DOI: 10.48011/asba.v2i1.1098

Transferência de Aprendizado por Reforço em Problemas de Otimização Combinatória

André Luiz C. Ottoni [*] **Marcos S. de Oliveira** [**]
Daniela C. R. de Oliveira [**] **Erivelton G. Nepomuceno** [***]

[*] *Centro de Ciências Exatas e Tecnológicas, Universidade Federal do Recôncavo da Bahia, Cruz das Almas, BA, Brasil (e-mail: andre.ottoni@ufrb.edu.br).*
[**] *Departamento de Matemática e Estatística, Universidade Federal de São João del-Rei, São João del-Rei, MG, Brasil (e-mails: mso@ufsj.edu.br e daniela@ufsj.edu.br)*
[***] *Grupo de Controle e Modelagem, Departamento de Engenharia Elétrica, Universidade Federal de São João del-Rei, São João del-Rei, MG, Brasil (e-mail: nepomuceno@ufsj.edu.br)*

Abstract: The Reinforcement Learning (RL) is a Machine Learning technique with important applications in combinatorial optimization problems. However, the literature lacks studies on the Transfer Reinforcement Learning between optimization domains. Based on this, the objective of this paper is to apply and analyze the Transfer RL between the problems: Travelling Salesman Problem (TSP) and Sequential Ordering Problem (SOP). For this, the Travelling Salesman Problem Library (TSPLIB) and the SARSA algorithm were adopted. The proposed methodology comprises: RL system modeling, generating the knowledge base (QTSP), experiments for transfer learning and results analysis. The results obtained from statistical tests show that, in general, adopting the transfer of knowledge between the problems made possible the calculation of better performance metrics of the target domain (SOP).

Resumo: O Aprendizado por Reforço (AR) é uma técnica de *Machine Learning* com importantes aplicações em problemas de otimização combinatória. No entanto, a literatura carece de estudos sobre a transferência de AR entre domínios de otimização. Baseando-se nisso, o objetivo deste trabalho foi aplicar e analisar a transferência de conhecimento do AR entre o Problema do Caixeiro Viajante (TSP) e o *Sequential Ordering Problem* (SOP). Para isso, foi adotada a biblioteca de instâncias TSPLIB e o algoritmo SARSA. A metodologia proposta compreende: modelagem do sistema de AR, geração da base de conhecimento (QTSP), experimentos para transferência de aprendizado e análise dos resultados. Os resultados obtidos a partir de testes estatísticos, apontam que, em geral, adotar a transferência de conhecimento entre os problemas possibilitou o cálculo de melhores métricas de desempenho do domínio objetivo (SOP).

Keywords: Reinforcement Learning; Transfer Learning; Combinatorial Optimization; Travelling Salesman Problem; Sequential Ordering Problem.

Palavras-chaves: Aprendizado por Reforço; Transferência de Aprendizado; Otimização Combinatória; Problema do Caixeiro Viajante; Sequential Ordering Problem.

1. INTRODUÇÃO

O Aprendizado por Reforço (AR), em inglês, *Reinforcement Learning*, reúne importantes métodos de *Machine Learning* (ML) (Watkins and Dayan, 1992; Russell and Norving, 2013; Sutton and Barto, 2018). No AR, um agente aprende a partir da interação com o ambiente (Russell and Norving, 2013; Sutton and Barto, 2018). Além disso, o AR é fundamentado nos Processos de Decisão de Markov e o objetivo é maximizar a recompensa recebida ao longo do tempo (Russell and Norving, 2013; Sutton and Barto, 2018).

Um campo de estudo do AR e do ML é o de Transferência de Aprendizado (TL - *Transfer Learning*) (Carroll and Peterson, 2002; Taylor and Stone, 2009; Lazaric and Restelli, 2011; Da Silva and Reali Costa, 2019; Da Silva et al., 2020). No TL, um dos objetivos principais é transferir o conhecimento aprendido por um agente entre tarefas distintas (Taylor and Stone, 2009). Nesse sentido, aplicar o TL em problemas de AR pode permitir, por exemplo, acelerar o aprendizado, diminuir o custo computacional e melhorar o desempenho no domínio objetivo (Taylor and Stone, 2009; Da Silva and Reali Costa, 2019).

Seguindo essa linha, técnicas de *Transfer Reinforcement Learning* vêm sendo aplicadas em importantes domínios, com destaque para as áreas de Robótica (Peterson et al., 2001; Wang et al., 2014; Tommasino et al., 2019; Arnekvist et al., 2019) e Sistemas Multiagentes (Hou et al., 2017; Da Silva and Reali Costa, 2019; Cai et al., 2020). No entanto, a literatura tem dado pouca atenção para a

transferência de Aprendizado por Reforço em problemas de otimização combinatória.

A área de otimização combinatória é um campo com vários estudos com AR (Gambardella and Dorigo, 1995; Bianchi et al., 2009; Lima Júnior et al., 2010; Costa et al., 2016; Alipour et al., 2018; Lins et al., 2019). Nesse aspecto, alguns domínios com aplicações de técnicas de AR são: Problema do Caxeiro Viajante (TSP) (Gambardella and Dorigo, 1995; Lima Júnior et al., 2010), Problema dos K-Servos (Costa et al., 2016; Lins et al., 2019), Problema da Mochila Multidimensional (Ottoni et al., 2017), Roteamento de Veículos (Silva et al., 2019) e *Sequential Ordering Problem* (SOP) (Ottoni et al., 2020).

