



Aplicar o Aprendizado de Reforço para Resolver o Problema de Roteamento de Veículos Capacitados

Luis Augusto Lüders Meza^{1*}, Leonardo de Carvalho Vidal² e Helton Sereno³

¹Instituto Federal do Rio de Janeiro

²Instituto Federal do Rio de Janeiro

³Instituto Federal do Rio de Janeiro

*Email do autor principal: 1.luis.meza@gmail.com

Grupo Temático: Tecnologia e Inovação Tecnológica

Resumo: O Problema do Roteamento de Veículos Capacitados (PRVC) é um problema de otimização combinatória, que consiste em arranjar o conjunto de rotas ótimas para determinada frota de veículos, de maneira a atender a um grupo de clientes. A importância do problema de roteamento de veículos é dada pela sua relevância prática e sua considerável dificuldade. Na resolução desse problema, tratada neste artigo, é aplicado o Aprendizado de Reforço (AR) comparando a solução resultante com resultado de casos clássicos. Dessa forma, conseguiu-se quantificar, neste estudo de caso, o desempenho do algoritmo de Aprendizado de Reforço desenvolvido para estudo.

Palavras-chave: Aprendizado de Reforço, PRVC, Roteamento, Veículos, Caixeiro Viajante

1 Introdução

O trabalho propõe uma discussão sobre o uso do Aprendizado de Máquina para solucionar um problema de otimização combinatória, o PRVC. Gerando um conjunto de rotas de modo que as restrições envolvidas sejam atendidas e os custos globais minimizados. Para isso, aplica-se a técnica de Aprendizado de Reforço para solucionar PRVC. Sendo que os conceitos utilizados neste trabalho podem ser empregados em empresas que atuam nos setores de correios, logística, transporte de carga, transporte de passageiros e em diversas áreas que necessitem estabelecer a menor rota ou caminho.

O aumento na exigência dos mercados, vinculado à competitividade entre setores produtivos, faz com que haja uma busca constante por melhorias no desempenho operacional, para que os recursos disponíveis sejam aproveitados de forma a minimizar, principalmente,

os custos logísticos do processo produtivo, já que esses consistem em uma grande parcela dos custos totais de produção. Desta forma, é desejável desenvolver uma operação que atenda aos clientes de maneira satisfatória, garantindo um baixo custo desde a fabricação de produtos até a entrega final (BREISSAN; OLIVEIRA CAMPOS, 2020).

Portanto, esta pesquisa se justifica pela sua relevância econômica, visto que na maior parte das indústrias, a atividade de transporte representa um dos elementos mais importantes na composição do custo logístico (BITTENCOURT et al., 2012) e (TOTH; VIGO, 2002). Por isso, afirma-se que o PRVC é um dos problemas fundamentais na otimização combinatória com uma série de aplicações práticas em transporte, distribuição e logística (BORČINOVÁ, 2017).

2 O Problema de Roteamento de Veículos Capacitados (PRVC)

O PRVC é a versão mais simples do Problema de Roteamento de Veículos (PRV) (VIEIRA, 2013). Este roteamento faz parte da área de pesquisa operacional e de sua sub-área programação linear, que basicamente consiste em métodos de otimização combinatória de conjuntos finitos. Neste contexto, escolhem-se os valores das variáveis de decisão de forma a obter o melhor resultado da função objetivo sujeita às restrições especificadas. Assim definem-se (PIMENTEL, s.d.):

- **Função objetivo:** determina a qualidade da solução em função das variáveis de decisão. A solução do problema será, justamente, a tentativa de maximização ou minimização da função objetivo.
- **Restrições:** limitações físicas do algoritmo, que determinam os possíveis valores de cada variável de decisão e assim controlam o comportamento e a estabilidade do sistema.
- **Variáveis de decisão:** são as incógnitas, a serem determinadas pela solução do modelo, e os parâmetros (coeficientes), que são valores fixos no problema.

O objetivo do PRVC é encontrar um conjunto de rotas de custo total mínimo para uma frota de veículos capacitados com base em um único depósito, para atender a um conjunto de clientes sob as seguintes restrições (BORČINOVÁ, 2017) e (LAPORTE, 2007):

- Cada rota começa e termina no depósito;
- Cada cliente é visitado exatamente uma vez;
- A demanda total de cada rota não exceda a capacidade do veículo.

