



**HAL**  
open science

## Exploration de modèles multi-agents

Etienne Delay, Sébastien Rey-Coyrehourcq

► **To cite this version:**

Etienne Delay, Sébastien Rey-Coyrehourcq. Exploration de modèles multi-agents. École thématique. Oléron, France. 2017. cel-02285696

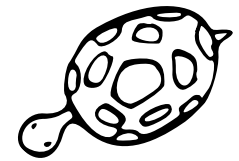
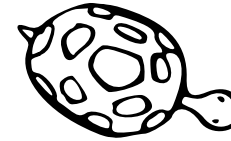
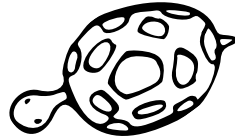
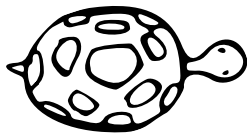
**HAL Id: cel-02285696**

**<https://shs.hal.science/cel-02285696>**

Submitted on 12 Sep 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



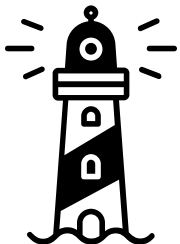
# Exploration de modèles multi-agents

Etienne Delay

<http://geolab.univ-bpclermont.fr/spip.php?article184>

Sébastien Rey Coyrehourcq

<http://orcid.org/0000-0001-7296-9695>



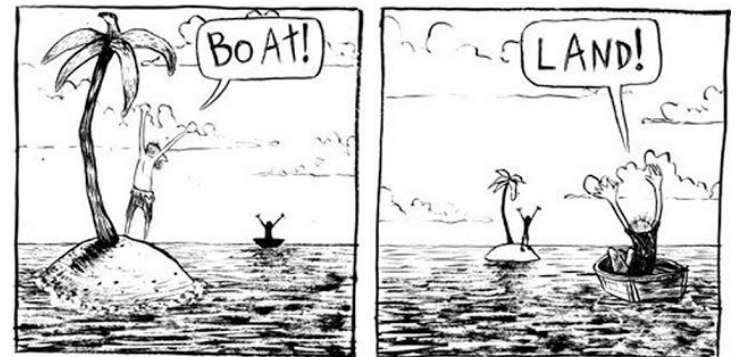
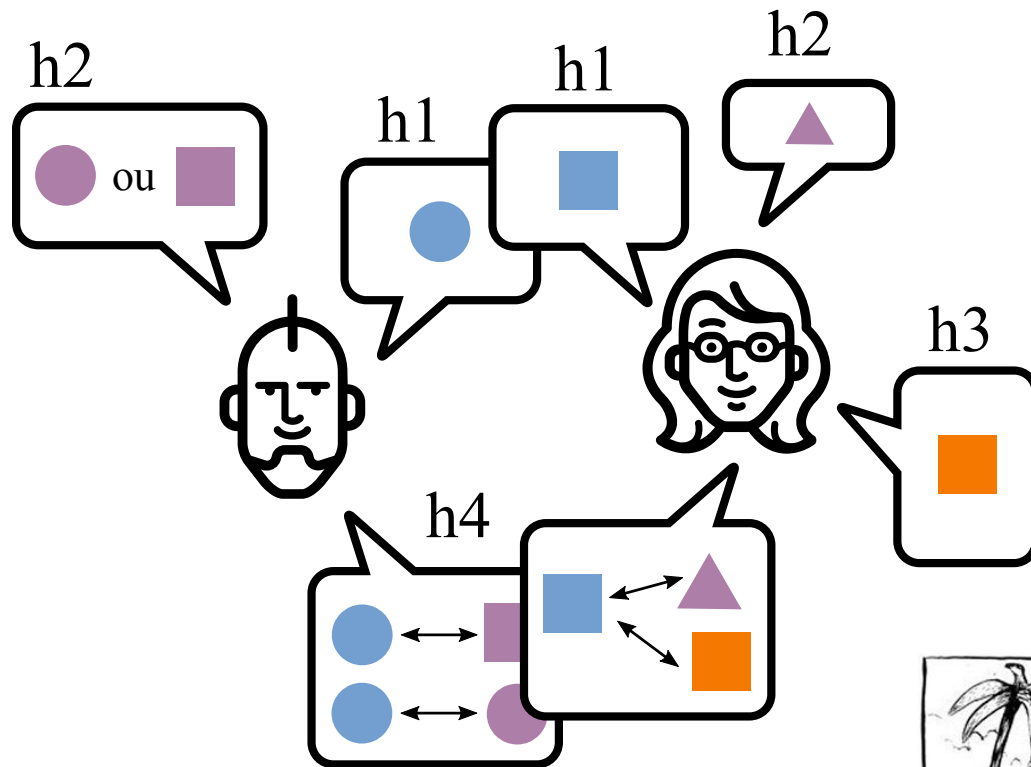
MAPS 10 - Oléron  
06/2017

Résultats et script disponible sur [github](#) 

<https://slides.com/sebastienreycoyrehourcq/>

# Explorer pour discuter

Des conceptions complémentaires de la réalité





# Explorer pour faire des choix

connaître et agir sur la dépendance au chemin

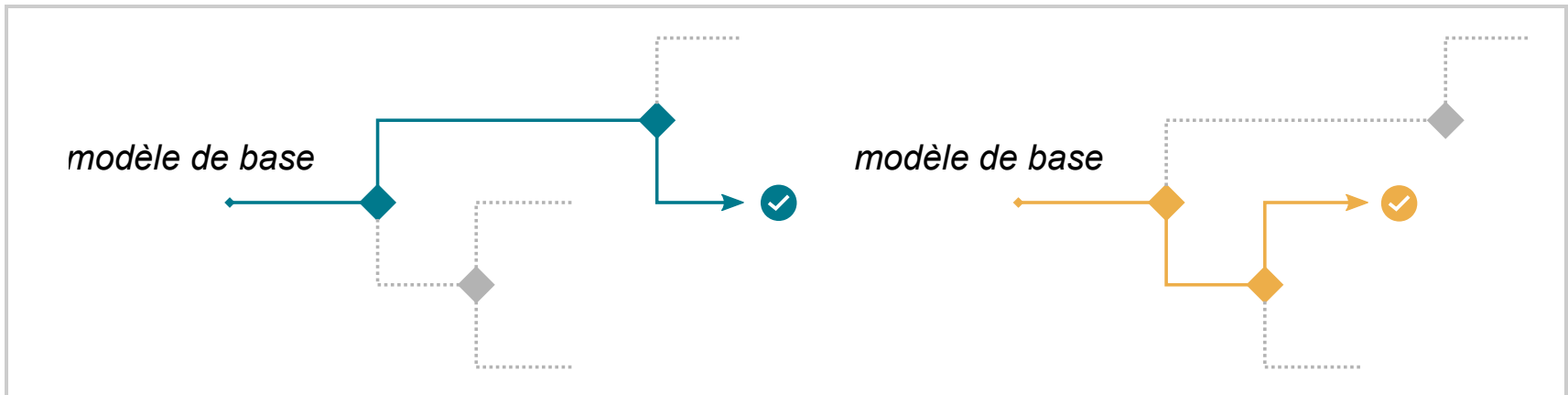


(1, 2)

[Grimm2016, Cottineau2016]



Les **choix** génèrent des **chemins** expliquant le même **pattern**



Deconstruire & reconstruire pour mieux comprendre ...

1 **Look back** and challenge choices (◆)

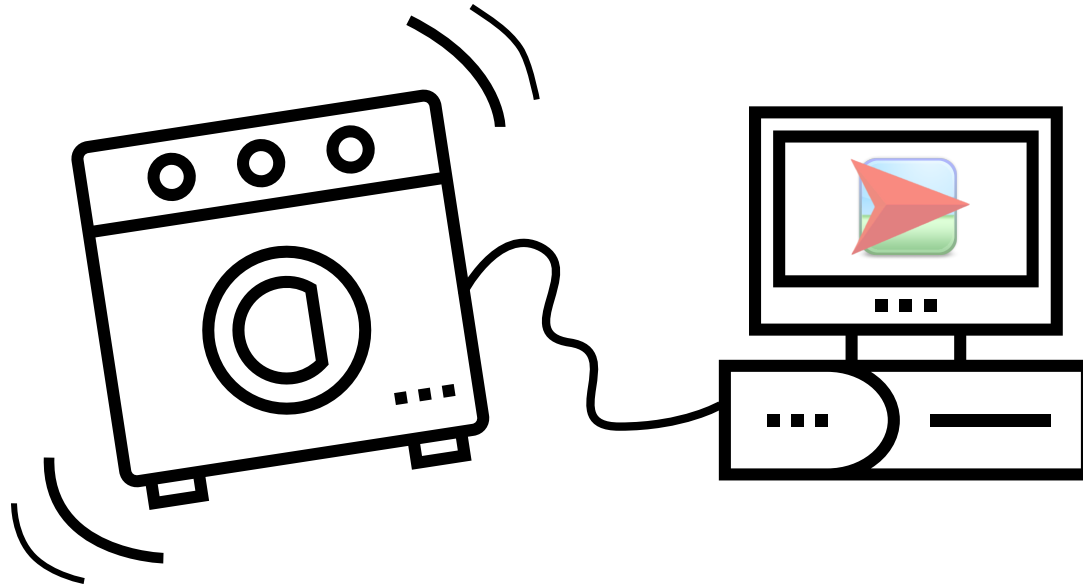
2 Know **where** and **when** **parameters** and **mechanisms** break

3 **Share** your results and **challenge** others

*Cumulative knowledge*

# Explorer pour comprendre

le "laboratoire virtuel", un agitateur d'idées  
mais pas que ...



comportements attendus / simulés ?  
défaillances ?  
robustesse(s) ?



# Bénéfices de l'exploration

L'exploration des modèles revient donc à **organiser** :

- a) la saine **confrontation** des points de vue
- b) la mesure des **écarts** entre **comportements internes attendus et simulés**
- c) la chasse aux **bugs** et **cas limites**

Autrement dit, un processus **systematique** qui stimule **l'effet de surprise**.