Dessa forma, o objetivo deste trabalho é propor e analisar a transferência de AR entre dois relevantes domínios de otimização combinatória: TSP (Gambardella and Dorigo, 1995; Lima Júnior et al., 2010; Alipour et al., 2018) e SOP (Escudero, 1988; Gambardella and Dorigo, 2000; Letchford and Salazar-González, 2016; Skinderowicz, 2017). Para isso, foram adotadas instâncias da biblioteca TSPLIB (Reinelt, 1991) e o algoritmo SARSA (Sutton and Barto, 2018). A metodologia proposta aborda a geração da base de conhecimento no domínio fonte (TSP) e os experimentos/análise da transferência de aprendizado para o domínio objetivo (SOP).

Este artigo está organizado em seções. Na seção 2, são definidos aspectos teóricos dos problemas de otimização combinatória adotados, AR e transferência de aprendizado. As seções 3 e 4, apresentam a metodologia e os resultados, respectivamente. Finalmente, na seção 5 são apresentadas as conclusões do trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Problemas de Otimização Combinatória

Os problemas de otimização combinatória abordados neste trabalho são: Problema do Caixeiro Viajante (TSP - *Travelling Salesman Problem* (Gambardella and Dorigo, 1995; Lima Júnior et al., 2010) e *Sequential Ordering Problem* (Escudero, 1988; Gambardella and Dorigo, 2000; Skinderowicz, 2017). O objetivo nesses dois domínios é minimizar a rota entre um conjunto de localidades. Além disso, o agente deve visitar cada cidade uma única vez e voltar ao nó inicial ao término do percurso.

O TSP pode ser formulado como um grafo com um conjunto de nós e arcos, sendo que c_{ij} é o custo dado para cada aresta (i, j) (Bodin et al., 1983; Gambardella and Dorigo, 2000). Em problemas de TSP simétrico, o custo de deslocamento entre duas cidades é equivalente nos dois sentidos de movimentação, ou seja, $c_{ij} = c_{ji}$. Já no TSP assimétrico (ATSP), o custo de ir de i para j pode ser diferente do custo do deslocamento de j para i ($c_{ij} \neq c_{ji}$) (Ottoni et al., 2020).

Nesse sentido, o SOP é semelhante ao ATSP com a adição de restrições de precedência ($c_{ij} = -1$) (Gambardella and Dorigo, 2000; Ottoni et al., 2020). O custo no SOP pode assumir $c_{ij} \geq 0$ ou $c_{ij} = -1$ (com $c_{ji} \geq 0$) (Gambardella and Dorigo, 2000). Se existe uma restrição de precedência no arco (i,j), $c_{ij} = -1$, indica que o nó j deve preceder o nó i na ordem de visita as localidades (Gambardella and Dorigo, 2000).

Uma formulação matemática baseada no TSP (Bodin et al., 1983) para o SOP é apresentada nas Eqs. (1) à (6) (Ottoni et al., 2020):

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

sujeito à:

$$\sum_{i=1}^{N} x_{ij} = 1 \qquad (\forall j = 1,...,N), \tag{2}$$

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad (\forall i = 1,...,N), \tag{3}$$

$$x_{ij} \in \{0,1\} \qquad (\forall i,j = 1,...,N), \tag{4}$$

$$X = x_{ij} \in S \qquad (\forall i,j = 1,...,N), \tag{5}$$

$$c_{ij} \geq 0 \vee c_{ij} = -1 \wedge c_{ji} \geq 0 \qquad (\forall i,j = 1,...,N), \tag{6}$$

em que, N é o conjunto de nós. A Eq. (1) representa o objetivo de minimizar a distância na rota. Nesse sentido, c_{ij} é o custo entre as cidades (i e j) e $x_{i,j}$ é a variável decisão. As Eqs. (2) e (3) garatem que cada nó será visitado uma única vez. Já as Eqs. (4) define x_{ij} como binária. Na Eq. (5), S representa qualquer conjunto de restrições que proíba a formação de soluções com sub-rotas. Finalmente, a Eq. (6) é a restrição de precedência do SOP.

A biblioteca *Travelling Salesman Problem Library* (TSPLIB) [1] (Reinelt, 1991) é exemplo de um repositório de dados com instâncias para estudos de casos de problemas de otimização combinatória, como TSP e SOP. A TSPLIB é amplamente abordada na literatura (Gambardella and Dorigo, 2000; Bianchi et al., 2009; Lima Júnior et al., 2010; Skinderowicz, 2017; Ottoni et al., 2020) e também é adotada neste trabalho.

2.2 Aprendizado por Reforço

O Aprendizado por Reforço é fundamentando nos Processos de Decisão de Markov (PDM) (Russell and Norving, 2013; Sutton and Barto, 2018). Um PDM é estruturado em: um conjunto finito de estados (S), um conjunto finito de ações (A), um conjunto finito de reforços (R) e um modelo de transição de estados (T) (Russell and Norving, 2013; Sutton and Barto, 2018).

No AR, um agente aprende por tentativa e erro a tomar decisões em um ambiente. Basicamente, o agente aprendiz realiza uma sequência de três passos em várias repetições: (i) percebe o estado atual (s), (ii) executa uma ação (a) e (iii) recebe uma recompensa ($r(s,a)$). Além disso, em cada instante de tempo (t) é atualizada uma matriz de aprendizado (Q), que armazena o conhecimento aprendido. Essa matriz Q possui a dimensão dada pelo número de estados versus o número de ações do modelo.