2.1 Formulação Matemática

O PRVC pode ser simplesmente declarado como o problema de determinar rotas ótimas através de um conjunto de localizações e definido em um grafo direcionado $G = (V, A)$ onde $V = (v_0, v_1, \dots, v_n)$ é um conjunto de vértices e $A = ((v_i, v_j) : v_i, v_j \in V, i \neq j)$ é um conjunto de arcos. O vértice v_0 representa um depósito onde está localizada uma frota de N_v veículos de mesma capacidade. O valor de N_v pode ser pré especificado ou livre, ou seja, seu valor máximo é definido por $N \leq n - 1$. Onde n é o número total de vértices e $n - 1$ todos os clientes (BARBAROSOGLU; OZGUR, 1999).

Uma matriz não negativa (distância/custo) $C = (c_{ij})$ é definida em A . Considerando que $c_{ij} = c_{ji} \forall (v_i, v_j)$, diz-se que o problema é simétrico e os arcos são representados por arestas não direcionadas. Um peso d_i , não negativo, é associado a cada vértice para representar a demanda do cliente em v_i , e naturalmente, o peso atribuído a qualquer rota não pode exceder a capacidade do veículo Q_v . Assim, o PRVC de depósito único visa determinar N_v rotas de veículos de custo total mínimo, cada uma começando e terminando no depósito, de modo que cada cliente seja visitado exatamente uma vez, sujeito as restrições mencionadas acima. Uma formulação matemática típica para o PRVC de depósito único é fornecida abaixo (BARBAROSOGLU; OZGUR, 1999).

Na formulação matemática a variável X_{ij}^v define uma variável binária que indica se um arco $A(v_i, v_j)$ está presente na rota percorrida pelo veículo v (BARBAROSOGLU; OZGUR, 1999).

Minimização das distâncias/custo dado pela Equação 1 (BARBAROSOGLU; OZGUR, 1999).

$$\sum_i \sum_j \sum_v c_{ij} X_{ij}^v \quad (1)$$

Restrição de um veículo por cliente, conforme Equações 2 e 3 (BARBAROSOGLU; OZGUR, 1999).

$$\sum_i \sum_v X_{ij}^v = 1, \forall j \quad (2)$$

$$\sum_j \sum_v X_{ij}^v = 1, \forall i \quad (3)$$

Veículo atende e deixa cliente através da Equação 4 (BARBAROSOGLU; OZGUR, 1999).

$$\sum_i X_{ip}^v - \sum_j X_{pj}^v = 0, \forall (p, v) \quad (4)$$

Demanda do cliente (d_i) não pode superar a capacidade do veículo (Q_v), conforme

Equação 5 (BARBAROSOGLU; OZGUR, 1999).

$$\sum_i d_i \left(\sum_j X_{ij}^v \right) \leq Q_v, \forall v \quad (5)$$

A disponibilidade do veículo não pode ser excedida e é dado pelas Equações 6 e 7 (BARBAROSOGLU; OZGUR, 1999).

$$\sum_{i=1}^n X_{i0}^v \leq 1, \forall v \quad (6)$$

$$\sum_{j=1}^n X_{0j}^v \leq 1, \forall v \quad (7)$$

Eliminação de subrota através das Equações 8, 9 e 10 (BARBAROSOGLU; OZGUR, 1999).

$$X_{ij}^v \in Z, \forall (i, j, v) \quad (8)$$

$$Z = \left\{ \left(X_{ij}^v \right) : \sum_{i \in B} \sum_{j \in B} X_{ij}^v \leq |B| - 1 \text{ } p / B \subseteq V / \{0\} \right\} \quad (9)$$

$$|B| \geq 2 \quad (10)$$

3 Aprendizado por Reforço (AR)

O aprendizado por reforço é um modelo de aprendizagem comportamental. O algoritmo recebe *feedback* da análise dos dados e então o usuário é guiado para o melhor resultado. O aprendizado por reforço difere de outros tipos de aprendizado supervisionado porque o agente aprendiz não é treinado com o conjunto de dados de amostra. Em vez disso, o agente aprende por tentativa e erro. Portanto, uma sequência de decisões bem sucedidas resultará no processo “reforço” porque resolve melhor o problema em questão (HURWITZ; KIRSCH, 2018).

Uma das maneiras mais fáceis de pensar sobre o aprendizado por reforço é a maneira como um animal é treinado para realizar ações com base em recompensas. Se o cão receber uma guloseima toda vez que se sentar ao comando, ele fará essa ação todas as vezes (HURWITZ; KIRSCH, 2018).