# Analyse de sensibilité ?



"Sensitivity analysis (SA) is used to **explore the influence of varying inputs on the outputs of a simulation model** (Ginot et al. 2006).

"By exploring the sensitivity of model output to changes in parameters, we learn about **the relative importance of the various mechanisms** represented in the model and how robust the model output is to parameter uncertainty."

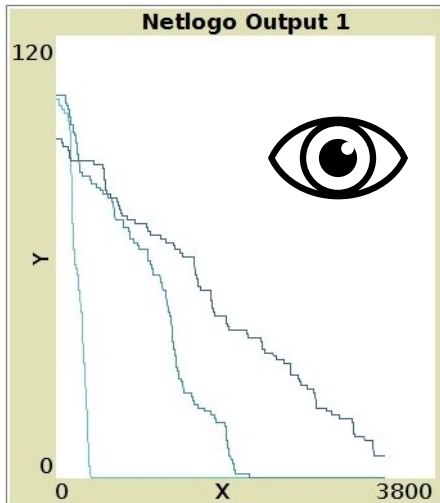
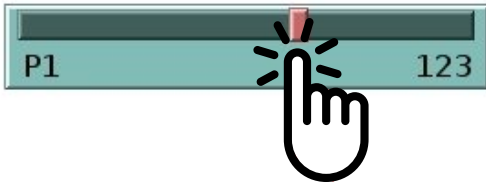
[Thiele2014]

- sensibilité **locale**
- sensibilité **globale**



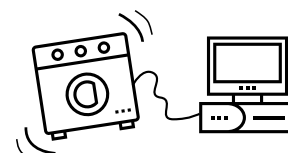
# Comment ?

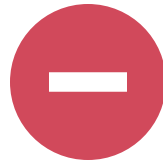
stratégie / DOE  
"One-Variable-At-A-Time"



évaluation à l'oeil  
"Face Validity"

## Sensibilité locale - OAT

- peu fiable 
- couteux en temps
- limité
- difficilement reproductible

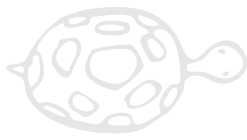


- ludique
- interactif
- exploratoire
- incrémental
- essai erreur

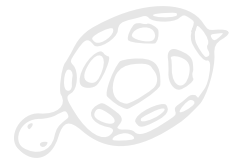


"**Tinkering**" de Papert & Retsnick



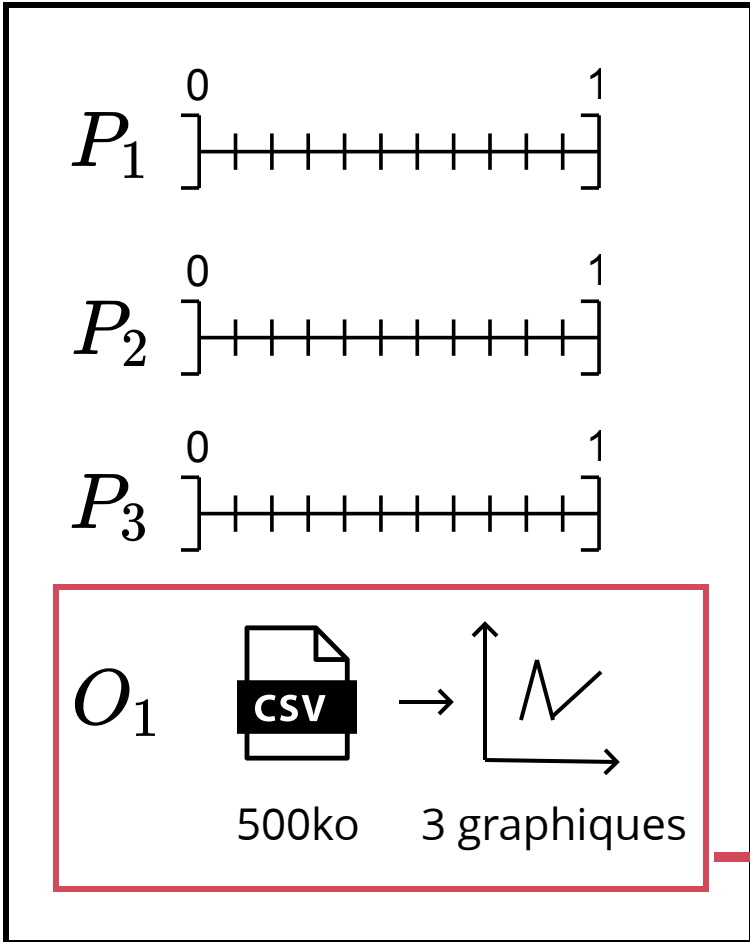


# Comment ?



Une première étape pour systématiser OAT ?

=> Sensibilité Globale



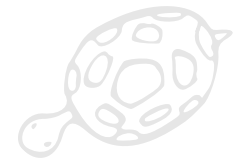
un modèle simple,  
un **plan d'expérience complet** :

**P1 x P2 x P3**

sorties systématiques



# Comment ?



## 1 er problème - Explosion combinatoire

$$discreteStep^{(\sum_{i=1}^n P_i)} * replication * duree$$

$$\{replication \in \mathbb{R} \mid 30\}$$

$$\{duree \in \mathbb{R} \mid 1minute\}$$

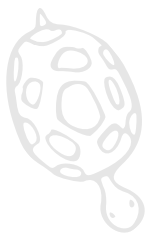
$$discreteStep = 11 \quad P = 3$$

$$11^3 * 30 * 1min = 39930min = 27journs$$

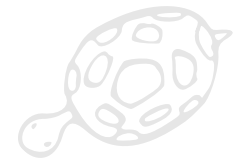
$$11^3 * 30 * 500ko = 19965000Ko = 19Go$$

$$11^3 * 30 * 3graphiques = \underline{119790graphiques}$$

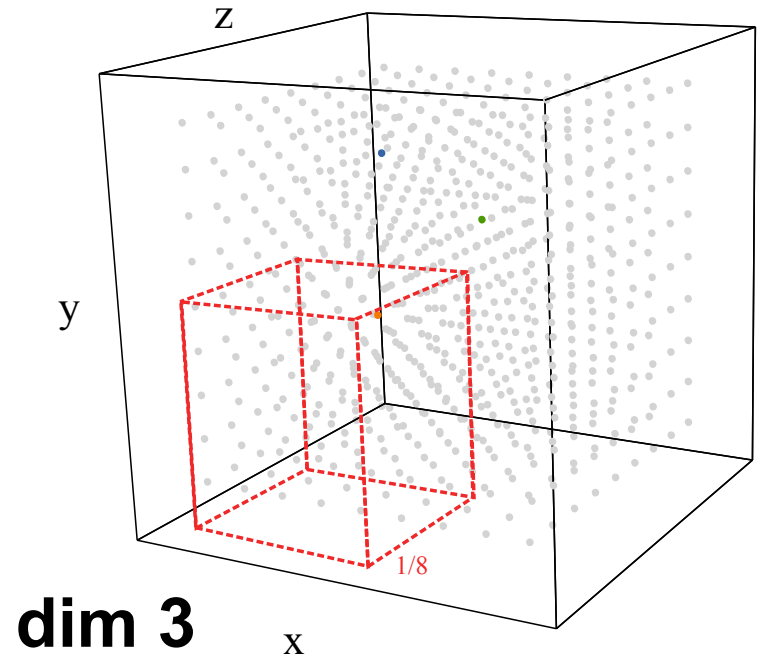
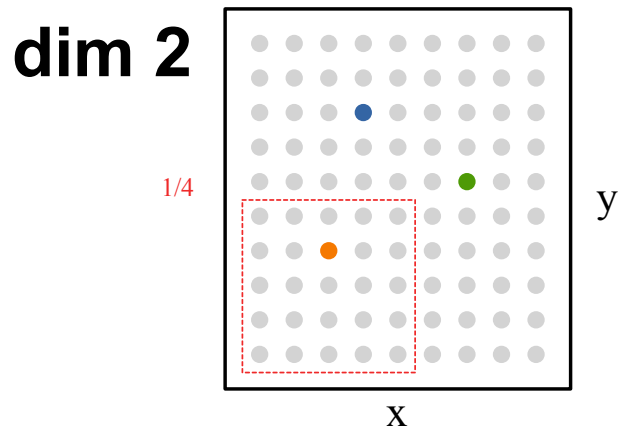
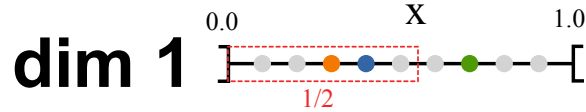
**Durée & Volumétrie impossible à gérer**