A Eq. (7) apresenta o método de atualização da matriz Q pelo Algoritmo SARSA (Sutton and Barto, 2018):

$$Q_{t+1} = Q_t(s,a) + \alpha[r(s,a) + \gamma Q_t(s',a') - Q_t(s,a)]. \tag{7}$$

[1] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

em que, s é o estado e a a ação no instante t; $r(s,a)$ é a recompensa pela execução de a em s; s' é o novo estado e a' é a nova ação selecionada; Q_t e Q_{t+1} são matrizes no instante atual e em $t+1$, respectivamente; α é a taxa de aprendizado; γ é o fator de desconto. O Algoritmo 1 representa o SARSA.

```
1.  Para cada (s,a) inicialize Q(s,a)=0;
2.  Observe  o estado s;
3.  Selecione a ação a usando a política e-greedy;
4.  Repita até o critério de parada ser satisfeito
5.      Execute a ação a;
6.      Receba a recompensa imediata  R(s,a);
7.      Observe o novo estado s';
8.      Selecione a nova ação a' usando e-greedy;
9.      Atualize o item Q(s,a) de acordo com (2);
10.     s = s';
11.     a = a';
12. Fim Repita
```

Algoritmo 1: SARSA.

No Algoritmo 1 é adotada a política de seleção de ações $\epsilon - greedy$ (Sutton and Barto, 2018). Esse método utiliza o parâmetro ϵ no controle entre gula e aleatoriedade na tomada de decisão. Por exemplo, se $\epsilon = 0{,}1$, o sistema selecionará em 10% dos casos ações aleatórias e em 90% das situações as ações mais bem estimadadas para cada estado na matriz de aprendizado.

2.3 Transferência de Aprendizado

As técnicas de *Transfer Learning* tem como ideia central utilizar o conhecimento já adquirido (tarefas já realizadas) em problemas relacionados (Fernández and Veloso, 2006; Taylor and Stone, 2009; Da Silva and Reali Costa, 2019). Nesse sentido, alguns objetivos dos métodos TL são melhorar o desempenho e reduzir o tempo necessário para aprender uma tarefa complexa (Taylor and Stone, 2009).

Para efetuar a transferência de aprendizado, a base de conhecimento é gerada em um domínio fonte e utilizada em um domínio objetivo (Carroll and Peterson, 2002; Taylor and Stone, 2009). Um dos métodos de TL em AR é o de transferência direta da matriz de aprendizado Q (Carroll and Peterson, 2002). Nesse caso, os valores Q (tarefa fonte) são adotados como ponto de partida da matriz de aprendizado no domínio objetivo. A Eq (8) representa a transferência direta de AR entre domínios (Carroll and Peterson, 2002):

$$\forall s, \forall a, Q_{objetivo}(s,a) = Q_{fonte}(s,a). \tag{8}$$

3. METODOLOGIA

A metodologia proposta neste trabalho possui quatro etapas: (1) Desenvolvimento do sistema de Aprendizado por Reforço; (2) Geração da base de conhecimento do TSP; (3) Realização de experimentos para transferência de aprendizado entre os domínios do TSP e SOP; (4) Análise comparativa dos resultados entre adotar ou não TL entre os problemas simulados.

3.1 Sistema de Aprendizado por Reforço

O sistema de AR desenvolvido visa a aplicação do algoritmo SARSA para experimentos com problemas de otimização combinatória: TSP e SOP. Para isso, foi adotada a estrutura de modelo (ações, estados e reforços) baseada em trabalhos da literatura (Gambardella and Dorigo, 1995; Bianchi et al., 2009; Lima Júnior et al., 2010; Ottoni et al., 2018, 2020):

- Estados: são as localidades que devem ser visitadas para a formação da rota. Assim, o número de estados varia de acordo com a quantidade de nós (N) da instância.
- Ações: representam os deslocamentos possíveis entre as localidades (estados). O número de ações inicial é equivalente ao quantitativo de estados do modelo. Porém, as ações disponíveis para execução variam de acordo com as cidades já visitadas no desenvolvimento da rota.
- Reforços: é dado em função do custo de deslocamento (c_{ij}) entre a cidade de partida (i) e a localidade de destino (j). Quanto maior a distância entre os nós, mais negativo será o reforço, conforme Eq. (9):

$$R = -c_{ij} \tag{9}$$

Além disso, para tratar as restrições de precedência do SOP, foi adotado o algoritmo RLSOP (Ottoni et al., 2020). O RLSOP (Algoritmo 2) verifica se as ações (localidade de chegada) selecionadas pelo método $\epsilon - greedy$ possuem restrições de precedência. Em caso positivo, outro destino é selecionado e novamente verificado.

```
1. a_t = e-greedy();
2. cont = 0;
3. Enquanto (cont == 0)
4.   Se (existem restrições de precedência  para a  ação
                selecionada)
5.      Se (ao menos uma ação correspondente às restrições
                de precedência de a_t ainda não foi sele-
                cionada)
6.          cont = 0;
7.      Senão
8.           cont = 1;
9.      FimSe
10.  FimSe
11.    Se (cont == 0)
12.        retirar a ação  a_t  da lista de ações disponí-
               veis no instante t;
13.        a_t = e-greedy();
14.    FimSe
15. FimEnquanto
16. Retornar a_t
```

Algoritmo 2: RLSOP - Algoritmo de análise de restrições de precedência na seleção de ações do AR para o SOP (Ottoni et al., 2020).

