Introdução ao Aprendizado por Reforço

Prof. Me. André Luiz Carvalho Ottoni

Aprendizado por Reforço: Engenharia e Estatística impulsionando a sociedade
Projeto financiado pelo Edital 001/2019/UFSJ/Reitoria

Centro de Ciências Exatas e Tecnológicas
UFRB - Universidade Federal do Reconcâvo da Bahia

São João del-Rei (MG), outubro de 2019

Plano de Aula

1. Navegação Autônoma - Grid World
2. Problema do Caixeiro Viajante
3. Problema da Mochila Multidimensional
4. Sequential Ordering Problem
5. Futebol de Robôs

Simulador[1]

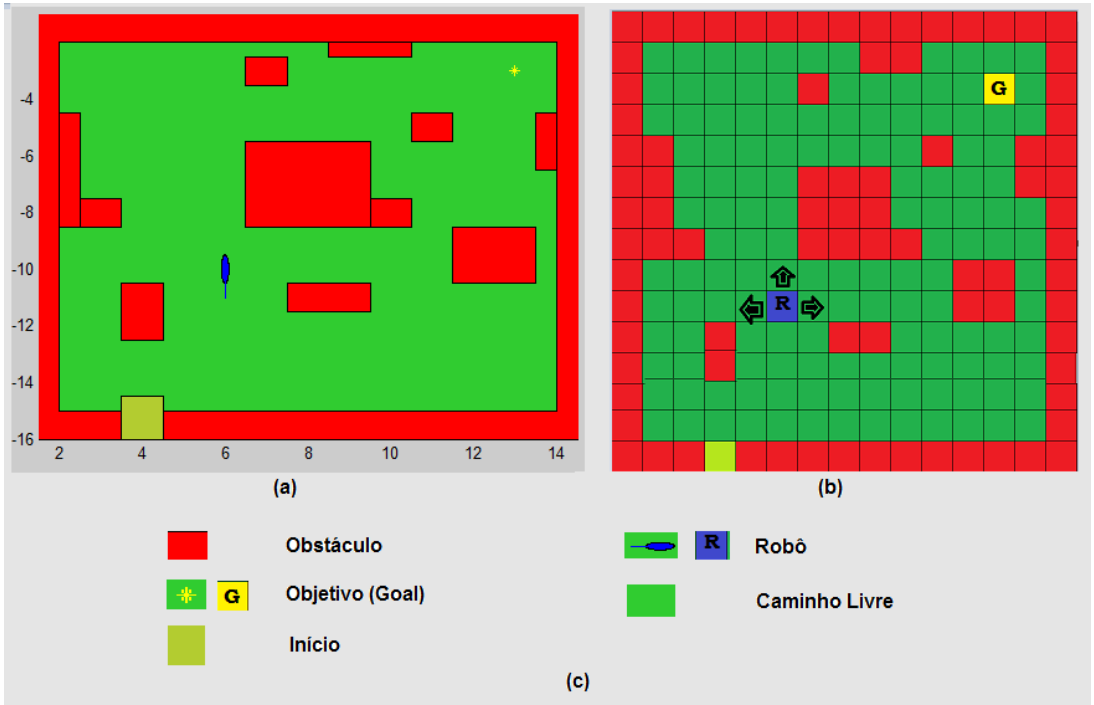


Figura 1: Simulador Desenvolvido.

[1] Ottoni et al (2016). Análise da influência da taxa de aprendizado e do fator de desconto sobre o desempenho dos algoritmos Q-learning e SARSA: aplicação do aprendizado por reforço na navegação autônoma. Revista Brasileira de Computação Aplicada

Definição das ações

1. Mover para o norte: o agente desloca uma célula para cima no eixo y.
2. Mover para o oeste: o agente desloca uma célula para a esquerda no eixo x.
3. Mover para o leste: o agente desloca uma célula para a direita no eixo x.

Navegação Autônoma - Grid World

Definição dos estados - Características

- Caminho livre: o agente possui livre acesso e recebe um retorno pouco negativo ao acessá-lo.
- Obstáculo: o agente possui acesso bloqueado e recebe um retorno muito negativo ao acessá-lo.
- Objetivo: representa o final da trajetória. Ao chegar, o agente recebe um retorno muito positivo.

Definição dos estados

1. Caminho livre somente no norte.
2. Caminho livre somente no oeste.
3. Caminho livre somente no leste.
4. Caminho livre somente no norte e oeste.
5. Caminho livre somente no norte e leste.
6. Caminho livre somente no oeste e leste.
7. Caminho livre no norte, oeste e leste.
8. Objetivo no norte.
9. Objetivo no oeste.
10. Objetivo no leste.

Navegação Autônoma - Grid World

Definição das recompensas

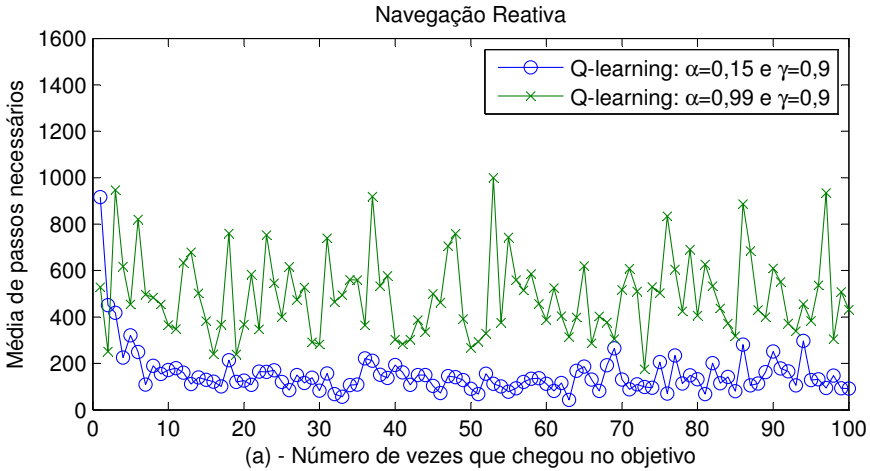
- -1: movimentar por um caminho livre;
- -100: bater em um obstáculo.
- -100: escolher uma ação que não leve ao objetivo, quando esse se encontra a um passo.
- 1000: chegar ao objetivo.