# Comment ?



## 2 ème problème - "Curse Dimensionality"

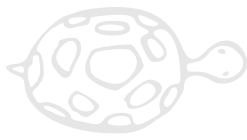


dimensions

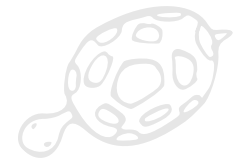
=



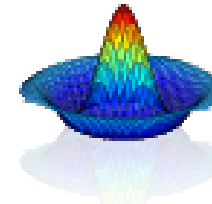
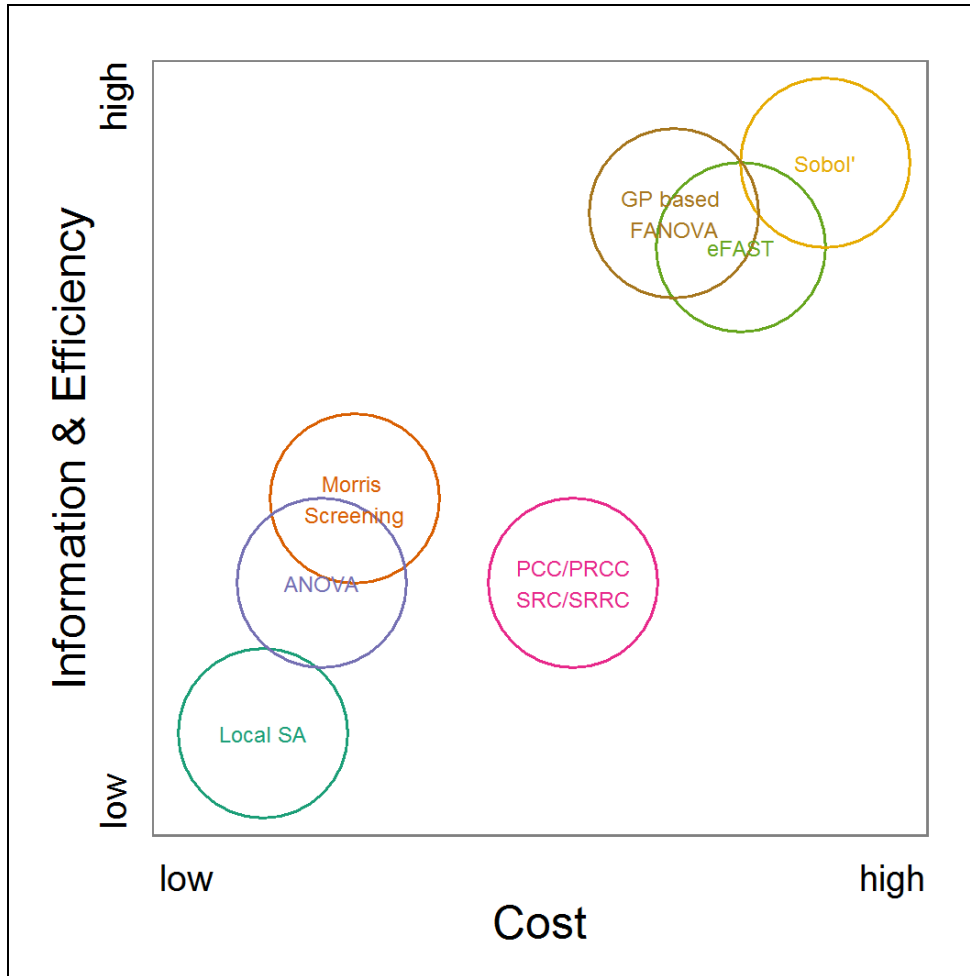
couverture à nb points égal



# Comment ?

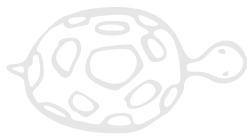


Une première solution, des méthodes plus sioux.

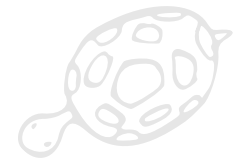


**MEXICO**  
WEXICO

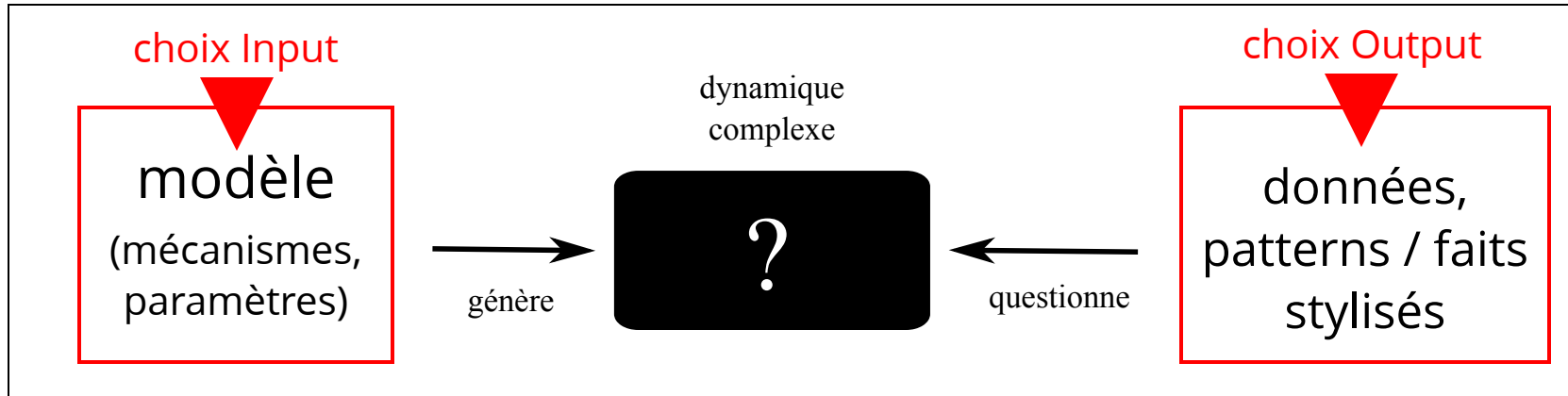
<http://reseau-mexico.fr/>



# Comment ?



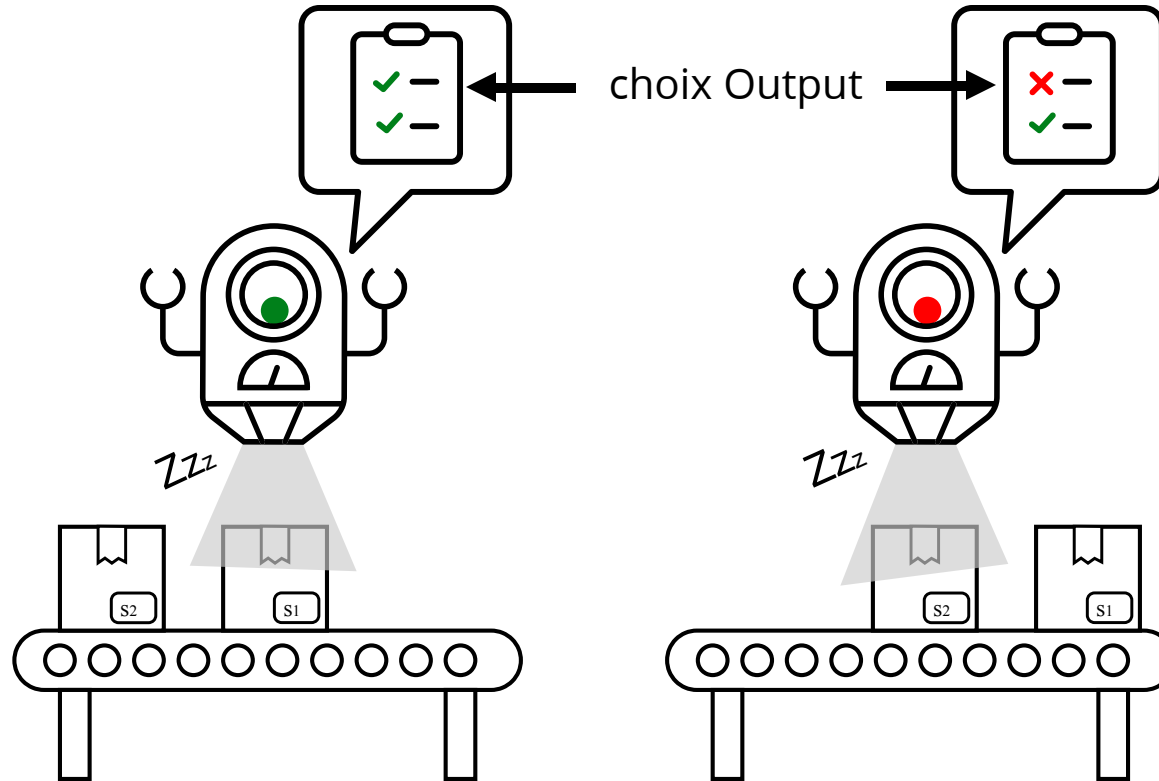
Une deuxième solution, changer d'approche !



Organiser une **mise en tension** entre les  
**choix Input - Output**

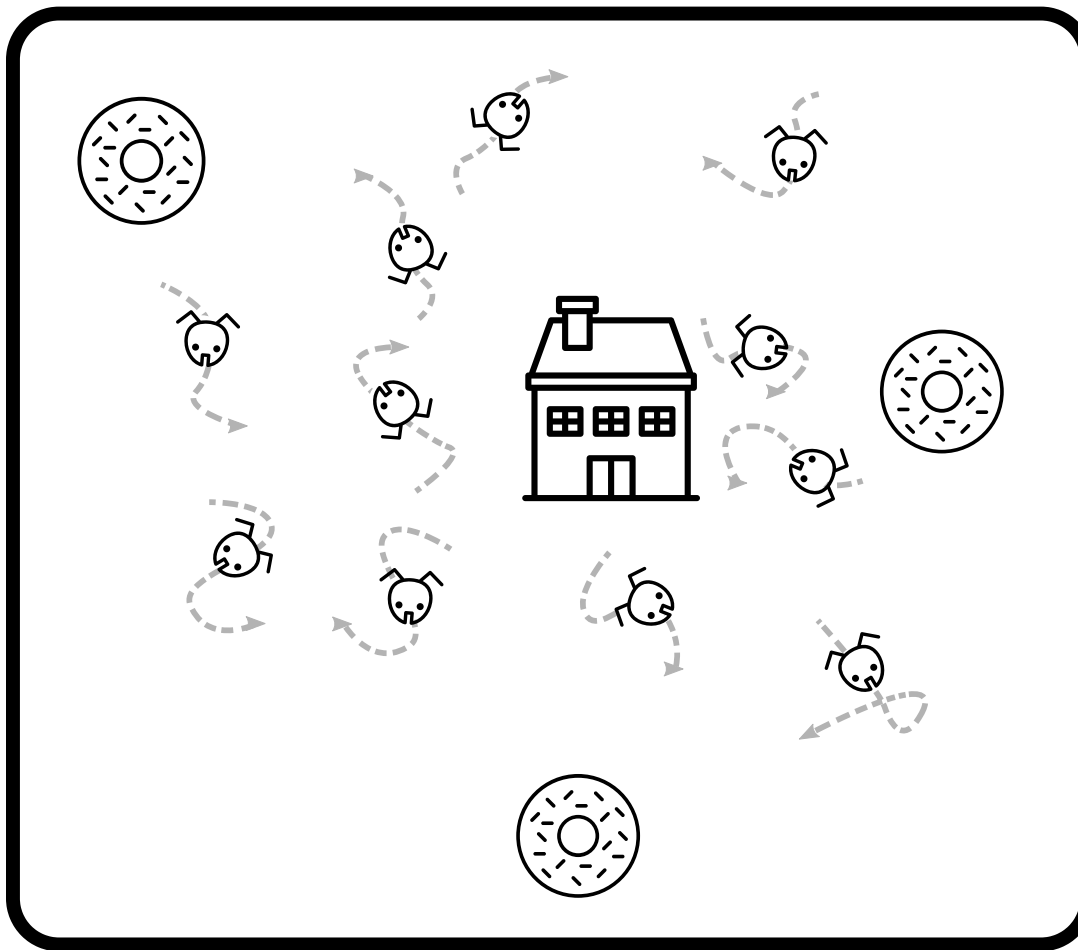
# Comment ?