3.2 Geração da Base de Conhecimento

A etapa de geração da base de conhecimento foi realizada através de experimentos com instâncias TSPLIB do Problema do Caixeiro Viajante Assimétrico. A Tabela 1 mostra os quatro problemas TSP adotados, com seus respectivos números de nós (N). Além disso, a Tabela 1

apresenta as instâncias SOP que são baseadas nos dados dos problemas TSP (distâncias entre os nós), acrescentado em cada uma delas um conjunto de restrições de precedência ($c_{ij} = -1$).

Tabela 1. Problemas TSP adotados, respectivos número de nós e valor da rota ótima. A quarta coluna apresenta as instâncias SOP originadas a partir desses problemas TSP.

Problema	Nós	Ótimo	Instâncias SOP
br17	17	39	br17.10 e br17.12
ft53	53	6905	ft53.1, ft53.2, ft53.3 e ft53.4
kro124p	100	36230	kro124p.1, kro124p.2, kro124p.3 e kro124p.4
p43	43	5620	p43.1, p43.2, p43.3 e p43.4

Os experimentos com cada uma das 4 instâncias TSP foram configurados com 10 mil episódios, sendo que, um episódio equivale a realizar uma rota completa entre os nós. A matriz de aprendizado nessas simulações foi inicializada com zeros. Além disso, os parâmetros foram definidos como $\alpha = 0{,}75$, $\gamma = 0{,}15$ e $\epsilon = 0{,}01$, baseados nos resultados de Ottoni et al. (2018, 2020).

Ao término de cada simulação foi armazenada a matriz de aprendizado Q final (QTSP). Nesse sentido, foram geradas 4 matrizes QTSP, uma por instância TSP (br17, ft53, kro124p e p43). Essa base de conhecimento foi adotada nos experimentos de transferência de aprendizado para o domínio SOP, explicitados na próxima seção.

3.3 Experimentos para Transferência de Aprendizado

Nesta etapa, foram realizados experimentos para avaliar a influência da tranferência de conhecimento entre os domínios TSP (fonte) e SOP (objetivo). A Tabela 2 apresenta as 14 instâncias SOP da TSPLIB adotadas, seus respectivos valor ótimo, número de nós e restrições de precedência.

Tabela 2. Problemas SOP adotados, respectivos número de nós, restrições de precedência e valor da rota ótima de acordo com a TSPLIB. O número de restrições é referente a quantidade de valores de $c_{ji} = -1$ na instância.

Problema	Nós	Restrições	Ótimo
br17.10	18	48	55
br17.12	18	55	55
ft53.1	54	117	7.536
ft53.2	54	135	8.026
ft53.3	54	322	10.262
ft53.4	54	865	14.425
kro124p.1	101	232	38.762
kro124p.2	101	267	39.841
kro124p.3	101	465	43.904
kro124p.4	101	2.504	73.021
p43.1	44	96	28.140
p43.2	44	119	28.480
p43.3	44	181	28.835
p43.4	44	581	83.005

Os experimentos foram realizados adotando duas condições iniciais para a matriz de aprendizado Q:

- Q0: sem transferência de aprendizado. A matriz de aprendizado é iniciada com todos os valores nulos.
- QTSP: adotando a base de conhecimento gerada a partir dos experimentos com o domínio fonte (TSP).

Para a utilização da base de conhecimento (QTSP), foi necessário efetuar ajustes nas matrizes de aprendizado originais (experimentos com o TSP). Isso porque, o espaço de estados dos domínios TSP e SOP são distintos. Por exemplo, a instância kro124p (fonte TSP) possui 100 nós, enquanto que os problemas SOP equivalentes contém $N+1$ localidades: kro124.p1, kro124.p2, kro124.p3 e kro124.p4. Dessa forma, foram adicionadas uma linha e uma coluna (com zeros) nas matrizes da base de conhecimento QTSP, para então ser utilizada pelo domínio SOP.

Os experimentos com cada um dos problemas SOP e condições iniciais (Q0 e QTSP) foram executados com 100 episódios em 10 épocas (repetições). Vale ressaltar que, foram utilizados somente 100 episódios nesta etapa, pois o objetivo é analisar a adoção da transferência de aprendizado como método para acelerar o AR no domínio objetivo (Taylor and Stone, 2009). Dessa forma, avaliar se a adoção da base de conhecimento (QTSP) reproduz bons resultados em poucos episódios de simulação do SOP.

Os parâmetros desta etapa foram definidos de forma equivalente a seção anterior: $\alpha = 0{,}75$, $\gamma = 0{,}15$ e $\epsilon = 0{,}01$.

3.4 Metodologia de Análise

A metodologia de análise proposta neste trabalho possui três etapas: (i) Análise Preliminar; (ii) Análise do Tempo Computacional; (iii) Análise Gráfica. Esse método foi baseado em métricas presentes na literatura (Taylor and Stone, 2009; Da Silva and Reali Costa, 2019).

Na Análise Preliminar, foram comparados as médias dos resultados alcançados nas instâncias SOP com e sem transferência de aprendizado do domínio de origem (TSP). Essa métrica é semelhante ao "*Total Reward*" (Taylor and Stone, 2009), que adota o reforço total recebido pelo agente durante o aprendizado.

Já a segunda etapa, buscou avaliar as diferenças entre os tempos computacionais de simulações com Q0 e QTSP. Conforme Taylor and Stone (2009), um possível objetivo da transferência de conhecimento entre domínios é a redução do tempo para aprender uma tarefa complexa.