Definição das recompensas

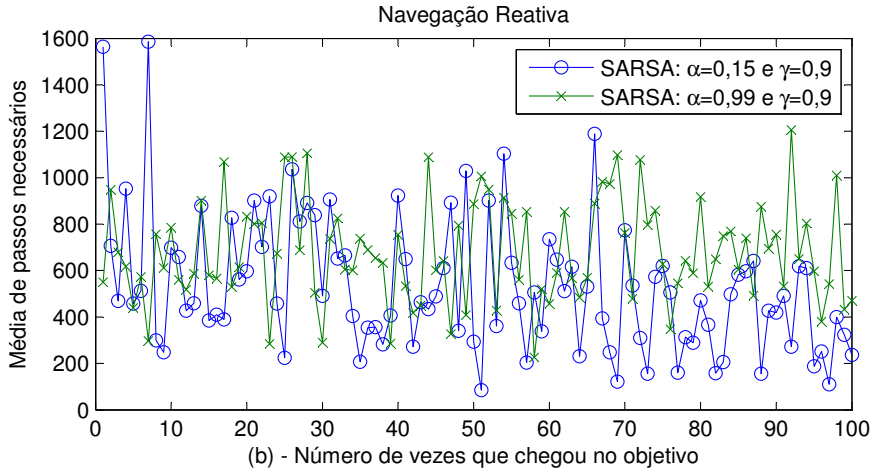
Tabela 1: Valores de reforços para cada para $S \times A$

Estado/Ação	Norte	Oeste	Leste
1	-1	-100	-100
2	-100	-1	-100
3	-100	-100	-1
4	-1	-1	-100
5	-1	-100	-1
6	-100	-1	-1
7	-1	-1	-1
8	1000	-100	-100
9	-100	1000	-100
10	-100	-100	1000

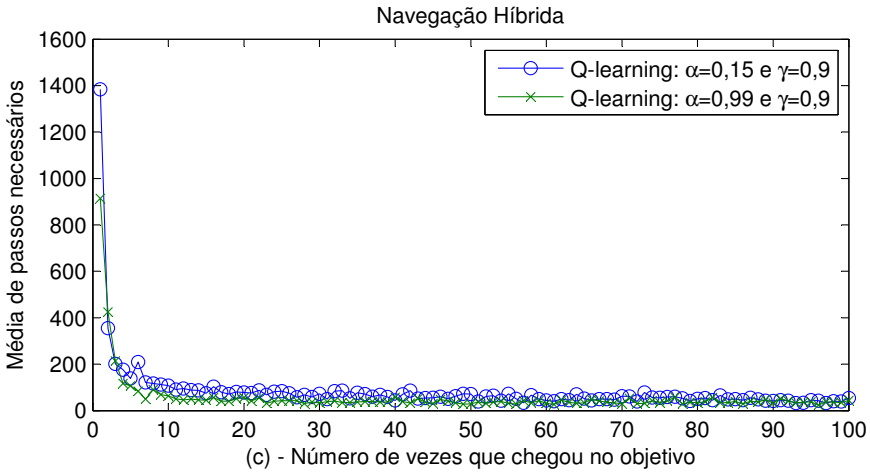
Resultados



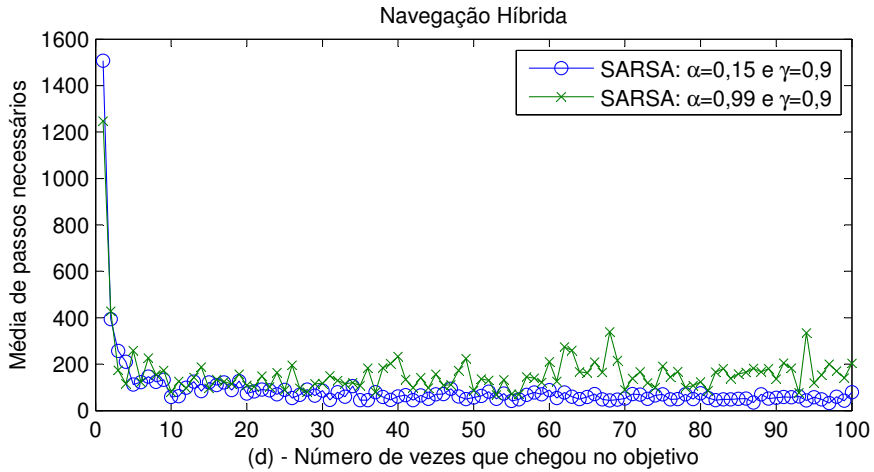
(a) - Número de vezes que chegou no objetivo



(b) - Número de vezes que chegou no objetivo



(c) - Número de vezes que chegou no objetivo



(d) - Número de vezes que chegou no objetivo

Problema do Caixeiro Viajante

- Objetivo: Definir o menor caminho entre o conjunto de n localidades.
- Aplicações[2]:
  - planejamento de rotas,
  - confecção de placas de circuitos impresso,
  - mapeamento de genoma e sequenciamento de DNA,
  - problemas de sequenciamento de tarefas.
- Alguns métodos[3]:
  - Algoritmos de Colônia de Formigas,
  - Algoritmos Genéticos,
  - Busca Tabu,
  - Redes Neurais.

