Alexis Couvreur
Alaric Tabariès

Université Nice Sophia Antipolis

Processions vers la complexité

M1 Informatique : Résolution de problèmes

8 Novembre 2017

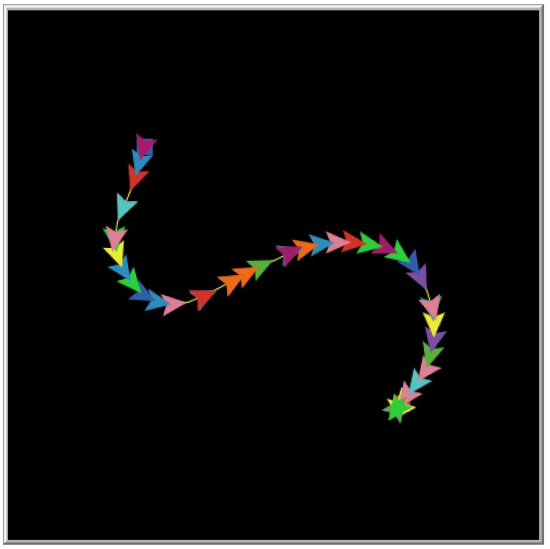
Exercice 1 : Modèle des chenilles processionnaires du pin

Après avoir réalisé les différentes étapes de l'exercice 1, on peut voir que le premier agent se dirige correctement vers notre curseur de souris, lorsque celui-ci est appuyé, et que chaque agent "suiveur" suit bien l'agent se trouvant juste devant lui. Enfin, quand l'agent de tête se rapproche de l'emplacement d'arrivée, il est attiré par ce point et par conséquent, les autres suivent.

La simulation se met dans un état d'attente lorsque le clic de la souris est relâché.

Extrait de code de l'exercice 1

```
to go-tete
   ask la-tete [
      ifelse (proche-arrive?) [ ; alors aller
directement à l'arrivée
           move-to patch-arrivee
         ] ; fin alors
         [ ; sinon ...
           ; attendre que le bouton de la souris
soit enfoncé
           while [not mouse-down?] [
             ; le bouton de la souris a été
enfoncé
             ; faire face a la souris et avancer
d'un pas
           ]
           face patch (mouse-xcor) (mouse-ycor)
           fd 1
         ] ; fin sinon
   ] ; fin ask
end

to go-suiveurs
   ;; créer un nouveau suiveur
   create-turtles 1 [ ; créer une seule turtle
      set size 2
      set shape "default"
      move-to patch-depart
      set devant turtle (who - 1)

   ]
   let les-suiveurs turtles with [who > 0]
   ask les-suiveurs [
     face devant
   ]
   ask les-suiveurs [
     fd 1
   ]
end
```



Résultat de l'exercice 1

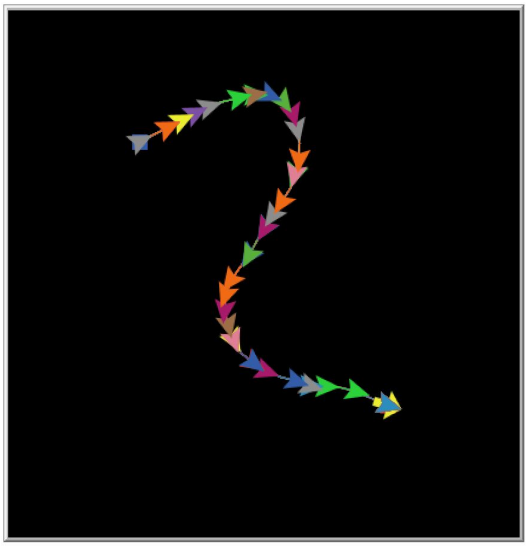
Exercice 2 : Modèle de la Descente aux flambeaux : exemple d'optimisation globale

Après avoir réalisé les différentes étapes de l'exercice 2, on peut voir qu'avec un délai égal à 1 tick, les agents se suivent sans voir leur trajectoire converger, alors qu'avec un délai supérieur à 1, leur trajectoire converge petit à petit vers un chemin rectiligne entre le point de départ et l'arrivée (cf: graphique).