Nessas duas primeiras etapas, foi utilizado o teste t de comparação de médias para duas amostras independentes. Esse método estatístico avalia se duas médias populacionais (μ_1 e μ_2) são estatisticamente iguais ou diferentes (Montgomery, 2017), através das hipóteses:

$$\begin{cases} H_0 : \mu_1 = \mu_2, \\ H_1 : \mu_1 \neq \mu_2. \end{cases}$$

Adotando um nível de significância de 5%, a regra de decisão fica da seguinte forma: se $p > 0{,}05$, considera-se que as médias são estatisticamente iguais; por outro lado, quando $p \leq 0{,}05$, conclui-se que as médias são estatisticamente diferentes. Para a validação dos resultados, é necessário verificar a suposição de normalidade para cada uma das amostras independentes. Neste estudo, foi adotado o teste de *Kolmogorov-Smirnov (KS)* (Lopes, 2011), em que a suposição de normalidade é satisfeita desde que o p-valor

deste teste seja maior que 5%. Dessa forma, após a verificação das suposições estarem garantidas pelo teste *KS* para todas as amostras, o teste *t* foi aplicado para comparar as médias dos resultados para distância na rota (análise preliminar) e tempo computacional para cada uma das 14 instâncias simuladas.

Por fim, na Análise Gráfica foram avaliadas as curvas de aprendizado das seguintes instâncias SOP: krop124p.1, krop124p.2, krop124p.3 e krop124p.4. Nesse aspecto, foram analisadas três métricas: distância calculada no episódio inicial (d_0), distância no episódio final (d_f) e menor solução encontrada (d_{min}). Avaliar a solução dos primeiros ("*Jump Start*") e últimos episódios ("*Asymptotic Performance*") é importante para verificar qual a diferença de resultados de Q0 e QTSP nessas situações (Taylor and Stone, 2009). Já o valor de d_{min} permite comparar se adotar uma base conhecimento (QTSP) proporcionou encontrar melhores soluções.

4. RESULTADOS

4.1 Análise Preliminar

A análise preliminar compreende comparar as soluções (distância na rota) de acordo com a condição inicial de aprendizado adotada (Q0 ou QTSP). Para cada instância foi calculada a média da solução das 10 repetições simuladas. A Tabela 3 apresenta os resultados desta etapa.

Tabela 3. Média da solução (distância) ao longo do aprendizado para resolução do SOP, diferença percentual (D) entre os resultados de Q0 e QTSP e resultados do teste *t*. Existe diferença significativa entre as médias da soluções de Q0 e QTSP se $p \leq 0{,}05$.

Problema	**Q0**	**QTSP**	**D(%)**	***t***	***p***
br17.10	99,6	117,2	17,67	-3,20	0,01
br17.12	92,7	100,8	8,74	-2,14	0,06
ft53.1	19.054,3	10.003,9	-47,50	126,10	0,00
ft53.2	19.735,9	12.057,4	-38,91	138,95	0,00
ft53.3	19.583,6	16.173,2	-17,41	25,40	0,00
ft53.4	19.360,0	18.245,4	-5,76	26,56	0,00
kro124p.1	179.266,4	56.146,9	-68,68	789,13	0,00
kro124p.2	179.859,2	59.280,3	-67,04	432,24	0,00
kro124p.3	168.223,3	71.605,7	-57,43	221,92	0,00
kro124p.4	124.900,0	99.131,0	-20,63	67,09	0,00
p43.1	72.411,4	30.453,1	-57,94	117,94	0,00
p43.2	71.953,6	32.726,0	-54,52	101,90	0,00
p43.3	67.146,8	33.151,5	-50,63	84,45	0,00
p43.4	93.488,1	86.304,4	-7,68	25,96	0,00
	Média		-33,41		

Na Tabela 3, quanto mais negativo é o valor percentual de diferença entre os resultados, maior foi a eficiência de adotar a base de conhecimento (QTSP) em relação a Q0. Nota-se que, experimentos com QTSP apresentam em média uma solução de 33,41% menor que em simulações com Q0.

Em relação aos resultados do teste *t*, observou-se que das 14 instâncias do SOP, em 13 delas houve diferença significativa ($p \leq 0{,}05$) entre as médias das distâncias das rotas. Mais especificamente, das 13 que houveram diferenças, 12 delas apresentaram melhores resultados com a utilização da base de conhecimento (QTSP). Apenas para uma instância (br17.10) houve desvantagem na adoção da transferência de aprendizado.

4.2 Análise do Tempo Computacional

A Tabela 4 apresenta as médias de tempos computacionais das simulações dos problemas SOP de acordo com matriz de aprendizado inicial (Q0 ou QTSP), a respectiva diferença percentual e os resultados do teste *t*. Novamente, assim como descrito na Tabela 3, quanto mais negativo é o valor percentual da diferença entre os tempos computacionais de simulação, maior foi a eficiência de adotar a base de conhecimento (QTSP) em relação a Q0. Constata-se que experimentos com QTSP apresentam em média um tempo computacional de 29,18% menor que em simulações com Q0.

Tabela 4. Médias de tempos computacionais (em segundos) para resolução do SOP, diferença percentual (D) entre os resultados de Q0 e QTSP e resultados do teste *t*. Existe diferença significativa entre as médias dos tempos computacionais de Q0 e QTSP se $p \leq 0{,}05$.