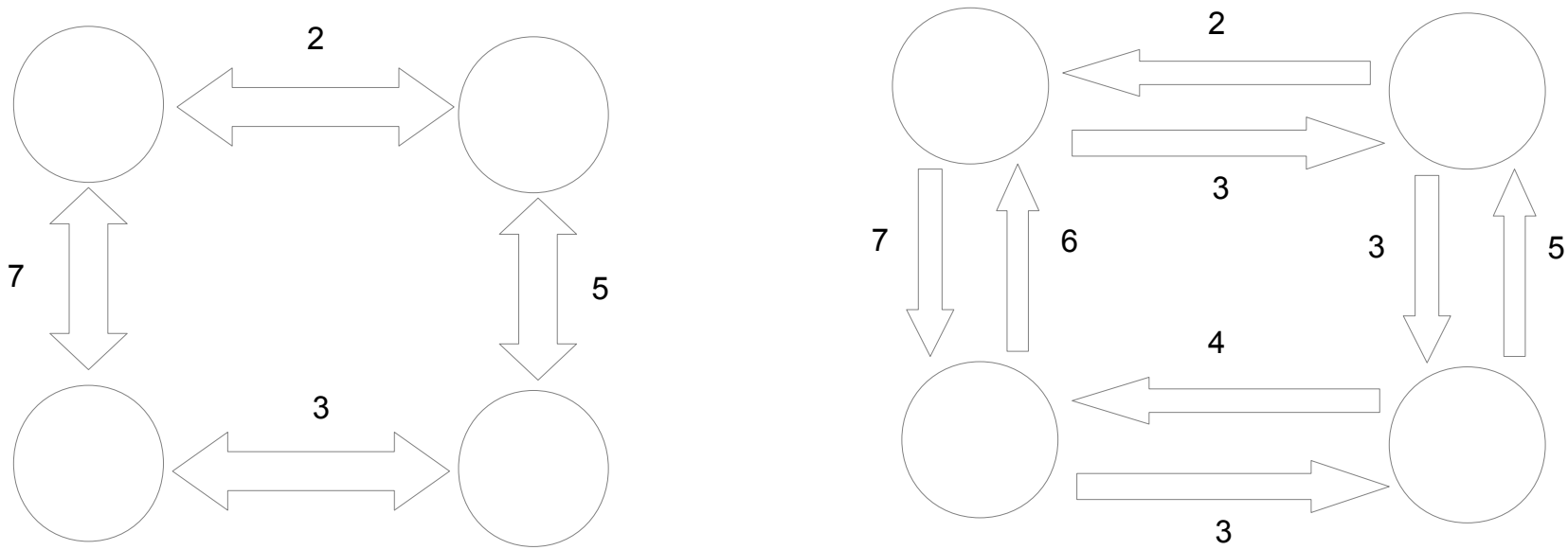
---

[2] Reinelt, G. (1994). The Traveling Salesman: Computational Solutions for TSP Applications. Springer-Verlag, Berlin, Heidelberg.

[3] Applegate, D., Bixby, R. E., Chvátal, V., e Cook, W. (2007). The Traveling Salesman Problem: A Computational Study. Princeton University Press Princeton.

Problema do Caixeiro Viajante

- PCV Simétrico: $c_{ij} = c_{ji}$,
- PCV Assimétrico: $c_{ij} \neq c_{ji}$,



(a) PCV Simétrico (TSP). (b) PCV Assimétrico (ATSP).

Figura 3: Exemplos de caminhos para o PCV.

Formulação Matemática:

$$Min \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} x_{ij}, \tag{1}$$

sujeito a:

$$\sum_{i=1}^{N} x_{ij} = 1 \qquad j = 1, \ldots, N, \tag{2}$$

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad i = 1, \ldots, N, \tag{3}$$

$$x_{ij} \in \{0, 1\} \qquad i,j = 1, \ldots, N, \tag{4}$$

$$X = x_{ij} \in S \qquad i,j = 1, \ldots, N, \tag{5}$$

Problema do Caixeiro Viajante

Formulação Matemática:

$$Min \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} x_{ij}, \tag{6}$$

- Função objetivo: minimizar o custo - distância percorrida.
- N: um conjunto de nós.
- c_{ij}: custo de deslocamento entre duas cidades (i e j).
- $x_{i,j}$:
  - Varíavel de decisão.
  - Assume 1 se o arco (i,j) compor a solução e 0 caso contrário.

Formulação Matemática:

Restrições:

$$\sum_{i=1}^{N} x_{ij} = 1 \qquad j = 1,\ldots,N, \tag{7}$$

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad i = 1,\ldots,N, \tag{8}$$

- Asseguram que cada localidade será visitada uma única vez.

$$x_{ij} \in \{0,1\} \qquad i,j = 1,\ldots,N, \tag{9}$$

- Garante que a variável x_{ij} é binária

$$X = x_{ij} \in S \qquad i,j = 1,\ldots,N, \tag{10}$$

- O conjunto S representa qualquer grupo de restrições que eliminem a formação de sub-rotas.

Modelo de AR[4]:

1. Estados (S): conjunto de todas as localidades.
2. Ações (A): intenção de ir para outra localidade (estado).
3. Reforços: distâncias entre as localidades multiplicada por -1.

$$r_{ij} = -d_{ij}, \tag{11}$$

- i e j: localidades.
- d_{ij}: distância entre i e j.
- r_{ij}: reforço recebido por partir de i para j.

Aplicação: AR no simulador em MATLAB.

[4]Ottoni A. L. C. et al. (2018). A Response Surface Model Approach to Parameter Estimation of Reinforcement Learning for the Travelling Salesman Problem. Journal of Control, Automation and Electrical Systems.

Modelo de AR - Exemplo de representação

$$D = \begin{bmatrix} \cdot & 2 & 7 & 10 \\ 2 & \cdot & 4 & 5 \\ 7 & 4 & \cdot & 3 \\ 10 & 5 & 3 & \cdot \end{bmatrix}, \quad R = \begin{bmatrix} \cdot & -2 & -7 & -10 \\ -2 & \cdot & -4 & -5 \\ -7 & -4 & \cdot & -3 \\ -10 & -5 & -3 & \cdot \end{bmatrix} \tag{12}$$

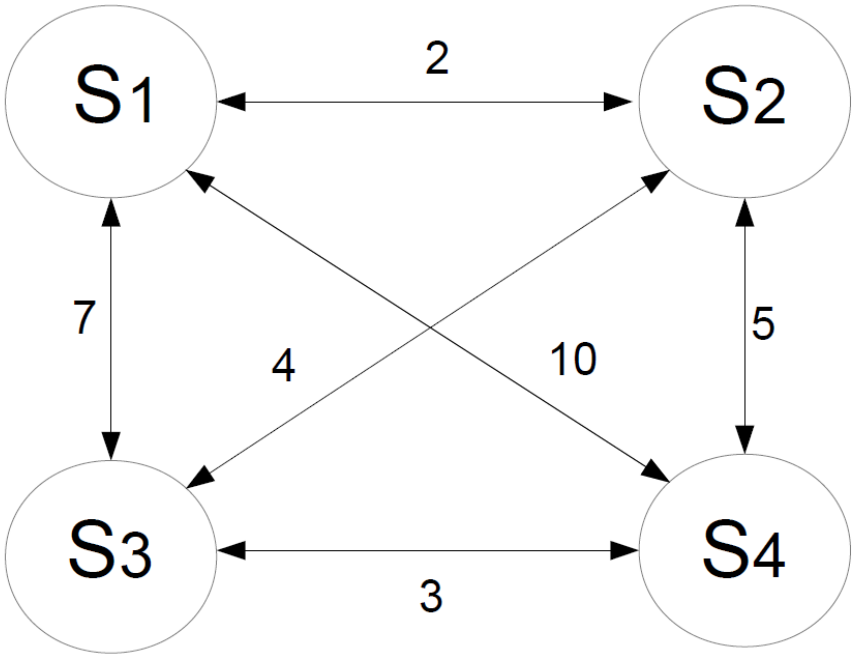


Figura 4: Exemplo de representação com 4 localidades.

Experimentos

- TSPLIB[5]: biblioteca de instâncias do PCV.
- Link: http://elib.zib.de/pub/mp-testdata/tsp/tsplib/

Tabela 2: Exemplos de Problemas da TSPLIB.

Tipo	Problema	Cidades (n)	Solução Ótima
Simétrico	berlin52	52	7542
	brazil58	58	25395
	kroA100	100	21282
	kroA200	200	29368
Assimétrico	br17	17	39
	ftv33	34	1286
	ftv44	45	1613
	ftv64	65	1839

[5]Reinelt, G. (1991). Tsplib - a traveling salesman problem library. ORSA Journal on Computing 3(4), 376-384.

