

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ENGENHARIA DA COMPUTAÇÃO

ANA CAROLINA MENDES FORTES
JOSÉ JOÃO MONTEIRO COSTA
LUCAS NATTAN PEREIRA SILVA
NARGYLLA FERNANDA CLOVIEL LIMA

AGENTES:

Modificando o modelo de formigas em NetLogo

São Luís, MA

2024

ANA CAROLINA MENDES FORTES
JOSÉ JOÃO MONTEIRO COSTA
LUCAS NATTAN PEREIRA SILVA
NARGYLLA FERNANDA CLOVIEL LIMA

AGENTES:

Modificando o modelo de formigas em NetLogo

Trabalho apresentado à disciplina de Inteligência Artificial, ministrada pelo Profº Dr Thales Levi Azevedo Valente, para compor a nota do primeiro bimestre.

São Luís, MA

2024

1. INTRODUÇÃO

O presente trabalho traz o conceito e a simulação de agentes reativos baseados em modelos, um assunto fundamental na área de Inteligência Artificial. Um agente é qualquer entidade capaz de perceber o ambiente ao seu redor através de sensores e de interagir com esse ambiente por meio de atuadores (Russel; Norving, 2003). Os agentes utilizados neste trabalho são agentes baseados em modelos, que, ao contrário dos agentes reativos simples, armazenam informações sobre o ambiente e as utilizam para formar uma representação interna do mundo (Alves; Diogo, 2010). Esses agentes atualizam constantemente seu modelo com base nas percepções obtidas, permitindo decisões mais informadas e adaptativas, com o objetivo de se aproximar cada vez mais da realidade do ambiente.

Para simular esse ambiente, foi utilizado o modelo computacional desenvolvido em NetLogo, inspirado no comportamento de colônia de formigas. A simulação mostra as formigas atuando como agentes, usando pistas químicas (feromônios) e interações locais para buscar e transportar alimentos até o ninho. O estudo busca entender como esses agentes, ao contrário dos simples agentes reativos, utilizam informações sobre o ambiente para formar representações internas (modelos), o que permite que tomem decisões mais adaptativas e interajam com os obstáculos ou desafios apresentados.

2. Desenvolvimento

Inicialmente, foi pensado em criar um modelo em que duas espécies de formigas diferentes deveriam coletar os alimentos e retornarem ao seu respectivo ninho. A primeira mudança realizada foi mudar a cor das formigas para azul, e distribuí-las em dois ninhos diferentes. Utilizando o **set color blue**.

```
to setup [ set size 2 ;; mais fácil de ver
clear-all set color blue
set-default-shape turtles "bug" ifelse who < population / 2 [
create-turtles population move-to patch 0 0
set ninho 1
] [
move-to patch (0.6 * max-pxcor) (0.6 * max-pycor)
set ninho 2
```

Depois disso, foi feito um estudo aprofundado do código para dividir o ninho em dois. Utilizando a documentação e IA generativa para entender um pouco mais do sistema, e realizar essa mudança. Utilizando esse trecho, os ninhos foram

```
ifelse who < population / 2 [
move-to patch 0 0
set ninho 1
] [
move-to patch (0.6 * max-pxcor) (0.6 * max-pycor)
set ninho 2
```

Para definir a posição dos ninhos no espaço, foi utilizado o trecho de ask patches, definindo assim a distância do ninho 1 e do ninho 2.

```
ask patches [
;; Ninho 1 no centro (0, 0)
ifelse distancexy 0 0 < 5 [
set nest-scent 200 - distancexy 0 0
]
[
;; Ninho 2 em (0.6 * max-pxcor, 0.6 * max-pycor)
ifelse distancexy (0.6 * max-pxcor) (0.6 * max-pycor) < 5 [
set nest-scent 200 - distancexy (0.6 * max-pxcor) (0.6 * max-pycor)
]
[
set nest-scent 0
]
]
end
```

Uma parte importante das ações das formigas é o ato de ir em busca da comida e retornar para o ninho. Para realizar isso, utilizou-se o trecho do código seguinte.