Quelle mise en tension ?



Des **critères quantifiés** pour "guider"  
une exploration automatisé

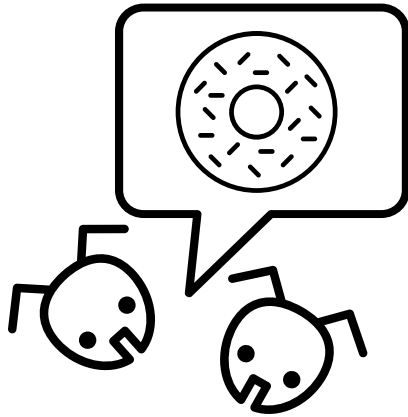
# Example "Ants"



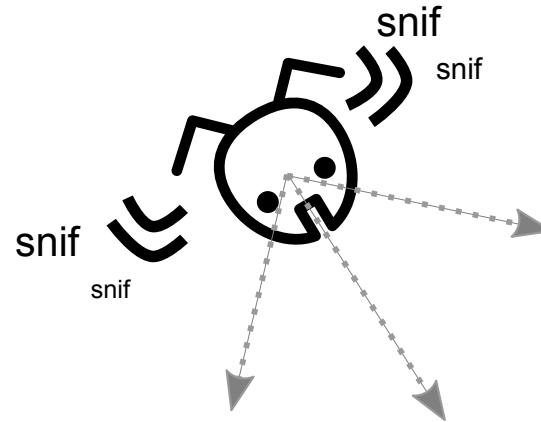
*"random walk"*



# Example "Ants"



Utilisation de traces  
ou "**phéromones**"



Alignement avec les  
traces **les plus fortes**

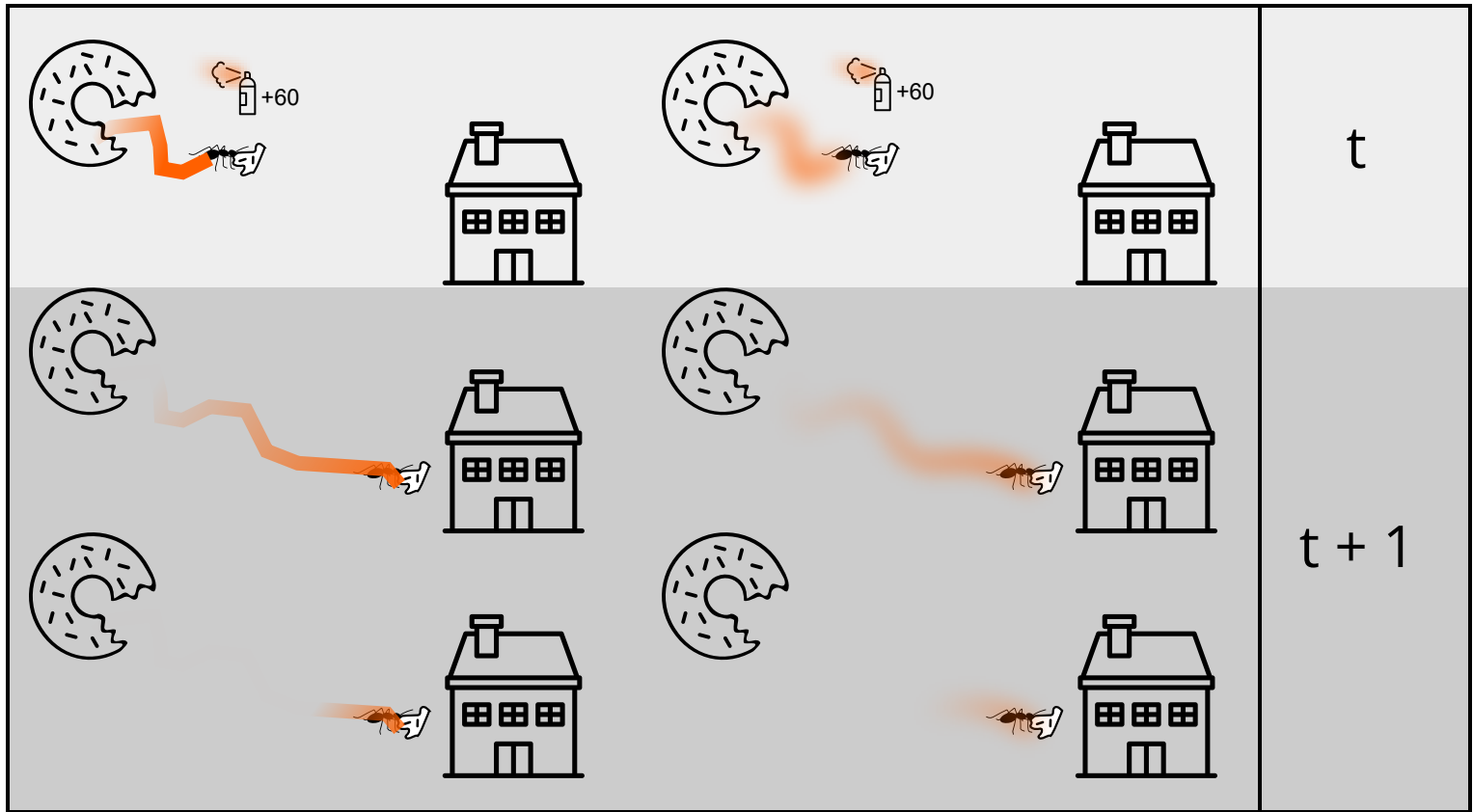
# Example "Ants"



