

Simulação de Tráfego: Coleta de Lixo

Modelo Baseado em Agentes Utilizando a Ferramenta NetLogo

Renan Ruseler
Engenharia de Software
UDESC - Ceavi
Ibirama – Santa Catarina
renan.ruseler@gmail.com

Rodrigo Souza Tassoni
Engenharia de Software
UDESC - Ceavi
Ibirama – Santa Catarina
rodrigo.tassoni@edu.udesc.br

RESUMO

Neste trabalho, um modelo de simulação de tráfego baseado em agentes usando a ferramenta NetLogo. Neste estudo de caso, um agente especial foi criado com o objetivo de fazer a coleta de lixo na região. Para isso, o agente deve percorrer todas as vias do ambiente e retornar até seu vértice de origem com o menor tempo. O objetivo do estudo de caso é usar a simulação para determinar a hora do dia e o percurso que esse agente deve fazer para cumprir sua tarefa no menor tempo possível. A principal medida de desempenho neste estudo de caso é o tempo gasto para coleta de lixo.

PALAVRAS-CHAVE

algoritmo, agente, controle, tráfego

Formato de Referência:

Renan Ruseler e Rodrigo Souza Tassoni. 2022. Simulação de Tráfego: Coleta de Lixo (B5EAG). ACM, Ibirama, Santa Catarina, Brasil.

Introdução

O NetLogo é um ambiente de modelagem programável para simular fenômenos naturais e sociais. Ele é uma ferramenta adequada para modelar sistemas complexos e que podem evoluir com o passar do tempo. Os programadores podem dar instruções a centenas ou milhares de agentes que operam de forma independente, o que torna possível explorar a conexão entre o comportamento dos indivíduos e os padrões que emergem da interação entre muitos desses indivíduos.

O ambiente de desenvolvimento integrado baseado em agentes permite que seja possível implementar um cenário específico de maneira bem simples, pois ele pode funcionar com apenas dois tipos de objetos, que são os patches e os turtles. O cenário pode ter qualquer dimensão NxM e é possível ter qualquer número de turtles. Todo modelo é iniciado com um cenário e um número de agentes e vai mudando à medida em que o tempo passa e os agentes interagem entre si.

Abordagem Proposta

No modelo proposto, usou-se como métrica de tempo a quantidade de 15 ticks, que são equivalentes a uma hora. A simulação foi iniciada no tick 0, que seria equivalente ao horário de 06:00. Na simulação, podemos inserir valores de ticks no campo “time-to-go”. Esses valores inseridos representam o horário de partida do caminhão de lixo do primeiro vértice.

O trânsito foi configurado para os seguintes horários:

- 15 tix/06:00 = 50 carros
- 15 tix/06:00 = 75 carros
- 15 tix/06:00 = 80 carros
- 15 tix/06:00 = 60 carros
- 70 tix/11:40 = 75 carros
- 70 tix/11:40 = 80 carros
- 150 tix/17:00 = 50 carros
- 150 tix/17:00 = 90 carros

Temos também os campos de exibição “Total Time” e “Tempo Parado”, que mostram os valores que foram coletados para as métricas.

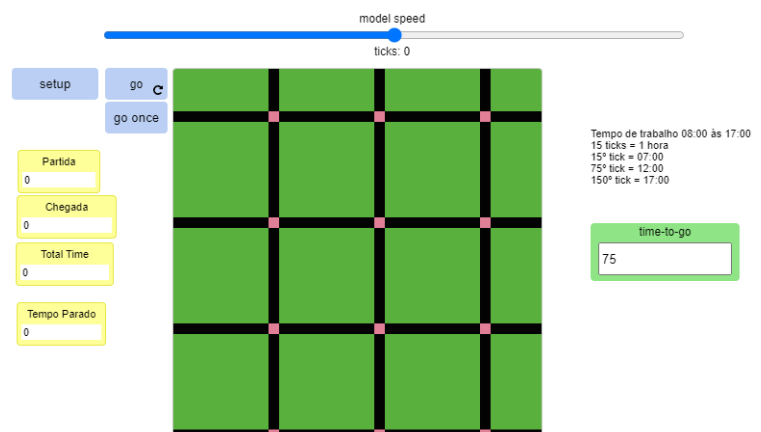


Figura 1: Representação do mapa antes da execução.