Para ilustrar concretamente a estrutura, vamos considerar os elementos necessários para construir um algoritmo AR, para isso temos (ZAI; BROWN, 2020):

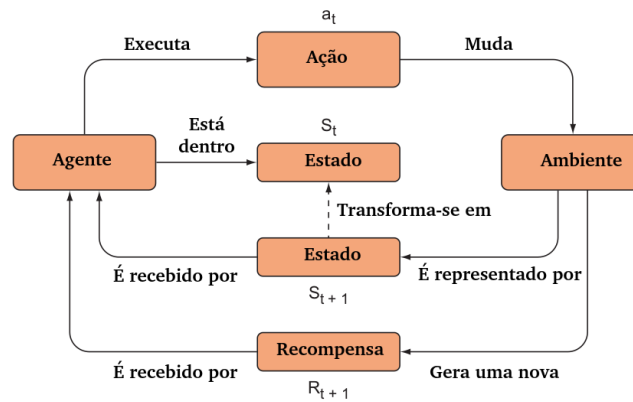
- Agente: é o algoritmo de aprendizado responsável pela tomada de ação ou decisão em

qualquer problema de AR;

- Ação (a_t): uma ação provoca uma alteração no estado do ambiente. Essa ação promove um efeito causal sobre o próximo estado do ambiente. No entanto, toda ação tomada é resultado da análise do estado atual do ambiente e da tentativa de tomar a melhor decisão com base nessas informações de um instante de tempo específico;
- Ambiente: é qualquer processo dinâmico que produz dados relevantes para atingir o objetivo;
- Estado (S_t): reflete o conhecimento do estado do ambiente em um instante de tempo;
- Recompensa (R_t): é um sinal que indica o quão bem o algoritmo de aprendizado está trabalhando para alcançar o objetivo global. A recompensa pode ser um sinal positivo (ou seja, indo bem, continue assim) ou um sinal negativo (ou seja, não faça isso), embora ambas as situações são chamadas de “recompensa”.

A Figura 1 ilustra como o agente aprendiz interage com o ambiente e o processo de aprendizado a cada instante t ($t = 1, 2, \dots, T$).

Figura 1 – Interação agente-ambiente no aprendizado por reforço.



Fonte: Zai e Brown (2020)

A função R_t é dada pela Equação 11 e é a soma das recompensas ao longo do tempo.

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad (11)$$

3.1 Método *Q-learning*

O algoritmo *Q-learning*, introduzido por Watkins e Dayan (1992), destaca-se por ser amplamente utilizado, por ser livre de modelos e por dispensar o conhecimento de uma política. Desta forma, o agente, através do algoritmo *Q-learning*, atualiza seu valor de função,

enquanto segue qualquer política. O objetivo é, a cada passo de um episódio, maximizar o valor da função $Q(s, a)$, definida como (SILVA, 2019):

$$Q_{t+1} = Q_t(S_t, a_t) + \alpha [r_t(S_t, a_t) + \gamma \max_{a_{t+1}} Q(S_{t+1}, a_{t+1}) - Q_t(S_t, a_t)] \quad (12)$$

Onde (SILVA, 2019):

- S_t é o estado atual;
- S_{t+1} é o estado resultante;
- a_t é a ação tomada;
- a_{t+1} é a ação resultante.

A taxa de aprendizado (α) e o fator de desconto (γ) são parâmetros que dependem diretamente do problema tratado. A função Q estima a utilidade esperada da tomada de uma ação a_t em um determinado estado S_t (SILVA, 2019). O Algoritmo 1 mostra o pseudocódigo utilizado para aplicar o Q -learning.

Algoritmo 1 - Q -learning

```

1: Definir os parâmetros:  $\alpha$ ,  $\gamma$  e  $\epsilon$ ;
2: Para cada  $(S_t, a_t)$  inicialize uma matriz  $Q_t(S_t, a_t) = 0$ ;
3: Observe o estado  $S_t$ ;
4: repita: até atingir o critério de parada
5:   Selecione a ação  $a_t$  usando a política  $\epsilon$ -greedy;
6:   Execute a ação  $a_t$ ;
7:   Receba a recompensa atual  $R_t(S_t, a_t)$ ;
8:   Observe o novo estado,  $S_{t+1}$ ;
9:   Atualize o item  $Q_t(S_t, a_t)$  de acordo com a Equação 12;
10:   $S_t = S_{t+1}$ ;
11: end
```

Fonte: Ottoni, Nepomuceno e Oliveira (2018)

3.2 Método SARSA

É uma modificação do Q -learning, esta modificação deu-se na forma em que a recompensa futura é concebida, enquanto o Q -learning recebe a recompensa máxima da matriz Q -learning o SARSA recebe a recompensa esperada baseada da última ação executada (SOUZA; OTTONI, 2021).