TSPLIB - Exemplo 1

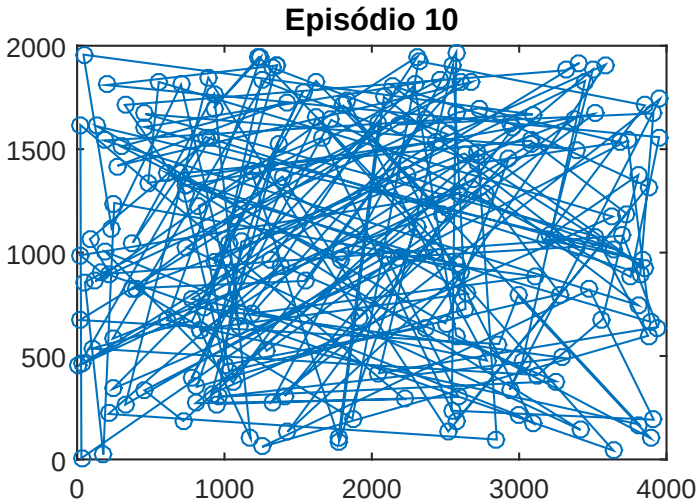
```
NAME: berlin52
TYPE: TSP
COMMENT: 52 locations in Berlin (Groetschel)
DIMENSION: 52
EDGE_WEIGHT_TYPE: EUC_2D
NODE_COORD_SECTION
1 565.0 575.0
2 25.0 185.0
3 345.0 750.0
4 945.0 685.0
5 845.0 655.0
6 880.0 660.0
7 25.0 230.0
8 525.0 1000.0
9 580.0 1175.0
10 650.0 1130.0
11 1605.0 620.0
12 1220.0 580.0
13 1465.0 200.0
14 1530.0 5.0
15 845.0 680.0
16 725.0 370.0
17 145.0 665.0
18 415.0 635.0
19 510.0 875.0
```

TSPLIB - Exemplo 2

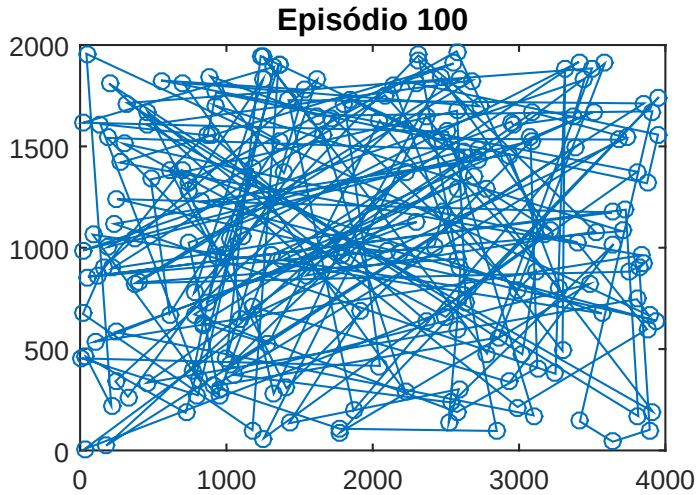
```
NAME:  br17
TYPE: ATSP
COMMENT: 17 city problem (Repetto)
DIMENSION:  17
EDGE_WEIGHT_TYPE: EXPLICIT
EDGE_WEIGHT_FORMAT: FULL_MATRIX
EDGE_WEIGHT_SECTION
 9999    3    5   48   48    8    8    5    5    3    3    0    3    5    8    8
    5
    3 9999    3   48   48    8    8    5    5    0    0    3    0    3    8    8
    5
    5    3 9999   72   72   48   48   24   24    3    3    5    3    0   48   48
   24
   48   48   74 9999    0    6    6   12   12   48   48   48   48   74    6    6
   12
   48   48   74    0 9999    6    6   12   12   48   48   48   48   74    6    6
   12
    8    8   50    6    6 9999    0    8    8    8    8    8    8   50    0    0
    8
    8    8   50    6    6    0 9999    8    8    8    8    8    8   50    0    0
    8
    5    5   26   12   12    8    8 9999    0    5    5    5    5   26    8    8
    0
    5    5   26   12   12    8    8    0 9999    5    5    5    5   26    8    8
    0
    3    0    3   48   48    8    8    5    5 9999    0    3    0    3    8    8
    5
    3    0    3   48   48    8    8    5    5    0 9999    3    0    3    8    8
    5
```

Planejamento de Rotas

- Exemplos de rotas para kroA200, com SARSA , $\alpha = 0{,}7894$, $\gamma = 0{,}0894$ e $\varepsilon = 0{,}01$:



(a) Episódio 10: *331.527*

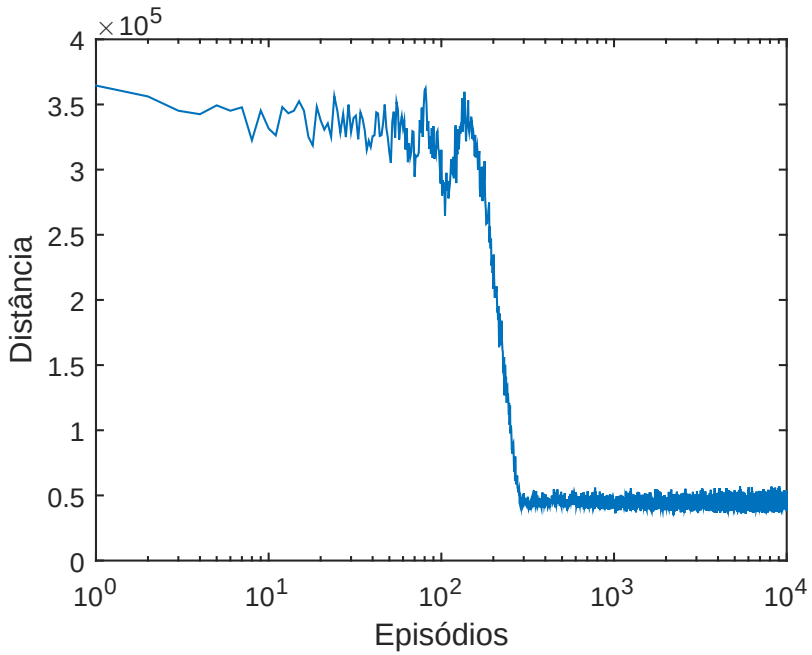


(b) Episódio 100: *315.239*

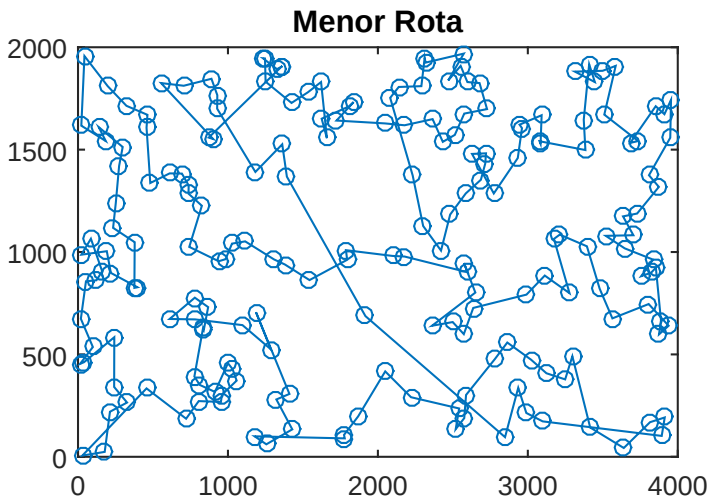
Figura 7: Representação de rotas encontrados para kroA200.

Planejamento de Rotas

- Exemplos de rotas para kroA200, com SARSA , $\alpha = 0{,}7894$, $\gamma = 0{,}0894$ e $\varepsilon = 0{,}01$:



(a) Episódios x Distância



(b) Episódio 2206: *34.795*

Figura 8: Representação de rotas encontrados para kroA200.

Planejamento de Rotas

Planejamento de Rotas com Reabastecimento[6]:

- Planejamento de rotas em uma malha rodoviária para veículos autônomos.
- O agente móvel deve percorrer um conjunto de cidades.
- Decidir os locais de reabastecimento de combustível.
- **Objetivo**: minimizar o gasto final na rota.
- Custo não-uniforme: o preço de combustível varia entre as cidades (ANP).

[6]Ottoni A. L. C. et al. (2019a). Estimação de Parâmetros do Aprendizado por Reforço para o Problema de Planejamento de Rotas com Reabastecimento. Simpósio Brasileiro de Automação Inteligente 2019

Modelo de AR:

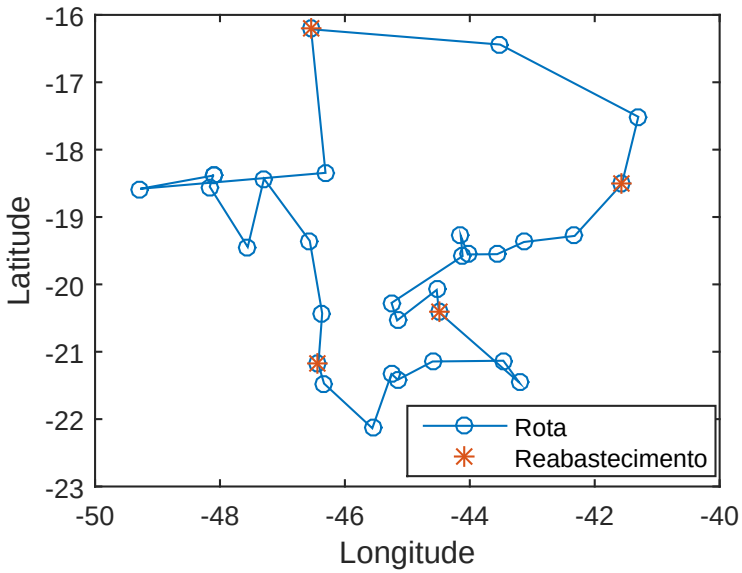
1. Estados (S): conjunto de todas as localidades.
2. Ações (A):
   - intenção de ir para outra localidade (estado).
   - Reabastecimento: é realizada sempre que o veéculo chegue em uma localidade com o nível do tanque menor do que 25% da capacidade total.
3. Reforços: associa o custo com movimentação e o gasto com reabastecimento.

$$r_{ij} = -(d_{ij} + c_j), \tag{13}$$

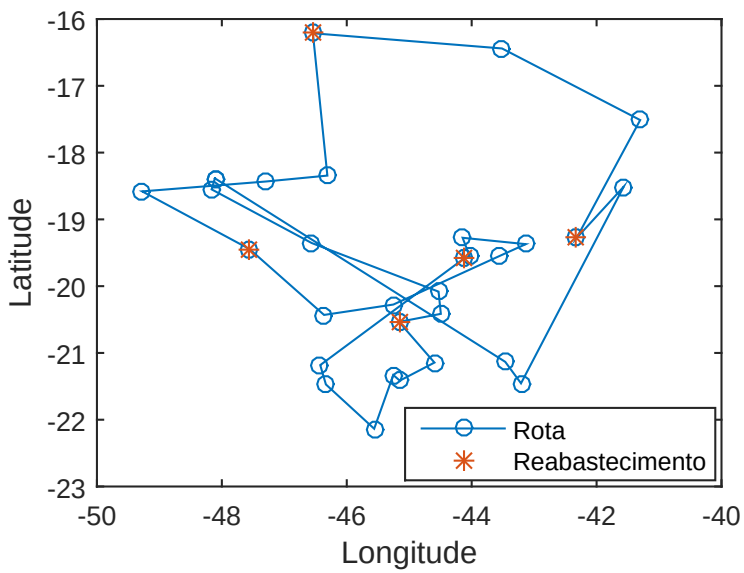
- i e j: localidades.
- d_{ij}: distância entre i e j.
- c_j: custo de reabastecimento na cidade j.
- r_{ij}: reforço recebido por partir de i para j.

Planejamento de Rotas

- Exemplos de rotas com reabastecimento:



(a)Gasto: R$ 1606,64 .
Distância: 3957,64 km



(b)Gasto: R$ 2178,35.
Distância: 5581,43 km

Figura 9: Representação de rotas encontradas.

Problema da Mochila Multidimensional

- Objetivo: Selecionar um conjunto de itens que maximizem o transporte desses objetos em m mochilas
- Deve ser respeita a restrição de capacidade para cada mochila.
- Diferença do problema unidimensional: apenas uma mochila.
- Aplicações:
  - Otimização de corte de materiais.
  - Problemas de embalagem.
  - Carregamento de veículos.

Problema da Mochila Multidimensional

Formulação Matemática[7]

$$Max \sum_{i=1}^{n} v_i x_i, \tag{14}$$

sujeito a:

$$\sum_{i=1}^{n} p_{ji} x_i \leq c_j \qquad j = 1, \ldots, m, \tag{15}$$

$$x_i \in \{0, 1\} \qquad i = 1, \ldots, n, \tag{16}$$

[7] P. C. Chu and J. E. Beasley, "A genetic algorithm for the multidimensional knapsack problem," Journal of heuristics, vol. 4, no. 1, pp. 63–86,1998.