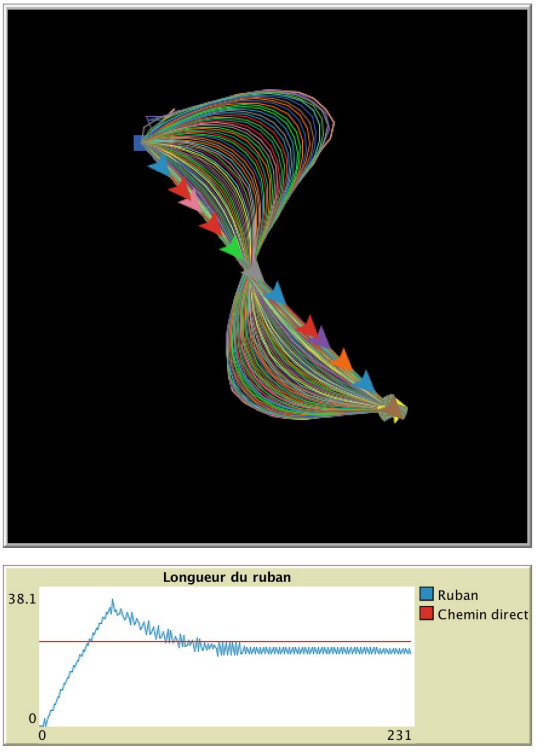
A noter que le léger décalage entre la longueur du ruban formé par les agents et la longueur du chemin direct entre le point de départ et le point d'arrivée s'explique par le petit décalage au départ dû au délai.

Extrait de code de l'exercice 2

```
to-report proche-arrive? ;; reporter exécuté par
une turtle
   report ((distance patch-arrivee) < 3)
end

to-report longueur-ruban
  let vdistance 0
  let les-suiveurs-en-route turtles with [not
proche-arrive? and who > 0]
  ask les-suiveurs-en-route [
    set vdistance (vdistance + distance devant)
  ]
  if (count les-suiveurs-en-route = 0)
  [report 0]
  report vdistance
end

to-report distance-depart-arrivee
   report [ distance patch-depart ] of
patch-arrivee
end
```



Résultat de l'exercice 2 avec un délai à 1 tick



Résultat de l'exercice 2 avec un délai à 2 tick

Ce résultat est démontré par A. M. Bruckstein dans son article "*WHY THE ANT TRAILS LOOK SO STRAIGHT AND NICE*" (SIAM 40'th Anniversary Meeting, Los Angeles, California, July 20-24, 1992) disponible à l'adresse suivante http://www.cs.technion.ac.il/~freddy/papers/Papers46.PDF.

Une traduction de son dernier paragraphe, "*Discussion*" :