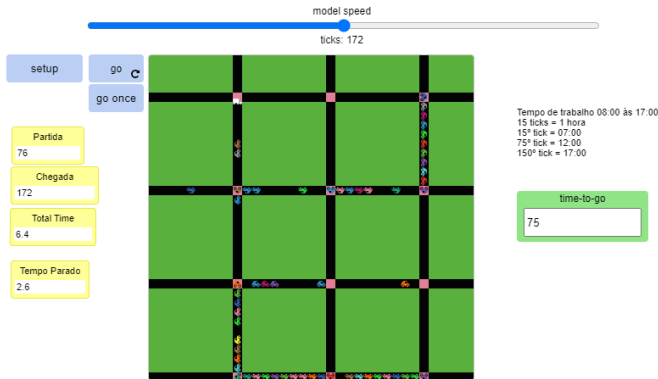


Figura 2: Representação do mapa após uma execução.

```

1  globals [
2
3    intersections
4    roads
5    spawned-cars1
6    spawned-cars2
7    spawned-cars3
8    spawned-cars4
9    spawned-cars5
10   spawned-cars6
11   spawned-cars7
12   spawned-cars8
13   spawned-cars9
14   spawned-trucks
15   time-start
16   time-finish
17   time-stopod
18   stop-condition?
19   time
20   truck-died?
21 ]

```

Figura 3: Globals.

```

23  breed [ lights light ]
24
25
26  breed [ cars car ]
27  breed [ carrs carr ]
28  breed [trucks truck]
29
30  cars-own [
31    speed
32  ]
33  carrs-own [
34    speed
35  ]
36  trucks-own [
37    speed
38  ]
39
40  patches-own
41  [
42    intersection?
43  ]

```

Figura 4: Breed.

```

45  to setup
46    clear-all
47    set-default-shape lights "square"
48    set-default-shape cars "car"
49    set-default-shape carrs "car"
50    set-default-shape trucks "truck"
51    set truck-died? false
52    ask patches
53    [
54      set intersection? false
55      set pcolor green
56    ]
57
58    ;; initialize the global variables that hold patch agentsets
59    set roads patches with
60      [(floor((pxcor + max-pxcor - floor(10 - 1)) mod 10) = 0) or
61       (floor((pycor + max-pycor) mod 10) = 0)]
62    set intersections roads with
63      [(floor((pxcor + max-pxcor - floor(10 - 1)) mod 10) = 0) and
64       (floor((pycor + max-pycor) mod 10) = 0)]
65
66    ask roads [ set pcolor black ]
67    ask intersections [ set pcolor pink
68
69
70  ]
71
72  reset-ticks
73  end

```

Figura 5: To setup.

```

75  to go
76
77    ask cars [ move1 ]
78    ask carrs [ move2 ]
79    ask trucks [move-truck]
80
81
82
83    make-new-car2 100 -8 -17 0
84    make-new-car3 100 2 13 180
85    make-new-car4 100 12 13 180
86    make-new-car5 100 12 3 270
87    make-new-car6 100 2 3 270
88    make-new-car7 100 -8 -7 90
89    make-new-car8 100 12 -17 270
90    make-new-truck 60 -8 13 90
91
92    if truck-died? [ stop ]
93    set time ticks
94
95    tick
96  end

```

Figura 6: To go.

```

98  to make-new-truck [ freq x y h ]
99
100   if (random-float 100 < freq) and not any? turtles-on patch x y and spawned-trucks < 1 and time-to-go = time[
101     create-trucks 1 [
102       set time-start ticks
103
104       setxy x y
105       set heading h
106       set color white
107       set spawned-trucks spawned-trucks + 1
108       adjust-speed
109     ]
110   ]
111
112  end

```

Figura 7: To make new truck.

```
114 to make-new-car1 [ freq x y h ]
115
116 if (random-float 100 < freq) and not any? turtles-on patch x y and spawned-cars1 < 50 and time > 15 [
117   create-cars 1 [
118     setxy x y
119     set heading h
120     set color one-of base-colors
121     set spawned-cars1 spawned-cars1 + 1
122     adjust-speed
123   ]
124 ]
125 end
126
127 to make-new-car2 [ freq x y h ]
128 if (random-float 100 < freq) and not any? turtles-on patch x y and spawned-cars2 < 70 and time > 75 [
129   create-cars 1 [
130     setxy x y
131     set heading h
132     set color one-of base-colors
133     set spawned-cars2 spawned-cars2 + 1
134     show spawned-cars2
135     adjust-speed
136   ]
137 ]
138 end
```