Problema da Mochila Multidimensional

Formulação Matemática

$$Max \sum_{i=1}^{n} v_i x_i, \tag{17}$$

- Função objetivo: maximizar o somatórios dos valores itens transportados.
- x_i: armazena a decisão se um objeto i foi selecionado (1 = 'Sim' ou 0 = 'Não').
- v_i: valor do item i;
- n: número total de itens disponíveis.
- m: número de mochilas.
- i: índice referente ao item.

Problema da Mochila Multidimensional

Formulação Matemática

Restrições:

$$\sum_{i=1}^{n} p_{ji} x_i \leq c_j \qquad j = 1, \ldots, m, \tag{18}$$

- Respeita a restrição de capacidade para cada mochila.
- j: índice referente à mochila.
- p_{ji}: peso do item i referente à mochila j.
- c_j: capacidade máxima da mochila j.

$$x_i \in \{0, 1\} \qquad i = 1, \ldots, n, \tag{19}$$

- Garante que a variável de decisão é binária.

Problema da Mochila Multidimensional

Exemplo de Modelo de AR

1. Estados (S): um estado (s_t) identifica qual objeto é inserido na mochila no instante t.
2. Ações (A):
   - uma ação (a_t) no instante t representa a intenção de inserir um determinado item na mochila em $t+1$.
   - ao colocar um objeto na mochila, o item é retirado da lista de ações disponíveis.
3. Reforços:
   - as recompensas buscam valorizar os objetos mais importantes da lista de itens.
   - penalizar a inserção de itens com altos valores de restrição.

$$r(s,a) = v_i - p_{ji}, \qquad j = 1,\ldots,m. \tag{20}$$

- v_i: valor do item i;
- p_{ji}: peso do item i referente à mochila j.

Experimentos

- ORLIB[8]
- Link: `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/mknap2.txt`

Tabela 3: Exemplos de Problemas da ORLIB.

	Problema	m	n	**Ótimo**
1	sento1	30	60	7772
2	weing1	2	28	141278
3	weing2	2	28	130883
4	weing3	2	28	95677
5	weing7	2	105	1095445
6	weish1	5	30	4554
7	weish2	5	30	4536
8	weish30	5	90	11191
9	hp1	4	28	3418

ORLIB - Exemplo 1

```
problem WEING1.DAT

++++++++++++++++++++++++++++++
2 28
1898 440 22507 270 14148 3100 4650 30800 615 4975
1160 4225 510 11880 479 440 490 330 110 560
24355 2885 11748 4550 750 3720 1950 10500
600 600
45 0 85 150 65 95 30 0 170 0
40 25 20 0 0 25 0 0 25 0
165 0 85 0 0 0 0 100
30 20 125 5 80 25 35 73 12 15
15 40 5 10 10 12 10 9 0 20
60 40 50 36 49 40 19 150

141278
```

Figura 10: Exemplo de instância da ORLIB - weing1.

Problema da Mochila Multidimensional

ORLIB - Exemplo 1 - Matlab

```
function [n,m,v,c,p,otimo,nome]=MKPweing1()
    otimo = 141278;
    nome = 'weing1';
    m = 2; %número de mochilas
    n = 28; %número de itens

    %Valor de cada objeto (variável da função objetivo)
    v = [1898 440 22507 270 14148 3100 4650 30800 615 4975
1160 4225 510 11880 479 440 490 330 110 560 24355 2885 11748
4550 750 3720 1950 10500];

    %Capdacidade de cada mochila (restrição)
    c = [600 600];

    %Peso de cada objeto em cada mochila (valores para
restrições)
    p= [45 0 85 150 65 95 30 0 170 0 40 25 20 0 0 25 0 0 25 0
165 0 85 0 0 0 0 100
        30 20 125 5 80 25 35 73 12 15 15 40 5 10 10 12 10 9 0
20 60 40 50 36 49 40 19 150];
end
```

Sequential Ordering Problem

- SOP: *Sequential Ordering Problem*.
- Baseado no Problema do Caixeiro Viajante Assimétrico (ATSP).
- PCV Assimétrico: $c_{ij} \neq c_{ji}$.
- Restrição de precedência: uma localidade j deve ser visitada antes de outra cidade i.
- $c_{ij} = -1$.

TSPLIB - Exemplo 1

```
NAME: ESC07.sop
TYPE: SOP
COMMENT: Received by Norbert Ascheuer / Laureano Escudero
DIMENSION: 9
EDGE_WEIGHT_TYPE: EXPLICIT
EDGE_WEIGHT_FORMAT: FULL_MATRIX
EDGE_WEIGHT_SECTION
9
    0    0    0    0    0    0    0    0 1000000
   -1    0  100  200   75    0  300  100    0
   -1  400    0  500  325  400  600    0    0
   -1  700  800    0  550  700  900  800    0
   -1   -1  250  225    0  275  525  250    0
   -1   -1  100  200   -1    0   -1   -1    0
   -1   -1 1100 1200 1075 1000    0 1100    0
   -1   -1    0  500  325  400  600    0    0
   -1   -1   -1   -1   -1   -1   -1   -1    0
EOF
```

Figura 12: Exemplo de instância da TSPLIB para o SOP - esc07.

Sequential Ordering Problem

Formulação Matemática:

$$Min \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} x_{ij}, \tag{21}$$

Sujeito à:

$$\sum_{i=1}^{N} x_{ij} = 1 \qquad (\forall j = 1,...,N), \tag{22}$$

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad (\forall i = 1,...,N), \tag{23}$$

$$x_{ij} \in \{0,1\} \qquad (\forall i,j = 1,...,N), \tag{24}$$

$$X = x_{ij} \in S \qquad (\forall i,j = 1,...,N), \tag{25}$$

$$c_{ij} \geq 0 \vee c_{ij} = -1 \wedge c_{ji} \geq 0 \qquad (\forall i,j = 1,...,N), \tag{26}$$

Restrição de precedência:

$$c_{ij} \geq 0 \vee c_{ij} = -1 \wedge c_{ji} \geq 0 \qquad (\forall i,j = 1,...,N), \tag{27}$$

- $c_{ij} \geq 0$: O custo entre duas localidades pode ser maior ou igual a zero, ou,
- $c_{ij} = -1$:
  - existe uma restrição de precedência.
  - O nó j precede o nó i.
  - Ou seja, a localide j deve ser acessada antes de i.
  - Então: $c_{ji} \geq 0$.