Declarando a função de **Setup-Food**, foi definido o tamanho e a distância do alimento das formigas, de maneira que fosse visualmente interessante. Para isso, também definimos as cores usando a função **recolor-patch**.

```
to setup-food ;; patchn procedure
;; setup food source one on the right
if (distancexy (0.6 * max-pxcor) 0) < 5
[ set food-source-number 1 ]
;; setup food source two on the lower-left
if (distancexy (-0.6 * max-pxcor) (-0.6 * max-pycor)) < 5
[ set food-source-number 2 ]
;; setup food source three on the upper-left
if (distancexy (-0.8 * max-pxcor) (0.8 * max-pycor)) < 5
[ set food-source-number 3 ]
;; set "food" at sources to either 1 or 2, randomly
if food-source-number > 0
[ set food one-of [1 2] ]
end

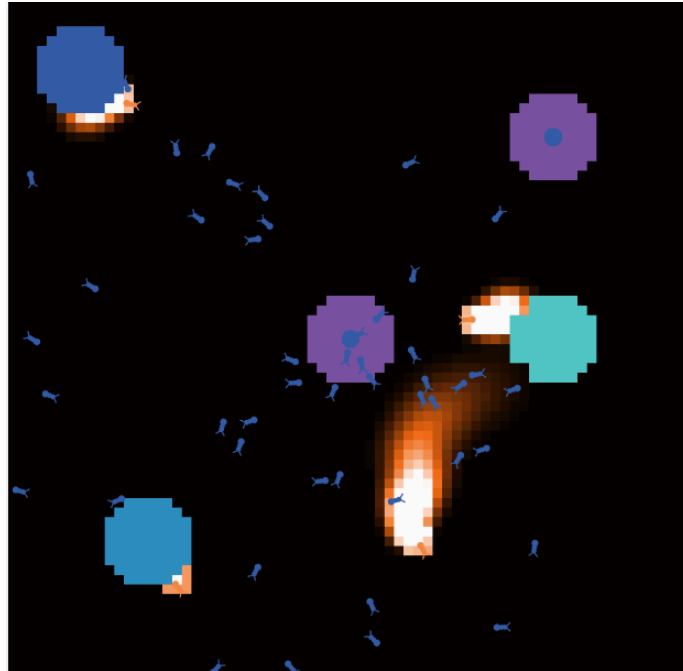
to recolor-patch ;; patch procedure
;; give color to nest and food sources
ifelse nest?
[ set pcolor violet ]
[ ifelse food > 0
[ if food-source-number = 1 [ set pcolor cyan ]
if food-source-number = 2 [ set pcolor sky ]
if food-source-number = 3 [ set pcolor blue ] ]
;; scale color to show chemical concentration
[ set pcolor scale-color orange chemical 0.1 5 ] ]
end
.....
```

Para a execução do código, os procedimentos definidos foram os seguintes. O ciclo principal é controlado por um botão **forever**, onde cada formiga avalia seu estado e toma decisões baseadas no ambiente. Quando a formiga não está carregando comida (indicada pela cor azul), ela procura por comida utilizando o procedimento **look-for-food**. Caso esteja carregando comida, ela segue o procedimento **return-to-nest** para retornar ao seu ninho, depositando substâncias químicas no caminho, o que cria um rastro que outras formigas podem seguir. O movimento das formigas é aleatório, com elas realizando desvios a cada passo, representado pelo comando **wiggle**. Além disso, a difusão e a evaporação das substâncias químicas são simuladas, com o código ajustando a quantidade de química no ambiente e nos patches ao longo do tempo, com base em taxas de difusão e evaporação. As formigas utilizam o cheiro das substâncias químicas no ambiente para tomar decisões, como seguir o rastro de comida ou do ninho, ajustando sua direção com base na intensidade do cheiro à frente, à direita ou à esquerda. Esse comportamento coletivo e a interação com o ambiente criam uma simulação dinâmica e realista do forrageamento e da comunicação entre as formigas, com base em princípios simples de busca e trilhas químicas.

```
to go |;; forever button
ask turtles
[ if who >= ticks [ stop ] ;; delay initial departure
ifelse color = blue
[ look-for-food ] ;; not carrying food? look for it
[ return-to-nest ] ;; carrying food? take it back to nest
wiggle
fd 1 ]
diffuse chemical (diffusion-rate / 100)
ask patches
[ set chemical * (100 - evaporation-rate) / 100 ;; slowly evaporate chemical
recolor-patch ]
tick
end
```

2. Resultados

Os resultados que ocorreram com as mudanças podem ser analisados abaixo na figura abaixo. Não obtivemos para realizar toda a ideação porque não conseguimos fazer a formiga se diferenciar, porém, conseguimos dividir o ninho e fazer as formigas coletarem os alimentos.



4. Conclusão

A configuração do ambiente no código NetLogo estabelece um ecossistema dinâmico onde agentes simulam o comportamento de formigas, interagindo com ninhos, fontes de comida e rastros químicos. Essa estrutura fornece uma base robusta para estudar agentes reativos baseados em modelos, destacando como estímulos externos e estados internos guiam ações, permitindo uma compreensão prática dos princípios da inteligência artificial aplicada. Conseguimos compreender a funcionalidade dos agentes reativos na prática principalmente os que são baseados em modelos, que foi o que nós abordamos no nosso trabalho.