O SARSA recebeu esse nome porque envolve em seus termos os seguintes parâmetros (OTTONI; NEPOMUCENO; OLIVEIRA, 2018):

- S_t é o estado atual;
- a_t é a ação tomada;

- $r_t(S_t, a_t)$: reforço em relação a $S_t \times a_t$;
- S_{t+1} é o estado resultante;
- a_{t+1} é a ação resultante.

A Equação 13 descreve a atualização da matriz Q após a execução da ação a_t no estado S_t (OTTONI; NEPOMUCENO; OLIVEIRA, 2018):

$$Q_{t+1} = Q_t(S_t, a_t) + \alpha [r_t(S_t, a_t) + \gamma Q(S_{t+1}, a_{t+1}) - Q_t(S_t, a_t)] \quad (13)$$

Para resolver a Equação 13 utiliza-se o pseudocódigo dado pelo Algoritmo 2.

Algoritmo 2 - SARSA

```

1: Definir os parâmetros:  $\alpha$ ,  $\gamma$  e  $\epsilon$ ;
2: Para cada  $(S_t, a_t)$  inicialize uma matriz  $Q_t(S_t, a_t) = 0$ ;
3: Observe o estado  $S_t$ ;
4: Selecione a ação  $a_t$  usando a política  $\epsilon$ -greedy;
5: repita: até atingir o critério de parada
6:   Execute a ação  $a_t$ ;
7:   Receba a recompensa atual  $R_t(S_t, a_t)$ ;
8:   Observe o novo estado,  $S_{t+1}$ ;
9:   Selecione uma nova ação  $(a_{t+1})$  usando a política  $\epsilon$ -greedy;
10:  Atualize o item  $Q_t(S_t, a_t)$  de acordo com a Equação 13;
11:   $S_t = S_{t+1}$ ;
12:   $a_t = a_{t+1}$ ;
13: end

```

Fonte: Ottoni, Nepomuceno e Oliveira (2018)

4 Metodologia

Para atingir os objetivos deste trabalho foram seguidas as etapas abaixo:

- Estudo d caso;
- Solução inicial;
- Solução do AR;
- Validação do Algoritmo.

5 Estudo de Caso

Para o estudo de caso, foi escolhido a instância E-n51-k5 conforme a literatura apresentada em (RALPHS, s.d.). O problema corresponde a depósito, clientes e veículos (com capacidade). A saída do problema são rotas (uma para cada veículo) partindo e retornando ao

depósito, atendendo um conjunto de clientes de modo a não ultrapassar a capacidade veículo em cada rota.

Uma matriz de distância é criada da seguinte maneira: para cada par de vértices com coordenadas (X_i, Y_j) e $(X_{i'}, Y_{j'})$, é calculada a distância euclidiana entre eles através da Equação 14. O resultado deste módulo é uma matriz simétrica com as distâncias euclidianas entre quaisquer dois vértices para cada instância.

A recompensa para o deslocamento do veículo entre dois pontos, é definida pela matriz distância e quando maior a distância percorrida menor será a recompensa.

$$d_{ij} = \sqrt{(X_i - X_{i'})^2 + (Y_j - Y_{j'})^2} \quad (14)$$

Tabela 1 – Matriz de distância.

	$i = 1$	$i = 2$	$i = 3$	...
$j = 1$	d_{11}	d_{21}	d_{31}	...
$j = 2$	d_{21}	d_{22}	d_{32}	...
$j = 3$	d_{31}	d_{32}	d_{33}	...
...	...	...	...	...

Fonte: Díaz (s.d.)