diffusion

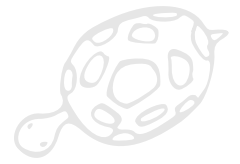


evaporation

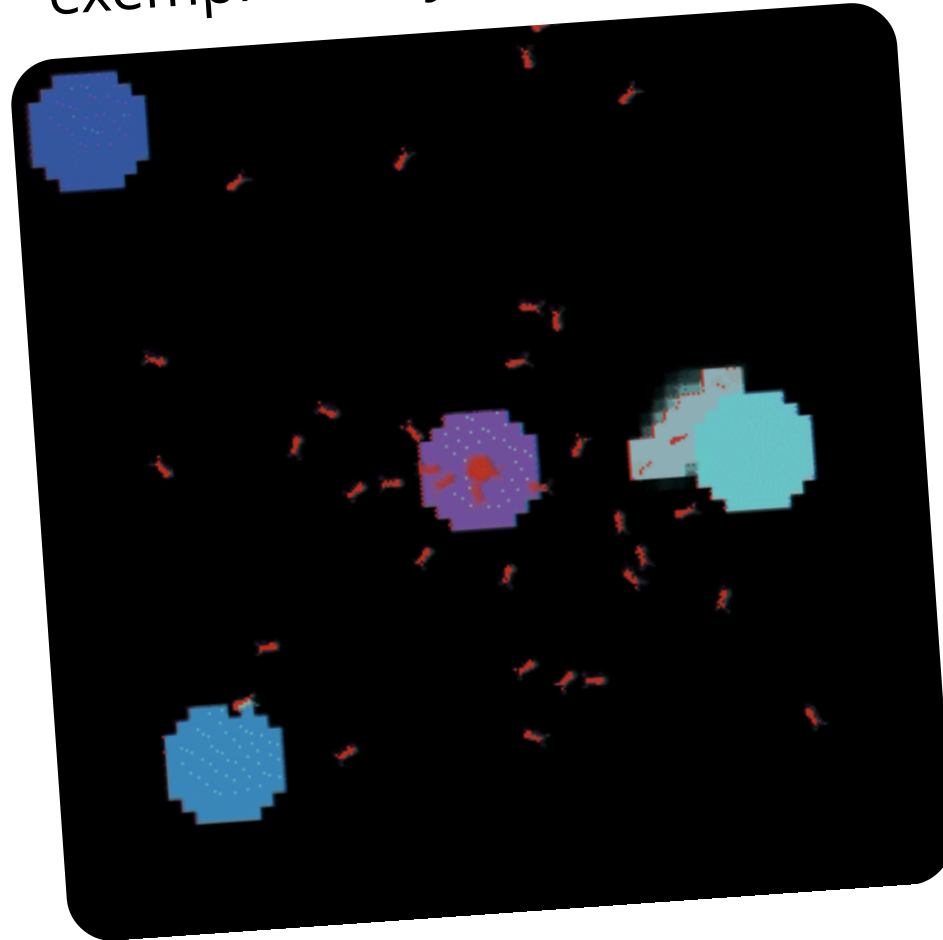




# Example "Ants"

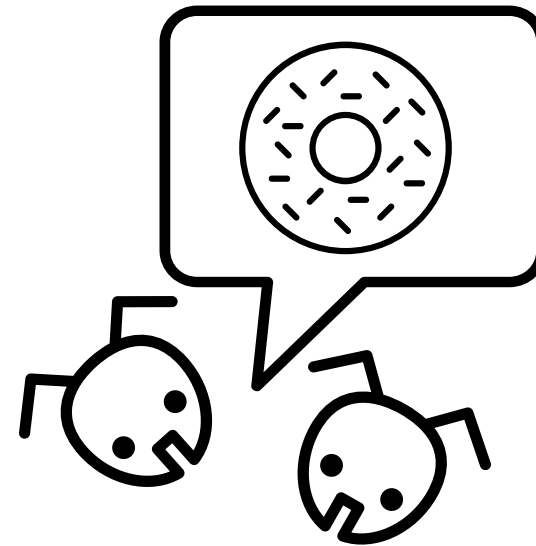
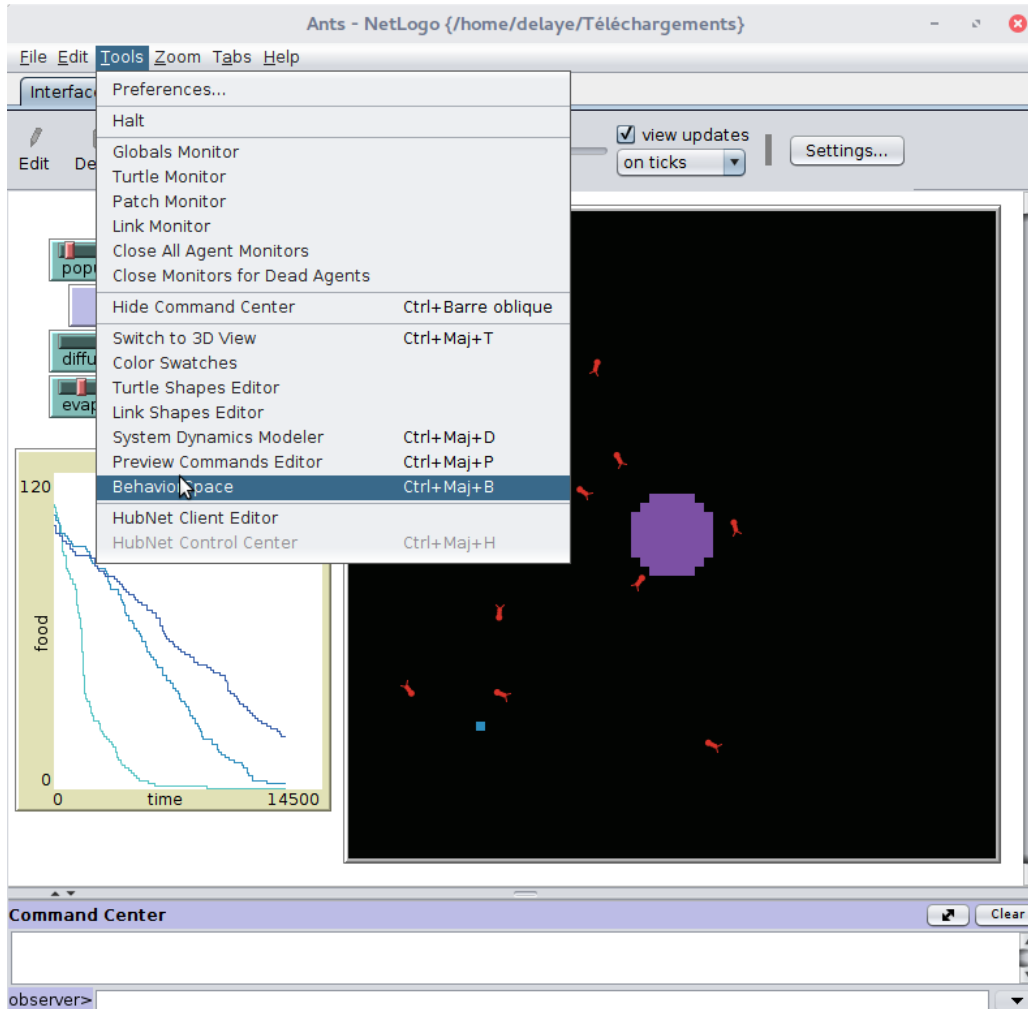


exemple de dynamique



# Example "Ants"

## Le BehaviorSpace



Utilisation de traces  
ou "phéromones"

# Example "Ants"

Le BehaviorSpace : config 4810 simulations

Experiment

Experiment name

Vary variables as follows (note brackets and quotation marks):

```
["evaporation-rate" [10 10 90]]  
["population" [100 20 200]]  
["diffusion-rate" [10 10 90]]
```

Either list values to use, for example:  
["my-slider" 1 2 7 8]  
or specify start, increment, and end, for example:  
["my-slider" [0 1 10]] (note additional brackets)  
to go from 0, 1 at a time, to 10.  
You may also vary max-pxcor, min-pxcor, max-pycor, min-pycor, random-seed.

Repetitions

run each combination this many times

Run combinations in sequential order  
For example, having ["var" 1 2 3] with 2 repetitions, the experiments' "var" values will be:  
sequential order: 1, 1, 2, 2, 3, 3  
alternating order: 1, 2, 3, 1, 2, 3

Measure runs using these reporters:

```
sum [food] of patches
```

one reporter per line; you may not split a reporter across multiple lines

Measure runs at every step  
if unchecked, runs are measured only when they are over

Setup commands:

Go commands:

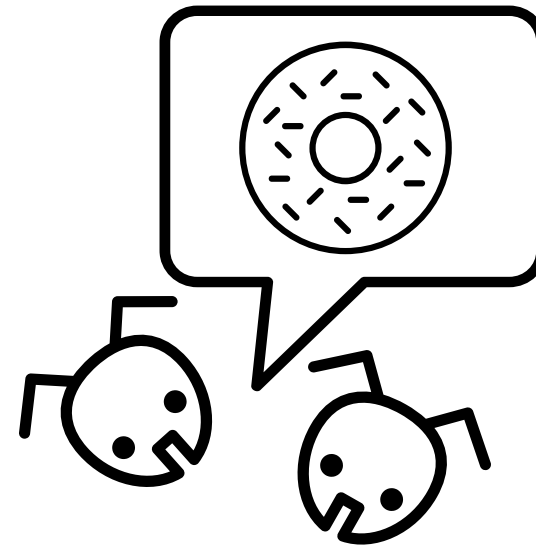
Stop condition:

Final commands:

the run stops if this reporter becomes true      run at the end of each run

Time limit

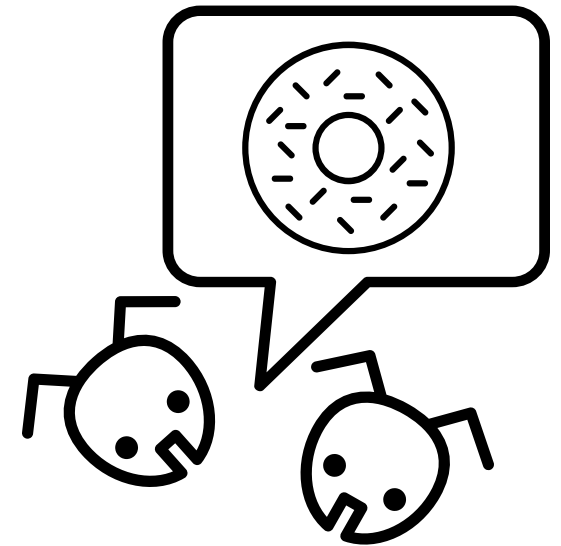
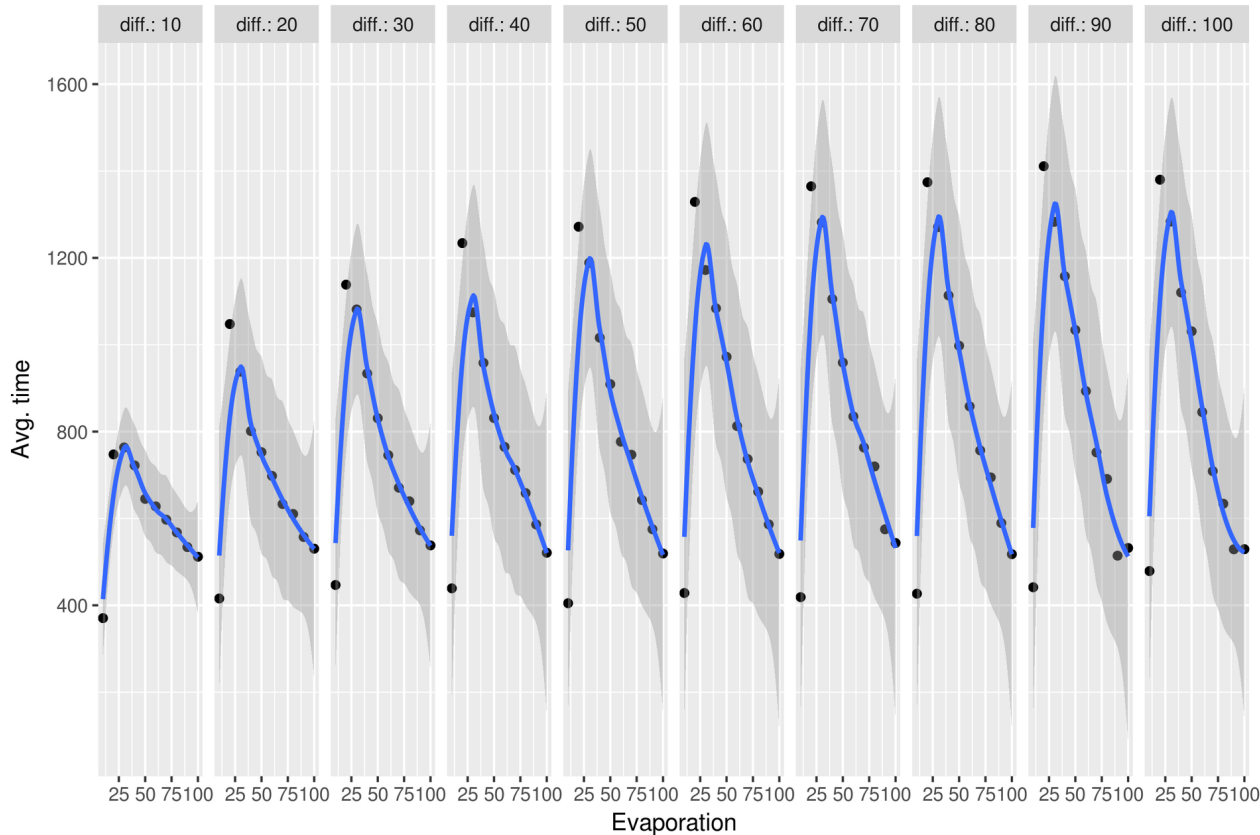
stop after this many steps (0 = no limit)