Problema	**Q0**	**QTSP**	**D(%)**	***t***	***p***
br17.10	0,33	0,22	-33,33	3,24	0,01
br17.12	0,30	0,22	-26,67	9,62	0,00
ft53.1	0,65	0,59	-9,23	2,87	0,01
ft53.2	0,84	0,66	-21,43	9,95	0,00
ft53.3	1,79	0,89	-50,28	60,65	0,00
ft53.4	2,62	1,27	-51,53	73,37	0,00
kro124p.1	1,63	1,15	-29,45	10,42	0,00
kro124p.2	2,12	1,12	-47,17	73,63	0,00
kro124p.3	3,36	1,57	-53,27	57,10	0,00
kro124p.4	7,61	2,88	-62,16	158,77	0,00
p43.1	0,55	0,59	7,27	-1,49	0,15
p43.2	0,79	0,72	-8,86	4,53	0,00
p43.3	1,13	1,08	-4,42	2,73	0,01
p43.4	1,84	1,51	-17,93	27,04	0,00
	Média		-29,18		

De acordo com o teste *t*, em 13 instâncias (das 14 analisadas), a média do tempo de execução foi menor ao adotar a base de conhecimento ($p \leq 0{,}05$).

4.3 Análise Gráfica

Nesta etapa, são analisadas os gráficos de evolução do aprendizado (distância calculada ao longo dos episódios) para o seguintes problemas SOP: kro124p.1, kro124p.2, kro124p.3 e kro124p.4. Esses quatro problemas são baseados em kro124p (TSP) e os dados entre essas instâncias variam em função do número e a disposição das restrições de precedências entre os nós (ver Tabelas 1 e 2).

Nas Figuras 1 a 4 é possível observar que as curvas de aprendizado de QTSP iniciam os episódios mais próximos do valor ótimo da instância em relação as simulações com Q0. Isso fica mais evidente, quando compara-se os valores de d_0. Em todos os gráficos apresentados, a diferença entre a solução no episódio inicial de QTSP e Q0 foi maior que $0{,}8 \times 10^5$ (80 mil) unidades de distância. Nesse sentido, detaca-se a relevância na tranferência de conhecimento entre os domínios TSP e SOP nos episódios inicias, favorecendo a acelereção do aprendizado.

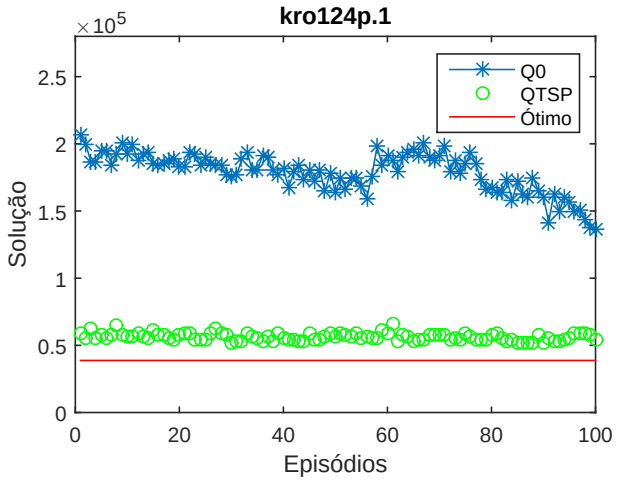

Figura 1. Curvas de aprendizado para a instância kro124p.1 (sem transferência de aprendizado - QO e com transferência de aprendizado - QTSP) e linha de valor ótimo da TSPLIB (38762). Medidas de desempenho para Q0 ($d_0 = 206824$, $d_f = 136261$ e $d_{min} = 136261$) e QTSP ($d_0 = 58497$, $d_f = 54649$ e $d_{min} = 52029$).

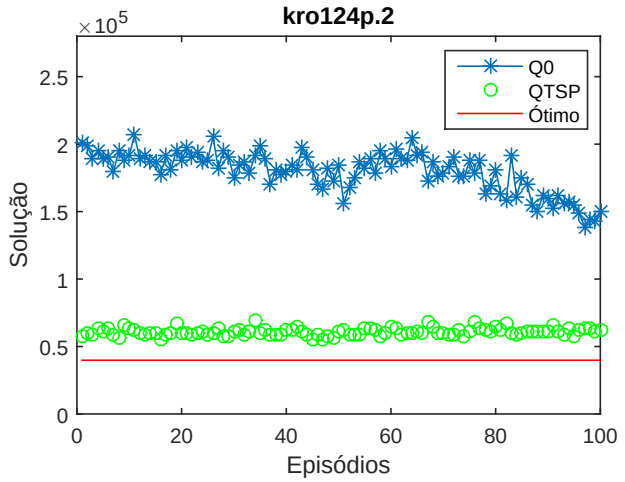

Figura 2. Curvas de aprendizado para a instância kro124p.2 (sem transferência de aprendizado - QO e com transferência de aprendizado - QTSP) e linha de valor ótimo da TSPLIB (39841). Medidas de desempenho para Q0 ($d_0 = 201214$, $d_f = 150206$ e $d_{min} = 138704$) e QTSP ($d_0 = 57230$, $d_f = 62778$ e $d_{min} = 55334$).

Além disso, os gráficos das Figuras 1 a 3 também retratam diferenças entre os valores de d_f para Q0 e QTSP. Para as instâncias kro124p.1, kro124p.2 e kro124p.3, adotar a transferência de aprendizado resultou em melhores resultados no último episódio da simulação. Na pior situação (experimentos do problema kro124p.4), a curva de aprendizado de Q0 ainda necesssitou de cerca de 50 episódios para alcançar resultados próximos aos apresentados pelas simulações adotando a base de conhecimento do TSP.