5.1 Solução Heurística

Para produzir uma solução inicial afim de avaliar os resultados produzidos pelos algoritmos *Q-learning* e SARSA utiliza-se uma heurística gulosa, conforme os passos descritos abaixo:

- Passo 1: Estabelece-se uma rota principal e única saindo do depósito em direção ao vizinho mais próximo e com o menor custo, retornando ao depósito após visitar todos os clientes. É idêntico ao Problema do Caixeiro Viajante, pois desconsidera-se a capacidade do veículo e as demandas dos clientes;
- Passo 2: Subdivide-se a rota principal respeitando a capacidade dos veículos e as necessidades dos clientes, gerando sub-rotas. Os veículos são carregados considerando a ordem estabelecida no passo 1;
- Passo 3: Carrega-se cada veículo até sua capacidade máxima. Se a mesma for ultrapassada pela demanda do cliente, busca-se o próximo cliente da ordem até que não hajam mais clientes;

- Passo 4: Cria-se uma nova sub-rotas para o próximo veículo e os clientes restantes, seguindo o passo 3 e a ordem de clientes da rota de referência do passo 1;
- Passo 5: Calcula-se o custo total considerando todas as rotas geradas.

5.2 Solução do AR

Abaixo são descritos os passos utilizados para solucionar o problema através dos algoritmos *Q-learning* e *SARSA*, sendo que a única diferença entre estes algoritmos são as Equações 12 e 13 utilizada por eles, respectivamente.

- Passo 1: Gera a uma matriz *Q* preenchida com zeros, esta matriz guarda os resultados das Equações 12 e 13 e as relaciona com cada ação executada;
- Passo 2: Baseado na política estabelecidas define-se uma ação a ser executada;
- Passo 3: Ao executar ação definida recebe-se a recompensa fornecida pela matriz distância;
- Passo 4: Aplica-se a Equação 12 ou a Equação 13 de acordo com o algoritmo utilizado;
- Passo 5: Salva a ação executada junto com o resultado obtido no Passo 4 na matriz *Q* de resultados;
- Passo 6: Repetem-se os Passos 2 a 5 por várias vezes;
- Passo 7: Montam-se as rotas baseado nos resultados gravados na matriz *Q* de resultados e na capacidade do veículo.

5.3 Validação do Algoritmo

Para a validar os resultados foram selecionadas heurísticas clássicas e trabalhos que utilizam a Busca Tabu para que pudessem ser comparados com os resultados obtidos pelos algoritmos implementados. A Tabela 2 apresenta os melhores resultados.

Tabela 2 – Melhores resultados.

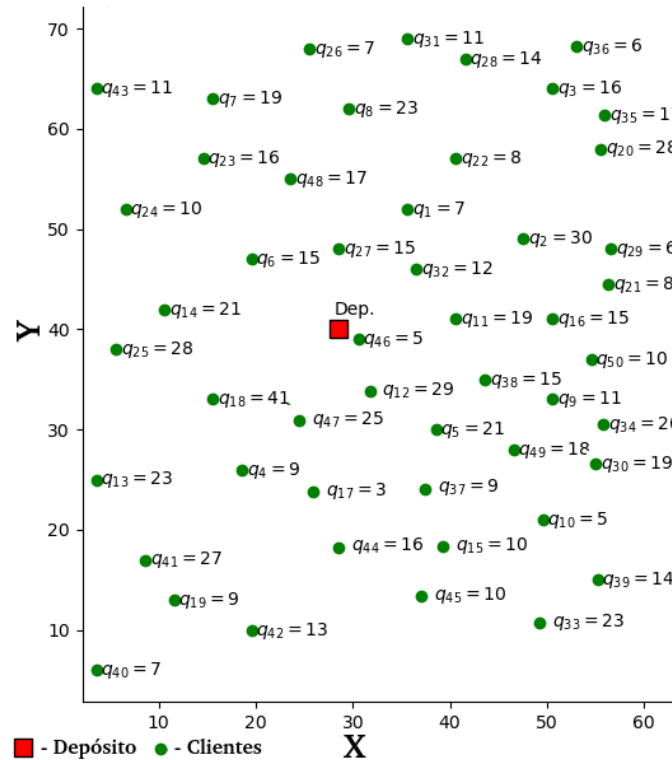
Instância	Christofides, Eilon Custo	Christofides, Mingozzi, Toth Custo
E051-05e		
E-n51-k5	521	524.61
Vrpnc1		

Fonte: Díaz (s.d.)

6 Resultados

Nesta seção são apresentados os resultados obtidos através das configurações e simulações realizadas para as instâncias E051-05e, E-n51-k5 e Vrpnc1, que são equivalentes. Sendo que todas as simulações foram executadas com o *Python* 3.9. A Figura 2 ilustra a localização do depósito e dos clientes para as instâncias estudadas.