Sequential Ordering Problem

Exemplo de Modelo de AR:

1. Estados (S): conjunto de todas as localidades.
2. Ações (A): intenção de ir para outra localidade (estado).
3. Reforços:

$$R_1 = -d_{ij}, \tag{28}$$

$$R_2 = \frac{1}{d_{ij}}, \tag{29}$$

$$R_3 = -(d_{ij})^2. \tag{30}$$

Sequential Ordering Problem

RLSOP

```
1  a_t = ε-greedy()
2  cont = 0
3  while (cont == 0) do
4      if (there are precedence constraints for the selected action)
       then
5          if (at least one action corresponding to the precedence
           constraints of a_t has not yet been selected) then
6              cont = 0
7          else
8              cont = 1
9          end
10     end
11     if (cont==0) then
12         Remove action a_t from the list of available actions at the
           instant t;
13         a_t = ε-greedy()
14     end
15 end
16 Return a
```

Sequential Ordering Problem

Experimentos

- TSPLIB[9]: biblioteca de instâncias do PCV.
- Link: http://elib.zib.de/pub/mp-testdata/tsp/tsplib/

Tabela 4: Exemplos de Problemas da TSPLIB.

Problema	Nós	Restrições	Ótimo
br17.10	18	48	55
esc07	9	22	2125
esc12	14	36	1675
esc25	27	62	1681
esc47	49	127	1288
esc63	65	360	62
esc78	80	440	18230
ft53.1	54	117	7531
ft53.4	54	864	14425
prob42	42	100	243
rbg109a	111	5548	1038

[9] Reinelt, G. (1991). Tsplib - a traveling salesman problem library. ORSA Journal

Futebol de Robôs

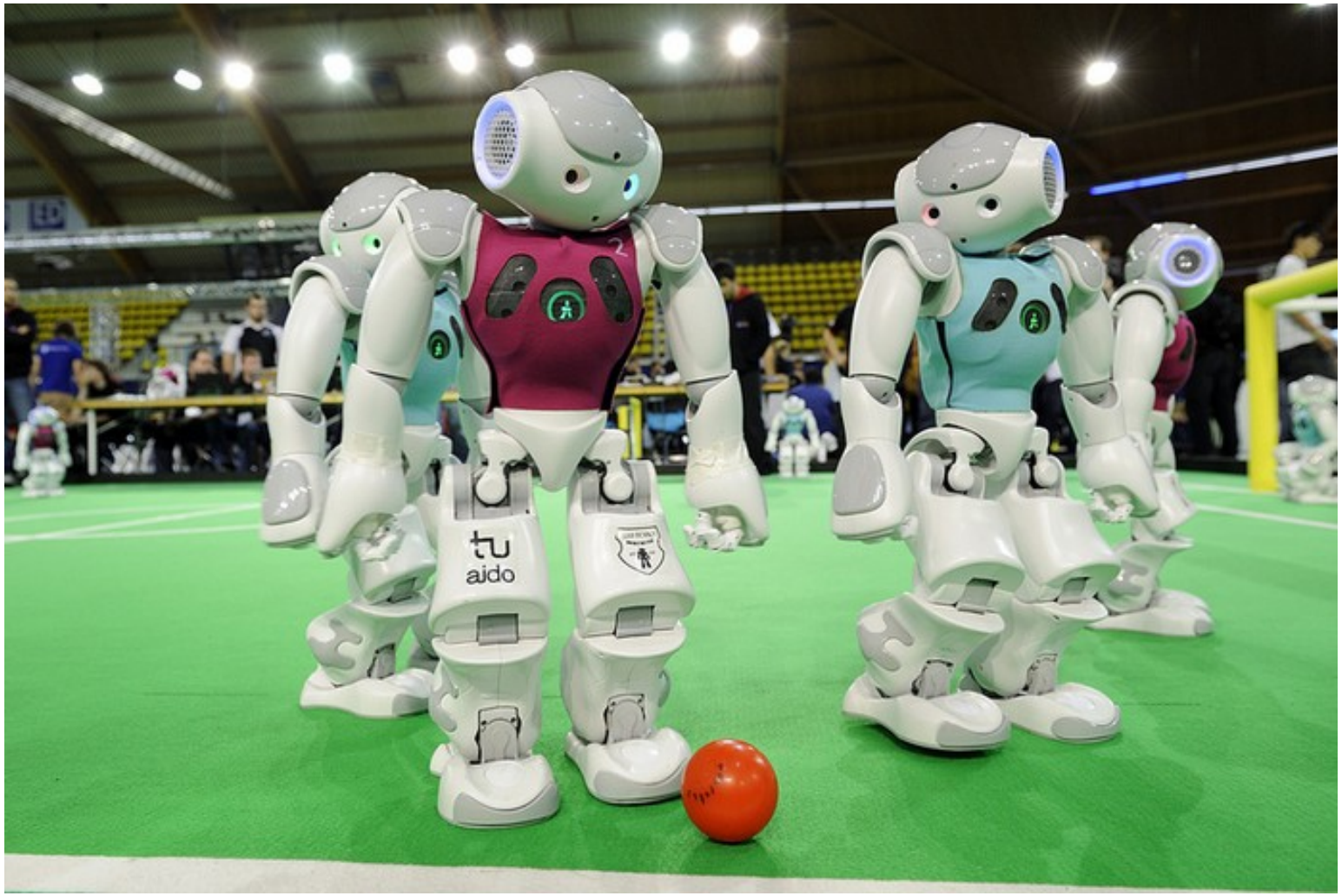


Figura 14: Futebol de Robôs.