Figura 8: To make new car.

```
140 to make-new-car3 [ freq x y h ]
141 if (random-float 100 < freq) and not any? turtles-on patch x y and spawned-cars3 < 75 and time > 100 [
142   create-cars 1 [
143     setxy x y
144     set heading h
145     set color one-of base-colors
146     set spawned-cars3 spawned-cars3 + 1
147     adjust-speed
148   ]
149 ]
150 end
151
152 to make-new-car4 [ freq x y h ]
153 if (random-float 100 < freq) and not any? turtles-on patch x y and spawned-cars4 < 80 and time > 120 [
154   create-cars 1 [
155     setxy x y
156     set heading h
157     set color one-of base-colors
158     set spawned-cars4 spawned-cars4 + 1
159     adjust-speed
160   ]
161 ]
162 end
```

Figura 9: To make new car.

```
164 to make-new-car5 [ freq x y h ]
165 if (random-float 100 < freq) and not any? turtles-on patch x y and spawned-cars5 < 80 and time > 15 [
166   create-cars 1 [
167     setxy x y
168     set heading h
169     set color one-of base-colors
170     set spawned-cars5 spawned-cars5 + 1
171     adjust-speed
172   ]
173 ]
174 end
175
176 to make-new-car6 [ freq x y h ]
177 if (random-float 100 < freq) and not any? turtles-on patch x y and spawned-cars6 < 60 and time > 15 [
178   create-cars 1 [
179     setxy x y
180     set heading h
181     set color one-of base-colors
182     set spawned-cars6 spawned-cars6 + 1
183     adjust-speed
184   ]
185 ]
186 end
```

Figura 10: To make new car.

```
188 to make-new-car7 [ freq x y h ]
189 if (random-float 100 < freq) and not any? turtles-on patch x y and spawned-cars7 < 50 and time > 150 [
190   create-cars 1 [
191     setxy x y
192     set heading h
193     set color one-of base-colors
194     set spawned-cars7 spawned-cars7 + 1
195     adjust-speed
196   ]
197 ]
198 end
199
200 to make-new-car8 [ freq x y h ]
201 if (random-float 100 < freq) and not any? turtles-on patch x y and spawned-cars8 < 90 and time > 150 [
202   create-cars 1 [
203     setxy x y
204     set heading h
205     set color one-of base-colors
206     set spawned-cars8 spawned-cars8 + 1
207     adjust-speed
208   ]
209 ]
210 end
```

Figura 11: To make new car.

```
212 to move1
213   adjust-speed
214   repeat speed [
215     fd 1
216     if not can-move? 1 [ die ]
217     if [pxcor] of patch-ahead 1 = 14 [die]; die when I reach the end of the world
218   ]
219 end
220
221 to move2
222   adjust-speed
223   repeat speed [
224     fd 1
225     if not can-move? 1 [ die ]
226     if [pxcor] of patch-ahead 1 = -17 [die]; die when I reach the end of the world
227   ]
228 end
```

Figura 12: To move 1.

```
230 to move-truck
231 if speed = 0 [set time-stoped time-stoped + 1]
232 adjust-speed
233 repeat speed [
234   if not can-move? 1 [ die ]
235   ifelse [pxcor] of patch-here = 12 and [pycor] of patch-here = 13[
236
237     ask trucks [
238       face patch 12 3
239       fd 0.1
240     ]
241   ]
242   [fd 0.1]
243
244   ifelse [pxcor] of patch-here = 12 and [pycor] of patch-here = 3[
245     ask trucks [
246       face patch 2 3
247       fd 0.1
248     ]
249   ]
250   [fd 0.1]
251   ifelse [pxcor] of patch-here = 2 and [pycor] of patch-here = 3[
252     ask trucks [
253       face patch 2 -7
254       fd 0.1
255     ]
256   ]
257   [fd 0.1]
258   ifelse [pxcor] of patch-here = 2 and [pycor] of patch-here = -7[
259     ask trucks [
260       face patch 12 -7
261       fd 0.1
262     ]
263   ]
264 ]
```

```

264 [fd 0.1]
265   ifelse [pxcor] of patch-here = 12 and [pycor] of patch-here = -7[
266     ask trucks [
267       face patch 12 -17
268       fd 0.1
269     ]
270   ]
271 [fd 0.1]
272   ifelse [pxcor] of patch-here = 12 and [pycor] of patch-here = -17[
273     ask trucks [
274       face patch -8 -17
275       fd 0.1
276     ]
277   ]
278 [fd 0.1]
279   ifelse [pxcor] of patch-here = -8 and [pycor] of patch-here = -17[
280     ask trucks [
281       face patch -8 13
282       fd 0.1
283     ]
284   ]
285 [fd 0.1]
286   if [pxcor] of patch-here = -8 and [pycor] of patch-here = 13[
287     set time-finish ticks
288     set truck-died? true
289   ]
290 ]
291 ]
292 ]
293 end