Figura 2 – Localização do depósito e dos clientes no eixo de coordenadas XY.



Fonte: Autor (2023)

6.1 Solução Caixeiro Viajante

Conforme mencionado no Passo 1 da Seção 5.1, para se determinar uma solução heurística primeiramente se estabelece uma rota principal. A rota incia-se no depósito que é a posição 0, depois vai para o primeiro cliente que está na posição 46 e segue pelos demais clientes até chegar no da posição 40, quando o veículo retorna ao depósito. O custo total desta rota é dado pela soma de todas as distâncias percorridas, deste a saída até o retorno ao depósito.

Rota:

[46, 12, 47, 4, 17, 37, 15, 44, 42, 19, 41, 13, 25, 14, 6, 48, 27, 1, 32, 11, 38, 5, 49, 9, 50, 16, 2, 29, 21, 34, 30, 10, 39, 33, 45, 18, 24, 23, 7, 26, 8, 31, 28, 3, 20, 35, 36, 22, 43, 40]

Custo Total: 565,61

6.2 Solução Heurística

Do caixeiro viajante evolui para a solução construtiva já considerando a capacidade do veículo e a demanda do cliente, conformes Passos 2 a 5 da Seção 5.1.

Rota 1: [46, 12, 47, 4, 17, 37, 15, 44, 42, 19, 41, 10]

Rota 2: [13, 25, 14, 6, 48, 27, 1, 32, 11]

Rota 3: [38, 5, 49, 9, 50, 16, 2, 29, 21, 34]

Rota 4: [30, 39, 33, 45, 18, 24, 23, 7, 26]

Rota 5: [8, 31, 28, 3, 20, 35, 36, 22, 43, 40]

Custo Total: 767.73

6.3 Solução Q-learning

Para chegar neste resultado foram seguidos os passos da Seção 5.2 junto com a Equação 12. Os hiperparâmetros foram definidos como sendo a taxa de aprendizado (α) = 0,9, o fator de desconto (γ) = 0,9 e a política ϵ -greedy (ϵ) = 0,5.

Rota 1: [18, 25, 29, 7, 13, 30, 22, 31]

Rota 2: [23, 40, 14, 12, 15, 34, 4, 45, 21, 6]

Rota 3: [24, 17, 27, 26, 33, 9, 10, 32, 38, 50, 19, 39, 43, 28]

Rota 4: [37, 42, 2, 36, 44, 5, 8, 11, 3]

Rota 5: [16, 48, 47, 1, 46, 20, 41, 49, 35]

Custo Total: 1745,68

6.4 Solução SARSA

Esta etapa também segue os passos da Seção 5.2, mas utiliza a Equação 13. Já como hiperparâmetros têm-se a taxa de aprendizado (α) = 0,9, o fator de desconto (γ) = 0,9 e a política ϵ -greedy (ϵ) = 0,5.

Rota 1: [12, 34, 42, 38, 8, 35, 22, 41]

Rota 2: [46, 2, 33, 14, 32, 27, 9, 44, 17, 16]

Rota 3: [30, 45, 13, 10, 50, 18, 39, 6, 43, 15]

Rota 4: [11, 36, 20, 7, 3, 23, 47, 19, 31, 29]

Rota 5: [21, 49, 5, 40, 25, 1, 26, 37, 4, 28, 24, 48]

Custo Total: 1641,92

7 Conclusão

A solução do PRVC pode ser obtida por diversas técnicas como heurísticas, metaheurísticas, aprendizado de máquina e também por combinações entre uma ou mais técnicas. Por tanto, a escolha adequada da técnica é fator decisivo para a obtenção de bons resultados, elevando-se o desempenho geral do processo.

Neste trabalho, estudou-se a versão clássica do PRVC considerando a capacidade do veículo como principal restrição. Assim adotou-se o *Q-learning* e SARSA para solucionar este problema em função da boa reputação destes. Com isso, foram implementados estes algoritmos e seus resultados foram comparados com uma solução trivial, a heurística gulosa, e com os resultados obtidos pelos trabalhos de referências, apresentados na Seção 5.3.

Ao comparar os resultados obtidos pelo *Q-learning* e SARSA é possível perceber que o *Q-learning* obteve um resultado melhor, mas que a diferença não foi tão significativa. Isso portanto revela que, com a metodologia adotada, não existe entre eles um algoritmo que seja recomendado em detrimento do outro.