Utilisation de traces  
ou "phéromones"

# Example "Ants"

Ant : results for pop. 150



Utilisation de traces  
ou "phéromones"

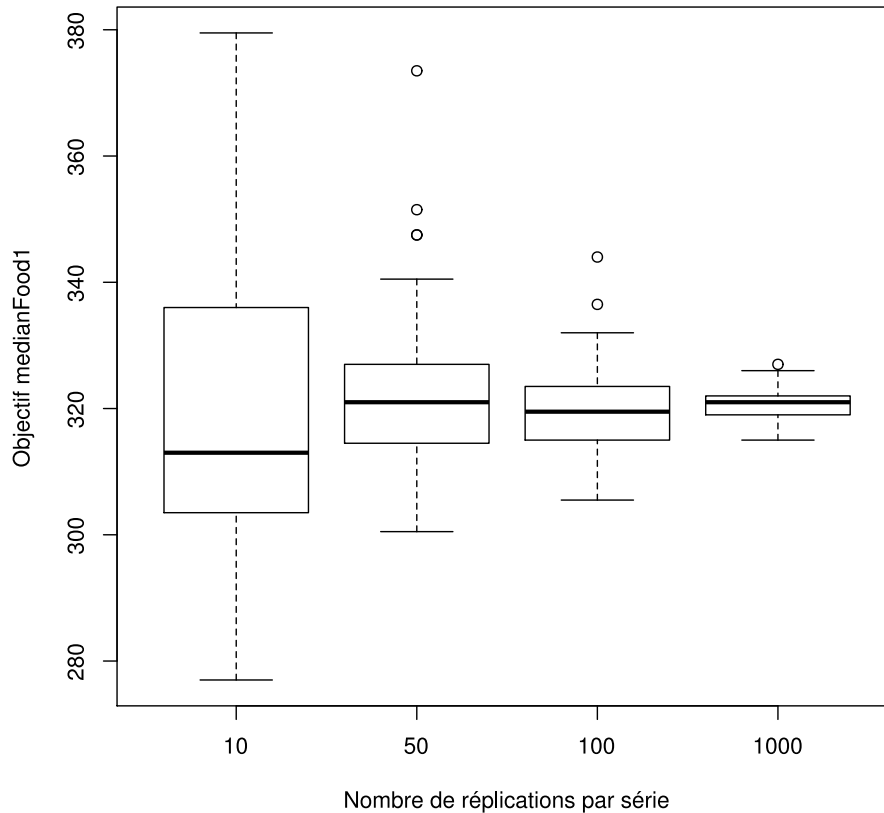


Résultats et script disponible sur [github](#)

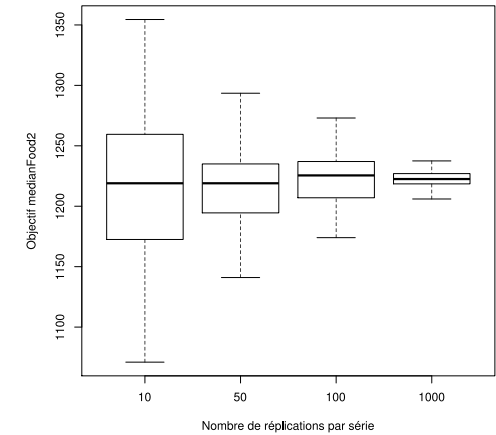
# Exemple "Ants"

Mesurer le rôle de **l'aléatoire** dans les simulations ?

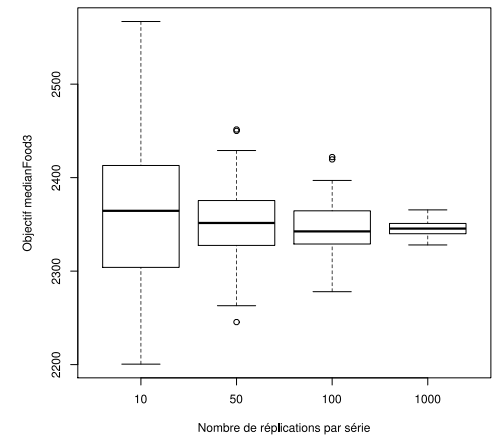
Médiane sur l'objectif food1 sur 100 séries de répliations



Médiane sur l'objectif food2 sur 100 séries de répliations



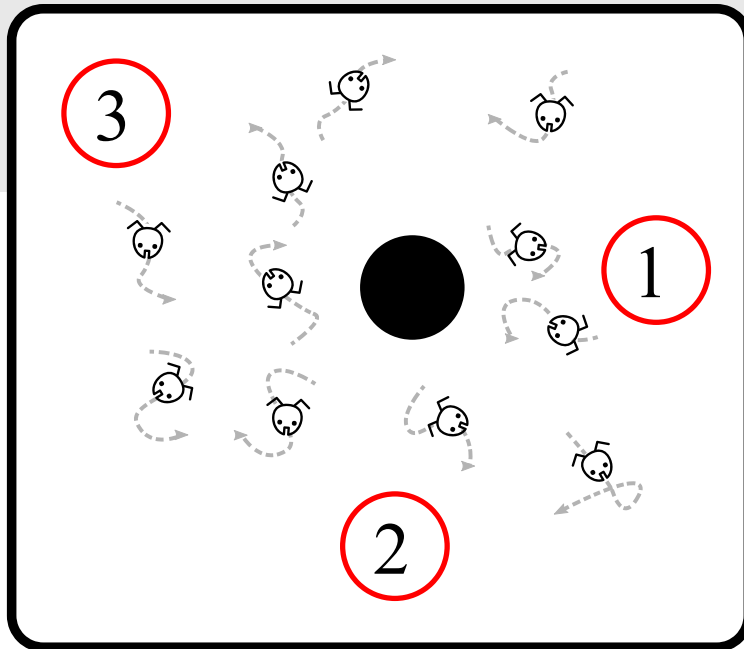
Médiane sur l'objectif food2 sur 100 séries de répliations



100 \* médiane de 10,50,100,1000 répliations

# Example "Ants"

Que veut on tester/savoir ?



Quelle valeurs de (*diffusion, evaporation*) **minimise** le temps de consommation des trois **tas de nourriture** ?

**trois objectifs**

$$Obj_1 = \textit{ticks} \text{ quand } \textcircled{1} \leq 0$$

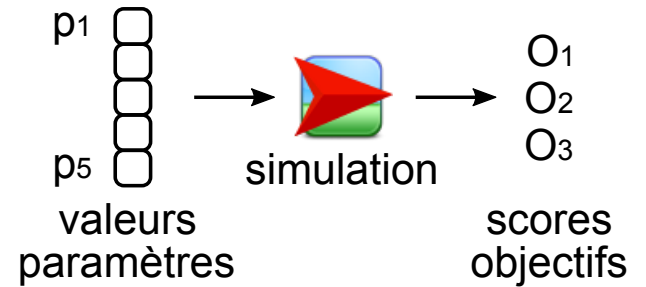
$$Obj_2 = \textit{ticks} \text{ quand } \textcircled{2} \leq 0$$

$$Obj_3 = \textit{ticks} \text{ quand } \textcircled{3} \leq 0$$

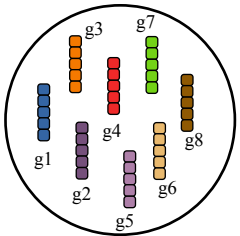


# Comment ?

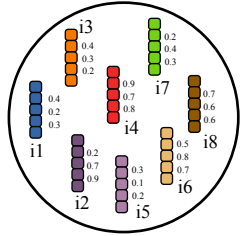
Algorithmes Evolutionnaires (AE)



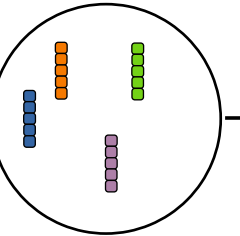
*initialisation*



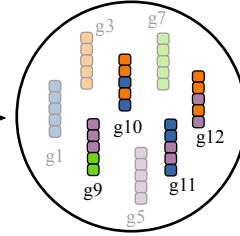
*évaluation (t)*



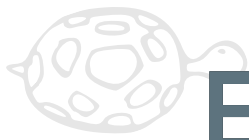
*selection (t)*



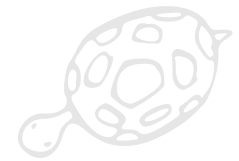
*reproduction (t)*



$(t \rightarrow t + 1)$



# Example "Ants"



## Le Behavior Search

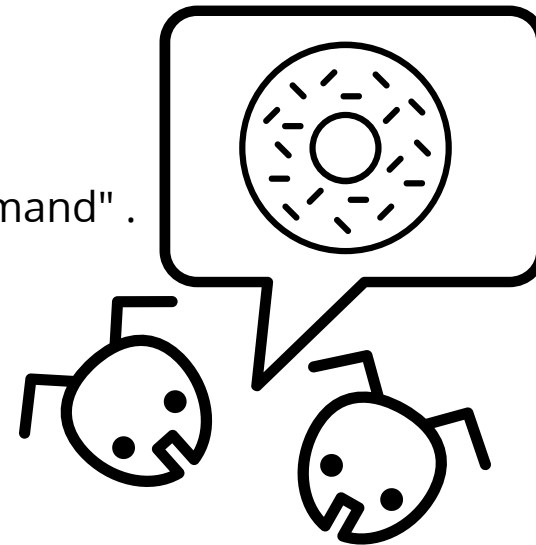
Disponible avec Netlogo depuis la version 6.0.1

**Windows:** Run

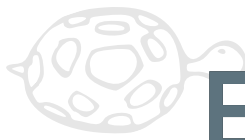
"app\behaviorsearch\behaviorsearch\_headless.bat".