Por fim, destaca-se que para os gráficos (Figuras 1 a 4), adotar transferência de aprendizado entre os domínios TSP e SOP resultou em calcular melhores soluções durante

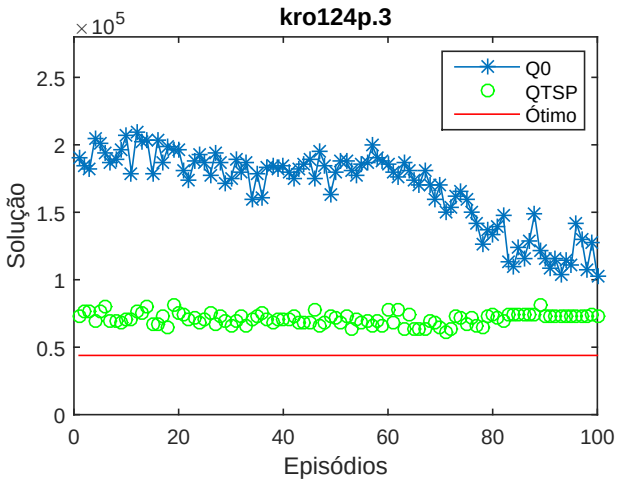

Figura 3. Curvas de aprendizado para a instância kro124p.3 (sem transferência de aprendizado - QO e com transferência de aprendizado - QTSP) e linha de valor ótimo da TSPLIB (43904). Medidas de desempenho para Q0 ($d_0 = 190308$, $d_f = 102856$ e $d_{min} = 102856$) e QTSP ($d_0 = 73178$, $d_f = 73407$ e $d_{min} = 61146$).

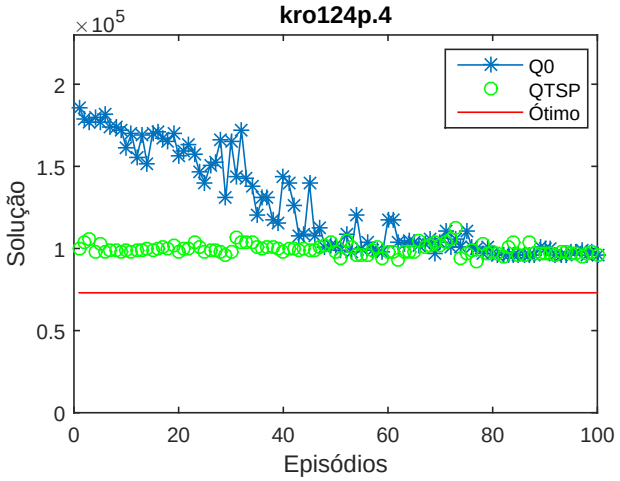

Figura 4. Curvas de aprendizado para a instância kro124p.4 (sem transferência de aprendizado - QO e com transferência de aprendizado - QTSP) e linha de valor ótimo da TSPLIB (73021). Medidas de desempenho para Q0 ($d_0 = 186028$, $d_f = 96283$ e $d_{min} = 96139$) e QTSP ($d_0 = 99903$, $d_f = 95602$ e $d_{min} = 92542$).

o aprendizado. Por exemplo, para a instância kro124p.1, $d_{min} = 52029$ ao adotar QTSP, enquanto que, para o aprendizado sem conhecimento prévio, o resultado foi de $d_{min} = 136261$.

5. CONCLUSÃO

O objetivo deste trabalho foi propor e avaliar a eficiência de uma metodologia para a transferência de AR entre dois domínios de otimização combinatória: TSP e SOP. Em termos de análise, foram avaliados os efeitos da utilização base conhecimento (QTSP) no domínio objetivo (SOP).

Os resultados obtidos a partir dos testes estatísticos, apontam que, em geral, adotar a transferência de aprendizado proporcionou o cálculo de menores rotas nos problemas (TSPLIB) do SOP. Além disso, em 13 instâncias, das 14 simuladas, o tempo médio computacional foi menor nos experimentos com a adoção da base QTSP. A análise gráfica permitiu avaliar as diferenças de comportamento nas curvas de aprendizado quando se utiliza ou não a matriz de transferência de AR.

Em trabalhos futuros, espera-se analisar a transferência de aprendizado de hiperparâmetros entre os domínios: α, γ, e função de reforço (Brazdil et al., 2009; Hutter et al., 2019; Ottoni et al., 2018, 2020). Além disso, será proposta uma estrutura de auto aprendizado de máquina (AutoML) (Hutter et al., 2019) para a realização de *Transfer Learning* entre os domínios TSP e SOP.

AGRADECIMENTOS

Agradecemos à CAPES, CNPq/INERGE, FAPEMIG, UFRB e UFSJ (Edital nº 001/2019/Reitoria).

REFERÊNCIAS

Alipour, M.M., Razavi, S.N., Feizi Derakhshi, M.R., and Balafar, M.A. (2018). A hybrid algorithm using a genetic algorithm and multiagent reinforcement learning heuristic to solve the traveling salesman problem. *Neural Computing and Applications*, 30(9), 2935–2951.

Arnekvist, I., Kragic, D., and Stork, J.A. (2019). Vpe: Variational policy embedding for transfer reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, 36–42.

Bianchi, R.A.C., Ribeiro, C.H.C., and Costa, A.H.R. (2009). On the relation between Ant Colony Optimization and Heuristically Accelerated Reinforcement Learning. *1st International Workshop on Hybrid Control of Autonomous System*, 49–55.

Bodin, L., Golden, B., Assad, A., and Ball, M. (1983). Routing and Scheduling of Vehicles and Crews – The State of the Art. *Computers and Operations Research*, 10(2), 63–211.

Brazdil, P., Carrier, C.G., Soares, C., and Vilalta, R. (2009). *Metalearning: Applications to data mining*. Springer Science & Business Media.