```

Figuras 13 e 14: To move truck.

```

293 to adjust-speed
294
295   ; Calcula o mínimo e máximo de speed
296   let min-speed max (list (speed - 4) 0)
297   let max-speed min (list (speed + 2) 5)
298
299   ; Set para ir o mais rápido possível
300   let target-speed max-speed
301
302   let blocked-patch next-blocked-patch
303   if blocked-patch != nobody [
304     ; Caso tenha obstáculo, reduz o speed
305     ; até que não tenha mais nada
306     let space-ahead (distance blocked-patch - 1)
307     while [
308       breaking-distance-at target-speed > space-ahead and
309       target-speed > min-speed
310     ] [
311       set target-speed (target-speed - 1)
312     ]
313   ]
314
315   set speed target-speed
316
317 end

```

Figura 16. To adjust speed.

```

319 to-report breaking-distance-at [ speed-at-this-tick ]
320
321   ; Caso freie no próximo tick, calcula a distância
322   let min-speed-at-next-tick max (list (speed-at-this-tick - 4) 0)
323   report speed-at-this-tick + min-speed-at-next-tick
324 end
325
326 to-report next-blocked-patch
327
328   ; Verifica todos os patches a frente até encontrar um bloqueado
329   let patch-to-check patch-ahead 1
330   while [ patch-to-check != nobody and not is-blocked? patch-to-check ]
331     set patch-to-check patch-ahead ((distance patch-to-check) + 1)
332   ]
333
334   ; Reporta o patch bloqueado ou nada caso não esteja
335   report patch-to-check
336 end
337
338 to-report is-blocked? [ target-patch ]
339   report
340     any? other cars-on target-patch or
341     any? other trucks-on target-patch
342 end

```

Figura 15: To report breaking distance, to report next blocked patch e to report if it is blocked.

```

348 to-report show-time-start
349   report time-start
350 end
351
352 to-report show-time-finish
353   report time-finish
354 end
355
356 to-report show-time-stoped
357   report time-stoped
358 end

```

Figura 17. To report.

Experimentos Computacionais

Os experimentos computacionais consistiram na execução de 10 coletas de lixo. Ao final de cada coleta, foram registrados os dados e foram feitas as médias.

EXECUÇÃO	PARTIDA	TOTAL TIME	TEMPO PARADO
1º	0	2.66666	0
2º	0	2.66666	0
3º	0	2.66666	0
4º	0	2.26666	0
5º	0	2.4	0
6º	0	2.26666	0
7º	0	2.4	0
8º	0	2.26666	0
9º	0	2.4	0
10º	0	2.2666	0

Figura 18. Foram feitas 10 execuções. Partida: 0. Total time (média): 2,426666. Tempo parado (média): 0.

EXECUÇÃO	PARTIDA	TOTAL TIME	TEMPO PARADO
1º	15	3.133333	0.533333
2º	15	3.53333	0.8
3º	15	3.33333	0.66666
4º	15	2.933333	0.333333
5º	15	3.4	0.6
6º	15	4	1.066666
7º	15	3.4	0.6
8º	15	3.6	0.8
9º	15	3.33333	0.6
10º	15	3.533333	0.733333

Figura 19. Foram feitas 10 execuções. Partida: 15. Total time (média): 3.4199. Tempo parado (média): 0,67333.

EXECUÇÃO	TOTAL TIME	TEMPO PARADO
1º	4,8	1,53
2º	3,73	0,86
3º	4,6	1,46
4º	3,46	0,73
5º	4	1,06
6º	3,2	0,33
7º	4,2	1
8º	3,66	0,8
9º	2,93	0,13
10º	3,8	0,93

Figura. Foram feitas 10 execuções. Partida: 30. Total time (média): 3,765. Tempo parado (média): 0,895.

EXECUÇÃO	TOTAL TIME	TEMPO PARADO
1º	4,46	1,46
2º	3,86	0,66
3º	3,93	1
4º	4,66	1,46
5º	4,93	1,66
6º	3,4	0,46
7º	4,73	1,4
8º	4,66	1,2
9º	4,2	1,13
10º	4	1

Figura. Foram feitas 10 execuções. Partida: 45. Total time (média): 4,33. Tempo parado (média): 1,165.