**Mac:** Run "behaviorsearch/behaviorsearch\_gui.command" .

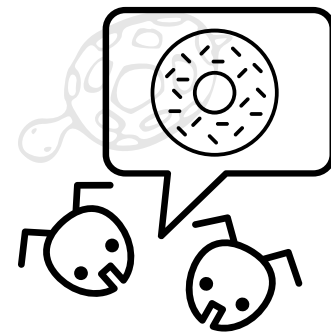
**Linux:** Run "./Behaviorsearch" .



Utilisation de traces  
ou "phéromones"



# Example "Ants"



## Le Behavior Search

Untitled - BehaviorSearch 1.02

File Help

/home/delaye/github/MAPS-10/supports-cours/explorations-modeles/Ants\_es.nlogo Browse for model...

---

**Parameter Specification** ?

```
["diffusion-rate" [0 1 99]]
["evaporation-rate" [0 1 99]]
["population" 150]
```

Load param ranges from model interface

**Setup:** setup  
**Step:** go  
**Measure:** ticks  
**Measure If:** true  
**Stop If:** sugar\_in\_word <= 0  
**Step Limit:** 4000 model steps

---

**Search Method Configuration** ?

StandardGA ?

Parameter	Value
mutation-rate	0.03
population-size	50
crossover-rate	0.7
population-model	generational
tournament-size	3

Use fitness caching

**Search Encoding Representation** ?

GrayBinaryChromosome ?

---

**Objective / Fitness Function** ?

**Goal:** Maximize Fitness ?

**Collected measure:** AT\_FINAL\_STEP

Fixed Sampling 10 replicates

**Combine replicates:** MEAN ?

Take derivative?  Use ABS value?

w.r.t. ----  $\Delta = 0.0$

**Evaluation limit:** 300 model runs

**BestChecking replicates:** 0

Run BehaviorSearch

---

**Search Progress**

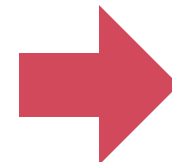
**From all searches:**

**Best found so far:**  
diffusion-rate=91,0000  
evaporation-rate=21,0000  
population=150,000

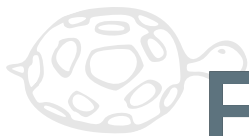
**Fitness= 2883,40**

Finished search 10 of 10: 100% (1:59:10 elapsed - -:27 remaining)

Done



Très lent ...



# Example "Ants"

Aller plus loin : R et NetLogo avec RNetlogo

```

1  ## Un exemple d'utilisation simple de RNetlogo
2  ## Formation MAPS10
3
4  library("RNetLogo")
5
6  #localiser l'installation de netlogo
7  nl.path <- "~/app/NetLogo_6.0.1/app"
8  NLStart(nl.path, gui = FALSE) #lance netlogo avec (TRUE) ou non (FALSE) une gui
9
10 ##Definition du chemin du modèle
11 model.path <- "/home/delaye/github/MAPS-10/supports-cours/explorations-modeles/Ants_es.nlogo"
12 ## chargement du modele dans netlogo
13 NLLoadModel(model.path)
14
15 ##Parametres
16 my.pop <- 120
17 my.diff <- 30
18 my.evap <- 20
19
20
21 ## change parameter
22 NLCommand("set population ", my.pop)
23 NLCommand("set diffusion-rate ", my.diff)
24 NLCommand("set evaporation-rate ", my.evap)
25
26 NLCommand("setup")
27 agent.df<- data.frame() ## data.frame pour stocker les pos des agents
28 j <- 0
29 for(i in 1:2000){ ##On fera 2000 itération ?
30   NLCommand("go")
31   if(j >= 10){ ## à chaque multiple de 10 on sauvegarde la position des agents
32     pos.agents <- NLGetAgentSet(c("who", "xcor", "ycor", "color",
33       "ticks"), "turtles")
34     agent.df <- rbind(agent.df , pos.agents)
35     j = 1
36   }
37   j <- j + 1
38 }
39 }
40
41 save(agent.df, file = "/home/delaye/github/MAPS-10/simulation.RData")
42
43 NLQuit()
44

```





# Example "Ants"



Aller plus loin : R et NetLogo avec RNetlogo

©Copyright JASSS



Jan C. Thiele, Winfried Kurth and Volker Grimm (2014)

Facilitating Parameter Estimation and Sensitivity Analysis of Agent-Based Models: A Cookbook Using NetLogo and R

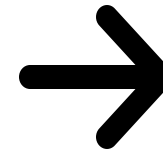
<http://jasss.soc.surrey.ac.uk/17/3/11.html>

# Example "Ants"

Aller encore plus loin ...



**"High Performance Computing"**



1954 - 2017



+



1993

A vertical box containing the text "1954 - 2017" at the top. Below it is the Fortran logo, which consists of the word "Fortran" in a serif font above a small red and white interface element. A plus sign is centered below the Fortran logo. Below the plus sign is the MPI logo, which features three blue squares of varying sizes to the left of the letters "MPI" in a bold, sans-serif font. At the bottom of the box is the year "1993".

**Wait? Really?**





# Example "Ants"

Aller plus loin encore avec **OpenMOLE**

<http://openmole.org>

Créer des "chaînes de traitements"  
**parallélisables, reproductibles, simples** (ou complexes) sans  
**aucune connaissance** des technologies **HPC**

**Plusieurs types de plan d'expériences** : Plan complet, LSH, Analyse de sensibilités, Algorithmes Evolutionnaires, etc.

```
val i = Val[Double]
val res = Val[Double]
```

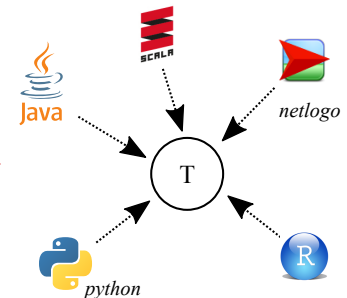
```
val exploration =
  ExplorationTask(i in (0.0 to 100.0 by 1.0))
```

```
val model =
  ScalaTask("val res = i * 2") set (
    inputs += i,
    outputs += (i, res)
  )
```

```
val env = LocalEnvironment(4)
```

```
exploration -< (model on env hook ToStringHook())
```