Durante o desenvolvimento do trabalho constatou-se a importância dos movimentos aleatórios para diversificar a solução, para isso, definiu-se um ϵ -greedy (ϵ) = 0,5. O que permite uma exploração eficaz fugindo de mínimos locais.

Uma melhoria que poderia ser incorporada em um trabalho futuro é a implementação do Aprendizado por Reforço de forma híbrida, combinado a outras heurísticas ou metaheurísticas de diversificação e intensificação.

Referências

BARBAROSOGLU, Gulay; OZGUR, Demet. A tabu search algorithm for the vehicle routing problem. **Computers & Operations Research**, v. 26, p. 255–270, 1999. DOI: [https://doi.org/10.1016/S0305-0548\(98\)00047-1](https://doi.org/10.1016/S0305-0548(98)00047-1).

BITTENCOURT, Gustavo Cunha de et al. Problema de Roteamento de Veículos Capacitados (PRVC): Solução Manual X Busca Dispersa. In: XLIV - Simpósio Brasileiro de Pesquisa Operacional (SBPO). [S.l.: s.n.], 2012.

BORČINOVA, Zuzana. Two models of the capacitated vehicle routing problem. **Croatian Operational Research Review**, v. 8, p. 463–469, 2017. DOI: <https://doi.org/10.17535/crorr.2017.0029>.

BRESSAN, Glaucia Maria; OLIVEIRA CAMPOS, Luciene Aparecida de. Otimização de problemas de roteamento de veículos: soluções heurísticas. **Revista Eletrônica Paulista de Matemática**, v. 18, p. 60–76, 2020. DOI:

<https://doi.org/10.21167/cqdvoll18ic202023169664gmb1aoc6076>.

DÍAZ, Bernabé Dorronsoro. The VRP Web. Acessado em abril de 2023. [S.l.]. Disponível em: <https://www.bernabe.dorronsoro.es/vrp/index.html?/results/results.html>.

HURWITZ, Judith; KIRSCH, Daniel. **Machine Learning For Dummies**. [S.l.]: John Wiley & Sons, Inc., 2018.

LAPORTE, Gilbert. What you should know about the vehicle routing problem. **Naval Research Logistics**, v. 54, p. 811–819, 2007. DOI:

<https://doi.org/10.1002/nav.20261>.

OTTONI, André L. C.; NEPOMUCENO, Erivelton G.; OLIVEIRA, Marcos S. de. A Response Surface Model Approach to Parameter Estimation of Reinforcement Learning for the Travelling Salesman Problem. **Journal of Control, Automation and Electrical Systems**, v. 29, p. 350–359, 2018. DOI:

<https://doi.org/10.1007/s40313-018-0374-y>.

PIMENTEL, Renato. GBC051 – Computação Científica e Otimização. Acessado em abril de 2023. [S.l.]. Disponível em:

<https://www.facom.ufu.br/~rpimentel/files/gbc051-2021-2/gbc051`2021`Notas`Aula.pdf>.

RALPHS, Ted. Vehicle Routing Data Sets. Acessado em abril de 2023. [S.l.]. Disponível em: <https://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/index.htm.old>.

SILVA, Maria Amélia Lopes. **AMAM: Framework Multiagente para Otimização usando Metaheurísticas**. 2019. F. 135. Tese (Doutorado) – CEFET-MG.

SOUZA, Gleice Kelly Barbosa; OTTONI, André Luiz Carvalho. AutoRL-TSP-RSM: sistema de aprendizado por reforço automatizado com metodologia de superfície de resposta para o problema do caixeiro viajante. **Revista Brasileira de Computação Aplicada**, v. 13, p. 86–100, 2021. DOI: <https://doi.org/10.5335/rbca.v13i3.12653>.

TOTH, Paolo; VIGO, Daniele. **Discrete Mathematics and Applications: The Vehicle Routing Problem**. [S.l.]: SIAM, 2002.

VIEIRA, Heloisa Passarelli. **Metaheurística para a Solução de Problemas de Roteamento de Veículos com Janela de Tempo**. 2013. F. 108. Diss. (Mestrado) – Universidade Estadual de Campinas.

ZAI, Alexander; BROWN, Brandon. **Deep Reinforcement Learning in Action**. [S.l.]: Manning Publications Co, 2020.