EXECUÇÃO	TOTAL TIME	TEMPO PARADO
1°	5	1,46
2°	4,26	0,8
3°	4,4	1,33
4°	4,26	0,93
5°	4,86	1,66
6°	4,73	1,26
7°	4,06	0,93
8°	6	2,6
9°	5,2	1,8
10°	5,73	2,26

Foram feitas 10 execuções. Partida: 60. Total time (média): 4,795. Tempo parado (média): 1,395.

EXECUÇÃO	TOTAL TIME	TEMPO PARADO
1°	8,66	4,73
2°	7,2	3,86
3°	7,4	3,46
4°	6,86	3,33
5°	10,93	7,06
6°	6,46	3
7°	8,13	4,46
8°	7,13	3,6
9°	8,33	4,2
10°	7,73	4,2

Figura. Foram feitas 10 execuções. Partida: 90. Total time (média): 7,565. Tempo parado (média): 4,03.

EXECUÇÃO	PARTIDA	CHEGADA	TOTAL TIME	TEMPO PARADO
1°	76	142	4.4	0.6
2°	76	147	6.06666	2.33
3°	76	174	6.53333	2.8
4°	76	173	6.46666	3.2
5°	76	135	3.93333	0.66
6°	76	170	6.26666	2.66
7°	76	149	4.86666	1.13
8°	76	250	11.6	7.53
9°	76	179	6.86666	2.8
10°	76	201	8.3333	4.13

Figura 18. Foram feitas 10 execuções. Partida: 75. Total time (média): 6,533. Tempo parado (média): 2,78666.

EXECUÇÃO	TOTAL TIME	TEMPO PARADO
1°	7,04	3,66
2°	8,4	4,53
3°	7,73	4
4°	5,46	1,86
5°	7,73	3,73
6°	7,06	3,53
7°	6,93	3
8°	8	4,13
9°	6,26	2,53
10°	7,33	3,33

Figura. Foram feitas 10 execuções. Partida: 105. Total time (média): 7,195. Tempo parado (media): 3,595.

EXECUÇÃO	TOTAL TIME	TEMPO PARADO
1°	5,86	2,13
2°	6,66	3,06
3°	7,2	3,4
4°	6,06	2,73
5°	5,4	1,73
6°	6,13	2,26
7°	6,46	2,53
8°	6,73	2,86
9°	7,93	4,4
10°	7	2,93

Figura. Foram feitas 10 execuções. Partida: 120. Total time (média): 6,56. Tempo parado (média): 2,795.

EXECUÇÃO	TOTAL TIME	TEMPO PARADO
1°	5.6	1.86
2°	6,53	3,33
3°	6,2	2,86
4°	6,73	2,93
5°	5,66	1,93
6°	5,2	2,13
7°	5,46	1,8
8°	6,2	2,8
9°	6,73	3,13
10°	5,53	2

Figura. Foram feitas 10 execuções. Partida: 135. Total time (média): 5,93. Tempo parado (media): 2,465.

EXECUÇÃO	TOTAL TIME	TEMPO PARADO
1°	5.33	1.6
2°	3.93	0.733
3°	4,6	1
4°	5	1,6
5°	4,73	1,06
6°	4,66	1,2
7°	5,13	1,26
8°	5,46	2,2
9°	6	2,1
10°	4,73	1,06

Figura. Foram feitas 10 execuções. Partida: 150. Total time (média): 4,865. Tempo parado (média): 1,23.

Considerações Finais

Com o avanço da área da computação, foram desenvolvidos diversos ambientes voltados a realizar simulações de agentes que visam fazer estudos de determinados comportamentos. Um deles é o NetLogo, que permite ao usuário explorar suas ferramentas afim de desenvolver simulações que visam estudar comportamentos de sistemas complexos. No nosso estudo em específico, a ferramenta foi utilizada para a execução de coleta de lixo. Foram feitas 10 execuções em cada um dos 11 cenários e chegou-se à conclusão de que o melhor cenário é o primeiro, em que a partida é 0.

REFERÊNCIAS

<https://www.netlogoweb.org/launch#Load> (acesso em: 28/07/2022)
https://eventos.unipampa.edu.br/eremat/files/2014/12/MC_FONSECA_00529108062.pdf (acesso em: 28/07/2022)