**Plusieurs tâches différentes !**

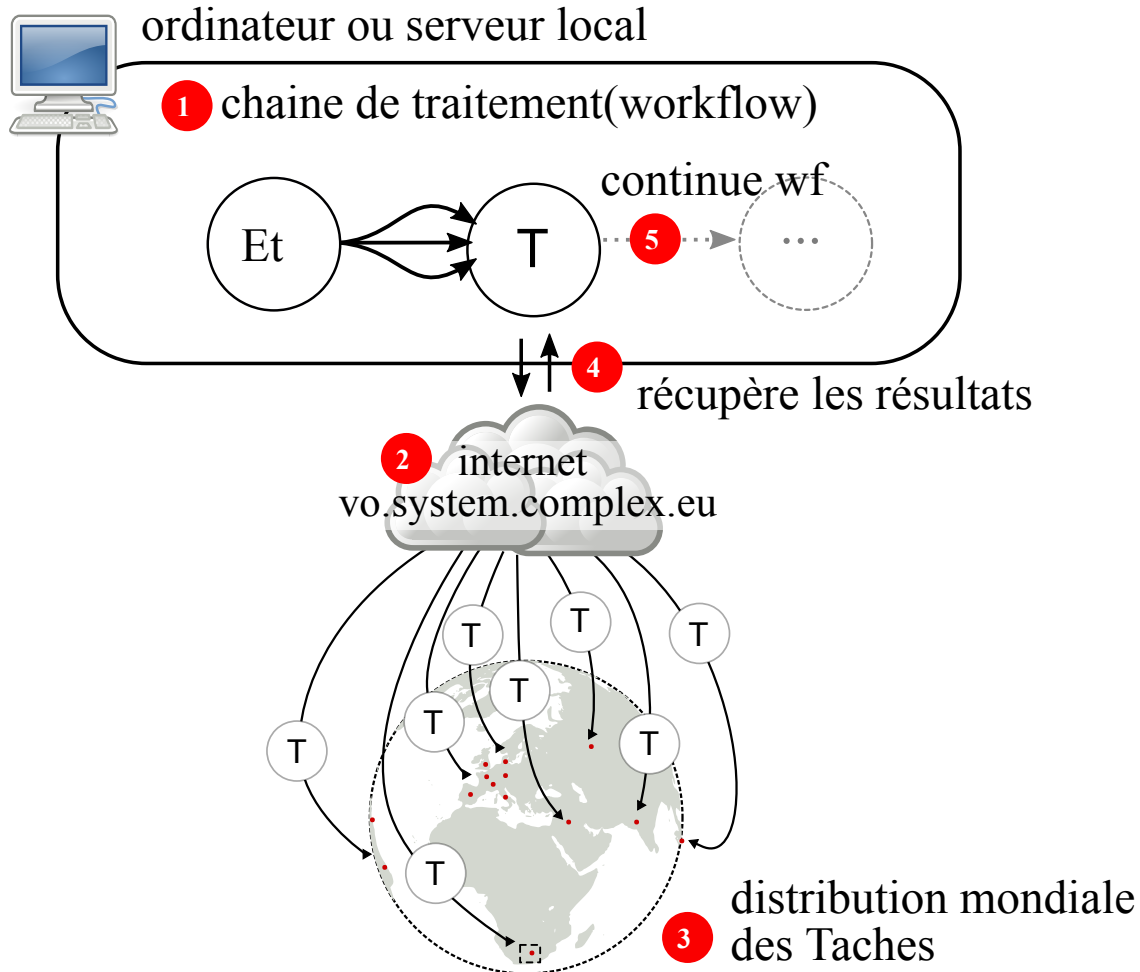


**Support de multiples technologies HPC** : Cluster, grille de calcul, etc.

Langage de scripts (DSL) **simple et flexible** pour composer les workflow

# Example "Ants"

Aller plus loin encore avec OpenMOLE





# Example "Ants"



<http://openmole.org>

D'un plan d'expérience simple ...

```
// Define the input variables
val gPopulation = Val[Double]
val gDiffusionRate = Val[Double]
val gEvaporationRate = Val[Double]
val seed = Val[Int]
val i = Val[Double]

// Define the output variables
val food1 = Val[Double]
val food2 = Val[Double]
val food3 = Val[Double]

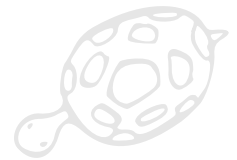
val exploration =
  ExplorationTask(
    (gDiffusionRate in (1.0 to 100.0 by 10.0)) x
    (gEvaporationRate in (1.0 to 100.0 by 10.0)) x
    (seed in (UniformDistribution[Int]() take 5)))

// Define the NetlogoTask
val cmds = Seq("random-seed ${seed}", "run-to-grid")
val ants =
  NetLogo5Task(workDirectory / "ants.nlogo", cmds) set (
    // Map the OpenMOLE variables to NetLogo variables
    netLogoInputs += (gPopulation, "gpopulation"),
    netLogoInputs += (gDiffusionRate, "gdiffusion-rate"),
    netLogoInputs += (gEvaporationRate, "gevaporation-rate"),
    netLogoOutputs += ("final-ticks-food1", food1),
    netLogoOutputs += ("final-ticks-food2", food2),
    netLogoOutputs += ("final-ticks-food3", food3),
    gPopulation:= 100.0,
    // The seed is used to control the initialisation of the random number generator of NetLogo
    inputs += seed,
    outputs += (seed, gDiffusionRate, gEvaporationRate)
  )

val saveHook = AppendToCSVFileHook(workDirectory / "ant_exploration.csv")

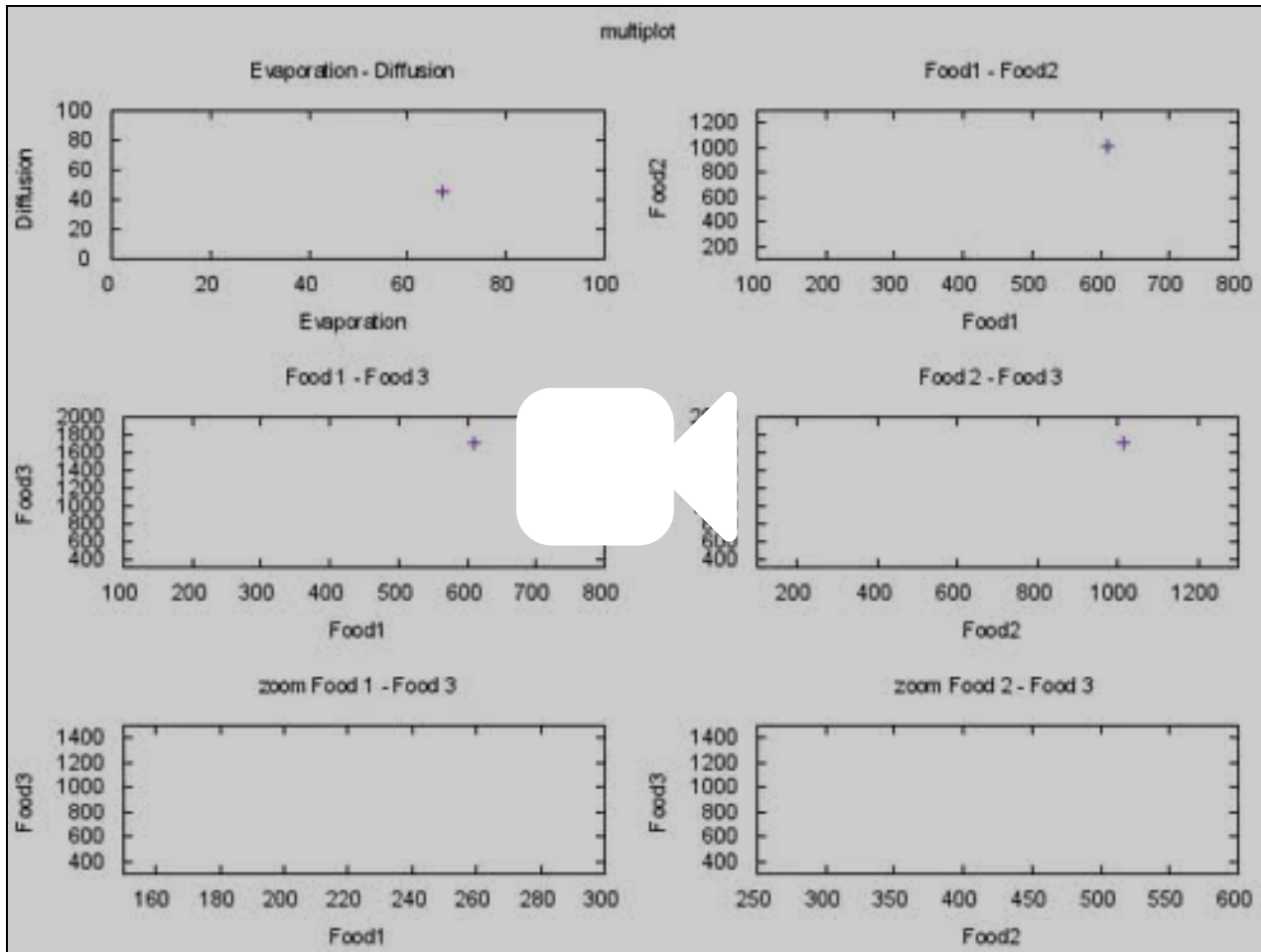
val env = LocalEnvironment(4)

exploration -< (ants on env) hook saveHook
```



# Example "Ants"

à une optimisation multi-critères ...



# Pour aller plus loin

## références et entrées vers + de biblio

**[Banos2016]** Banos, A., Lang, C., & Marilleau, N. (2016). *Agent-based spatial simulation with NetLogo* Volume 2: Advanced Concepts. Elsevier. [url](#)

**[Banos2016]** Banos, Arnaud (2016) Modéliser c'est apprendre : Itinéraire d'un géographe, Edition Matériologique

**[Chérel2015]** Chérel G., Cottineau C., Reuillon R., 2015, « Beyond Corroboration : Strengthening Model Validation by Looking for Unexpected Patterns. », PLoS ONE 10(9), e0138212. [doi:10.1371/journal.pone.0138212](https://doi.org/10.1371/journal.pone.0138212)

**[Cottineau2015]** Cottineau C., Reuillon R., Chapron P., Rey-Coyrehourcq S., Pumain D., 2015, "A Modular Modelling Framework for Hypotheses Testing in the Simulation of Urbanisation.", *Systems*, 3, Numéro Spécial "Agent-Based Modelling of City Systems", 348-377. DOI : [10.3390/systems3040348](https://doi.org/10.3390/systems3040348)

**[Cottineau2015]** Cottineau C., Chapron P., Reuillon R., 2015, "Growing models from the bottom up. An evaluation-based incremental modelling method (EBIMM) applied to the simulation of systems of cities", *Journal of Artificial Societies and Social Simulation (JASSS)*, Vol. 18, No. 4, 9. DOI : [10.18564/jasss.2828](https://doi.org/10.18564/jasss.2828).

**[Cottineau2016]** Cottineau, Clémentine, Rey-Coyrehourcq Sébastien (2016) "Back to the future of multi modelling" , Conférence RGS , [slides](#)

**[Delay2015]** Delay E., « Réflexions géographiques sur l'usage des systèmes multi-agents dans la compréhension des processus d'évolution des territoires viticoles de fortes pentes : Le cas de la Côte Vermeille et du Val di Cembra », Thèse de doctorat, Université de Limoges, Limoges, 2015. [HAL-SHA](#)

**[Rey2015]** REY-COYREHOURCQ, Sébastien (2015, October 13) "Une plateforme intégrée pour la construction et l'évaluation de modèles de simulation en géographie" Thèse Paris 1. <https://zenodo.org/record/50212>

**[Reuillon2015]** Reuillon, R., Schmitt, C., De Aldama, R., & Mouret, J.-B. (2015). A New Method to Evaluate Simulation Models: The Calibration Profile (CP) Algorithm. *Journal of Artificial Societies and Social Simulation*, 18(1), 12. Retrieved from <http://jasss.soc.surrey.ac.uk/18/1/12.html>

**[Reuillon2013]** Reuillon, R., Leclaire, M., & Rey-Coyrehourcq, S. (2013). OpenMOLE, a workflow engine specifically tailored for the distributed exploration of simulation models. *Future Generation Computer Systems*, 29(8), 1981–1990. <https://doi.org/http://dx.doi.org/10.1016/j.future.2013.05.003>

**[Schmitt2015]** Schmitt, C., Rey, S., Reuillon, R., & Pumain, D. (2015). Half a billion simulations: Evolutionary algorithms and distributed computing for calibrating the SimpopLocal geographical model. *Environment and Planning B*, 42(2), 300–315. [url](#)

**[Thiele2014]** Thiele, Jan C., Kurth, Winfried and Grimm, Volker (2014) 'Facilitating Parameter Estimation and Sensitivity Analysis of Agent-Based Models: A Cookbook Using NetLogo and 'R'' *Journal of Artificial Societies and Social Simulation* 17 (3) 11 <<http://jasss.soc.surrey.ac.uk/17/3/11.html>>. doi: 10.18564/jasss.2503

++ David O'Sullivan !! ++

++ Volker GRIMM !! ++



# Crédits icones

## **NounProject CC Commons :**

Indigo, David, Icon Fair, Corpus Delicti, Daniel Aczel, Mikicon, Iconic, Fuat Sanli, Sergey Demushkin, Chanut is Industries