Cai, L., Sun, Q., Xu, T., Ma, Y., and Chen, Z. (2020). Multi-auv collaborative target recognition based on transfer-reinforcement learning. *IEEE Access*, 8, 39273–39284.

Carroll, J.L. and Peterson, T. (2002). Fixed vs. dynamic sub-transfer in reinforcement learning. In *Proceedings of the International Conference on Machine Learning and Applications*, 3–8.

Costa, M.L., Padilha, C.A.A., Melo, J.D., and Neto, A.D.D. (2016). Hierarchical reinforcement learning and parallel computing applied to the k-server problem. *IEEE Latin America Transactions*, 14(10), 4351–4357.

Da Silva, F. and Reali Costa, A. (2019). A survey on transfer learning for multiagent reinforcement learning systems. *Journal of Artificial Intelligence Research*, 64, 645–703.

Da Silva, F., Warnell, G., Costa, A., and Stone, P. (2020). Agents teaching agents: a survey on inter-agent transfer learning. *Autonomous Agents and Multi-Agent Systems*, 34(1).

Escudero, L. (1988). An inexact algorithm for the sequential ordering problem. *European Journal of Operational Research*, 37(2), 236–249.

Fernández, F. and Veloso, M. (2006). Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 720–727.

Gambardella, L.M. and Dorigo, M. (1995). Ant-Q: A reinforcement learning approach to the traveling salesman problem. *Proceedings of the 12th International Conference on Machine Learning*, 252–260.

Gambardella, L.M. and Dorigo, M. (2000). An ant colony system hybridized with a new local search for the sequential ordering problem. *INFORMS Journal on Computing*, 12(3), 237–255.

Hou, Y., Ong, Y., Feng, L., and Zurada, J.M. (2017). An evolutionary transfer reinforcement learning framework for multiagent systems. *IEEE Transactions on Evolutionary Computation*, 21(4), 601–615.

Hutter, F., Kotthoff, L., and Vanschoren, J. (eds.) (2019). *Automated Machine Learning: Methods, Systems, Challenges*. Springer. In press, available at http://automl.org/book.

Lazaric, A. and Restelli, M. (2011). Transfer from multiple mdps. In *Advances in Neural Information Processing Systems*, 1746–1754.

Letchford, A.N. and Salazar-González, J.J. (2016). Stronger multi-commodity flow formulations of the (capacitated) sequential ordering problem. *European Journal of Operational Research*, 251(1), 74 – 84.

Lima Júnior, F.C., Neto, A.D.D., and Melo, J.D. (2010). *Traveling Salesman Problem, Theory and Applications*, chapter Hybrid Metaheuristics Using Reinforcement Learning Applied to Salesman Traveling Problem, 213–236. InTech.

Lins, R.A.S., Dória, A.D.N., and de Melo, J.D. (2019). Deep reinforcement learning applied to the k-server problem. *Expert Systems with Applications*, 135, 212–218.

Lopes, R.H.C. (2011). *Kolmogorov-Smirnov Test*, 718–720. Springer Berlin Heidelberg, Berlin, Heidelberg.

Montgomery, D.C. (2017). *Design and analysis of experiments*. New York: John Wiley & Sons., 9th edition.

Ottoni, A.L.C., Nepomuceno, E.G., and Oliveira, M.S. (2017). Análise do desempenho do aprendizado por reforço na solução do problema da mochila multidimensional. *Revista Brasileira de Computação Aplicada*, 9(3), 56–70.

Ottoni, A.L.C., Nepomuceno, E.G., and de Oliveira, M.S. (2018). A response surface model approach to parameter estimation of reinforcement learning for the travelling salesman problem. *Journal of Control, Automation and Electrical Systems*, 29(3), 350–359.

Ottoni, A.L.C., Nepomuceno, E.G., de Oliveira, M.S., and de Oliveira, D.C.R. (2020). Tuning of reinforcement learning parameters applied to sop using the scott–knott method. *Soft Computing*, 24, 4441–4453.

Peterson, T.S., Owens, N.E., and Carroll, J.L. (2001). Towards automatic shaping in robot navigation. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*,

volume 1, 517–522 vol.1.

Reinelt, G. (1991). TSPLIB - A Traveling Salesman Problem Library. *ORSA Journal on Computing*, 3(4), 376–384.

Russell, S.J. and Norving, P. (2013). *Artificial Intelligence*. Campus, 3st ed.

Silva, M.A.L., de Souza, S.R., Souza, M.J.F., and Bazzan, A.L.C. (2019). A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems. *Expert Systems with Applications*, 131, 148–171.

Skinderowicz, R. (2017). An improved ant colony system for the sequential ordering problem. *Computers & Operations Research*, 86, 1 – 17.

Sutton, R. and Barto, A. (2018). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 2nd edition.

Taylor, M.E. and Stone, P. (2009). Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul), 1633–1685.

Tommasino, P., Caligiore, D., Mirolli, M., and Baldassarre, G. (2019). A reinforcement learning architecture that transfers knowledge between skills when solving multiple tasks. *IEEE Transactions on Cognitive and Developmental Systems*, 11(2), 292–317.

Wang, H., Fan, S., Song, J., Gao, Y., and Chen, X. (2014). Reinforcement learning transfer based on subgoal discovery and subtask similarity. *IEEE/CAA Journal of Automatica Sinica*, 1(3), 257–266.

Watkins, C.J. and Dayan, P. (1992). Technical note Q-learning. *Machine Learning*, 8(3), 279–292.