Simulador da RoboCup

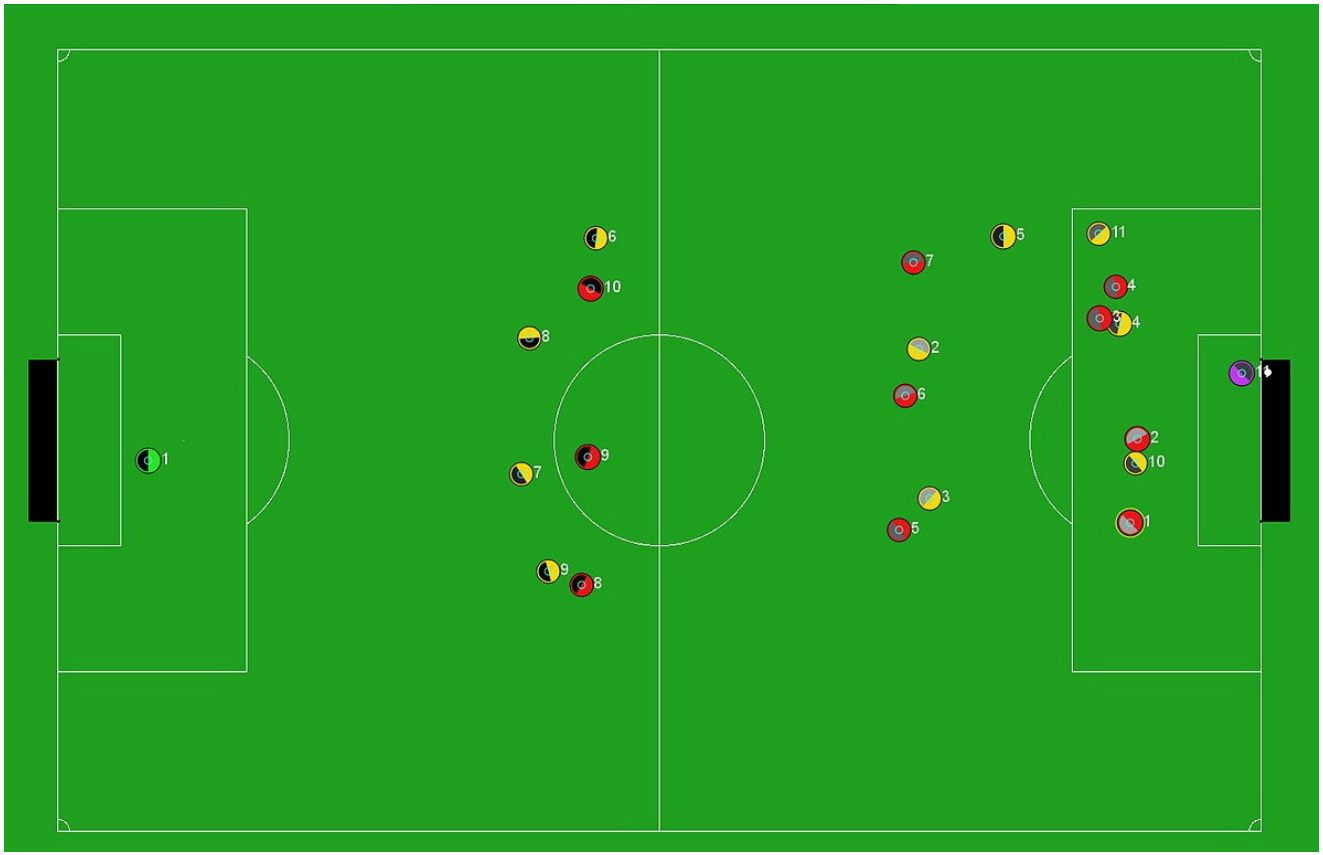


Figura 15: Simulador 2D da RoboCup.

Definição das Ações [10]

1. Drible A.
2. Drible B.
3. Passe A
4. Passe B.
5. Lançamento de bola.
6. Chute.

[10] Ottoni et al (2015). Análise do Aprendizado por Reforço via Modelos de Regressão Logística: Um Estudo de Caso no Futebol de Robôs. Revista Jr de Iniciação Científica em Ciências Exatas e Engenharia

Definição dos Estados

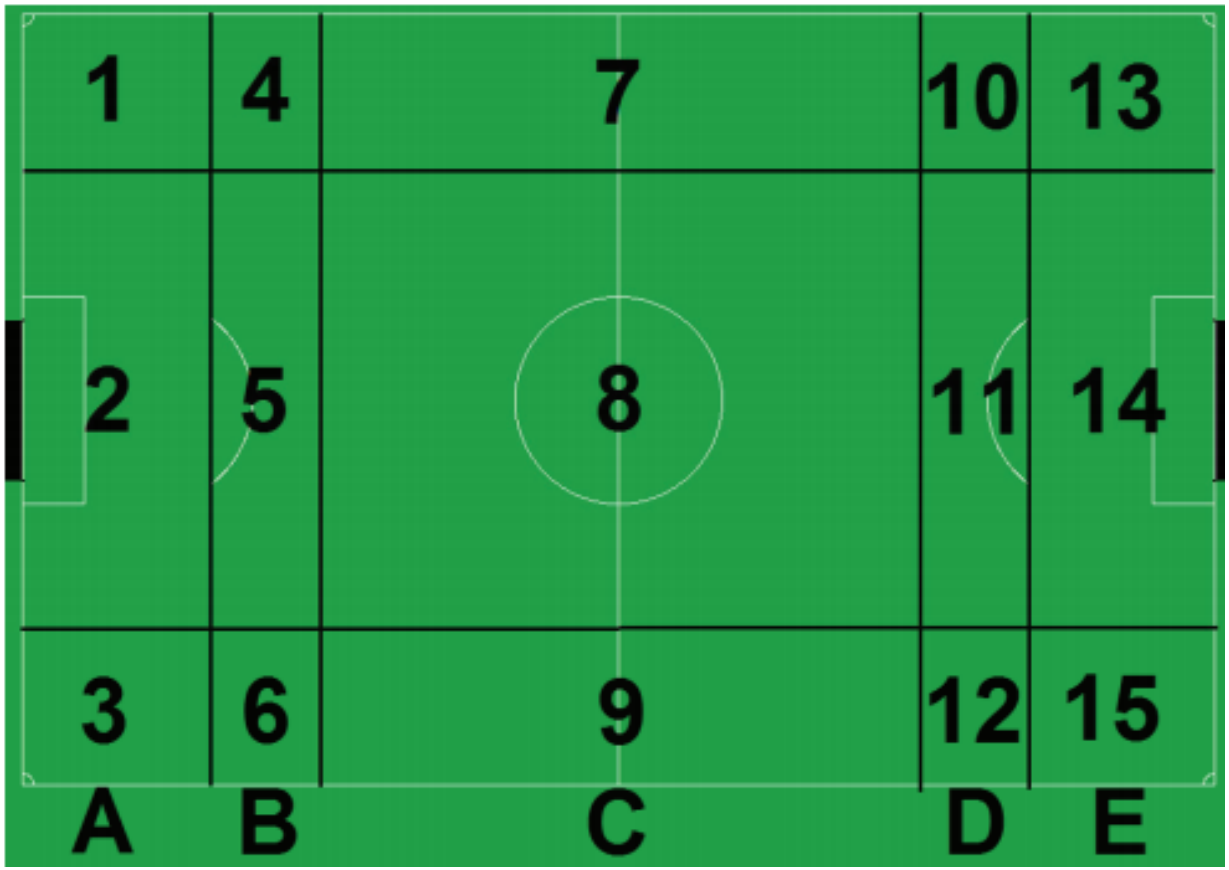


Figura 16: Zonas do Campo.

Definição das Recompensas

Zona	Penalidade	Reforço
A	-10	-1
B	-1	0
C	0	1
D	1	10
E	10	20
E (Célula14)	10	40

Figura 17: Recompensas por zona do campo.

Futebol de Robôs

- **Vídeo**: Aprendizado por Reforço no futebol de robôs.