Ant Colony System (ACS)
- ANSI C sous Linux

`http://iridia.ulb.ac.be/~mdorigo/ACO/aco-code/public-software.html`

ACOTSP : command line options (1)

75

```
-r          # number of independent trials
-s          # number of steps in each trial
-t          # maximum time for each trial
-i          f inputfile (TSPLIB format necessary)
-o          # stop if tour >= optimum is found
-m          # number of ants
-g,         # nearest neighbours in tour construction
-a          # alpha (influence of pheromone trails)
-b          # beta (influence of heuristic information)
-e          # rho: pheromone trail evaporation
-q          # q_0: prob. of best choice in tour construction
-c          # number of elitist ants
-f          # number of ranks in rank-based Ant System
-k          # No. of nearest neighbors for local search
-l          0: no local search   1: 2-opt   2: 2.5-opt   3: 3-opt
-d          1 use don't look bits in local search
```

ACOTSP : command line options (2)

76

```
-u, --as      apply basic Ant System  
-v, --eas    apply elitist Ant System  
-w, --ras    apply rank-based version of Ant System  
-x, --mmas   apply MAX-MIN ant system  
-y, --bwas   apply best-worst ant system  
-z, --acs    apply ant colony system  
  
-h, --help   display the help text and exit
```

Options `--as`, `--eas`, `--ras`, `--mmas`, `--bwas`, `--acs`, `--help` don't need arguments, while all the others do

Librairie TSPLIB

77

- <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>
- Cette librairie contient des exemples simples de TSP ; on se limitera à des TSP symétriques (Symmetric traveling salesman problem)
- La distance entre le noeud i et le noeud j est la même qu'entre les noeuds j et i
- Etant donné un ensemble de noeuds et de distances entre chaque paires de noeuds, il s'agit de trouver un chemin de longueur minimale visitant exactement une et une seule choix chaque noeud

Benchmark : ulysses22.tsp

78

COMMENT: Odyssey of Ulysses

DIMENSION: 22

EDGE_WEIGHT_TYPE: GEO

DISPLAY_DATA_TYPE:

COORD_DISPLAY

NODE_COORD_SECTION

1 38.24 20.42

2 39.57 26.15

3 40.56 25.32

4 36.26 23.12

5 33.48 10.54

6 37.56 12.19

7 38.42 13.11

8 37.52 20.44

9 41.23 9.10

10 41.17 13.05

11 36.08 -5.21

12 38.47 15.13

13 38.15 15.35

14 37.51 15.17

15 35.49 14.32

16 39.36 19.56

17 38.09 24.36

18 36.09 23.00

19 40.44 13.57

20 40.33 14.15

21 40.37 14.23

22 37.57 22.56

EOF

ulysses22.opt.tour : tour optimal

79

NAME : ulysses22.opt.tour

COMMENT : Optimal solution of ulysses22

DIMENSION : 22

TOUR_SECTION

1

14

13

12

7

6

15

5

11

9

10

19

20

21

16

3

2

17

22

4

18

8

-1