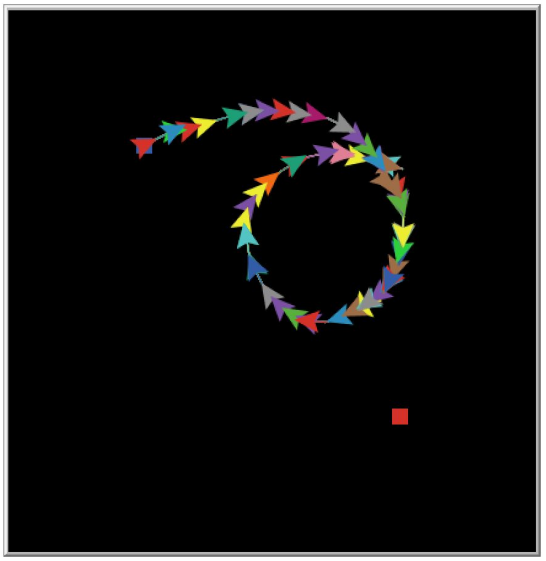
> L'analyse du problème de la procession des fourmis indique qu'une simple interaction d'entités peut résoudre le problème qui est de trouver un chemin optimal entre un point de départ et un point d'arrivée. Ce résultat se généralise aussi pour les espaces dimensionnels plus grands. Il est aussi possible de trouver un autre modèle de suivi de chemin, basé sur une simple interaction locale, qui fournit une convergence vers un chemin rectiligne. Il est d'un grand intérêt d'obtenir de genre de résultat qui montrent que, dans la majorité des cas, des solutions optimales aux problèmes de navigation peuvent être obtenues avec le résultat d'une coopération à courte distance entre deux simples entités.
>
> On peut aussi voir qu'une simple loi de procession circulaire est suffisante pour s'assurer que toutes les entités vont finalement se retrouver. Toutefois, leur chemin de procession peut s'avérer assez complexe.
>
> L'analyse du comportement général qui résulte de règles d'interactions simples et locales est un sujet fascinant qui peut même amener à une meilleure compréhension de la nature et du comportement des colonies artificielles d'animaux (via ex.: Braitenberg[12]). De telles idées peuvent aussi être utilisées dans les problèmes découlant du domaine de la robotique, par exemple.
>
> Pour conclure, il me semble approprié de citer un vieux dicton «Va vers la fourmis, paresseux; Considère ses voies, et deviens sage. »

Exercice 3 : Modèle en anneau d'agents processionnaires : cycle limite

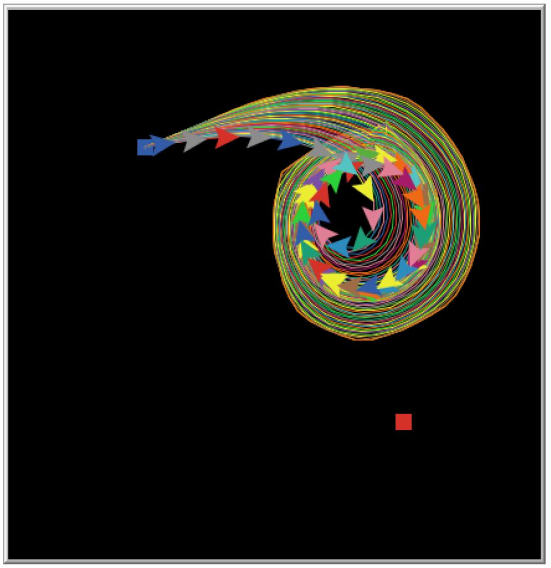
Après avoir réalisé les différentes étapes de l'exercice 3, on peut constater que, suite aux modifications, et avec un délai égal à 1 tick, lorsque l'on simule une boucle, l'entité de tête ainsi que les suiveurs bouclent et ne suivent plus le curseur. Lorsque le délai est supérieur à 1, le cercle formé par les entités converge : il voit son rayon se réduire pour tendre vers zéro.

Extrait de code de l'exercice 3

```
to go-tete
   ask la-tete [
    if(devant = nobody)[
     let t turtles-here with [who > 5];;
     if(count t > 0)
     [set devant one-of t]

      ifelse (proche-arrive?) [ ; alors aller
directement à l'arrivée

        move-to patch-arrivee
      ] ; fin alors
      [ ; sinon ...
        ; attendre que le bouton de la souris
soit enfoncé


          while [not mouse-down?] [
            ; le bouton de la souris a été
enfoncé

          ]
          ; faire face a la souris et avancer
d'un pas
          face patch (mouse-xcor) (mouse-ycor)
          fd 1
        ] ; fin 2e alors
      ] ; fin sinon
   ] ; fin ask
end

to go-suiveurs
   ;; créer un nouveau suiveur
   ...
   let les-suiveurs turtles with [devant !=
nobody]
   ...

end

;; Remarque : la définition d'un suiveur n'est
plus par rapport à who > 0 mais au fait qu'il
suive quelqu'un ou non (la tête devient un
suiveur)
```



Résultat de l'exercice 3 avec un délai à 1 tick



Résultat de l'exercice 3 avec un délai à 2 tick

Exercice 4 : Modèle en anneau : émergence de formes