



- $Q_{t+1}(s, a)$ tu \rightarrow o tr \rightarrow pr \rightarrow o no nst nt \rightarrow t + 1 pr \rightarrow cu \rightarrow o \rightarrow a no \rightarrow st o s
- $r(s, a)$ o r \rightarrow co p \rightarrow ns \rightarrow t p r \rightarrow cu \rightarrow o \rightarrow a no \rightarrow st o s
- $\max_a Q(s')$ ut \rightarrow s' ou s \rightarrow o v or \rightarrow on tr \rightarrow pr \rightarrow on n p o novo \rightarrow st o s'
- t \rightarrow pr \rightarrow o
- o r \rightarrow tor \rightarrow s \rightarrow conto
- $s \rightarrow s' \rightarrow a \rightarrow a' \rightarrow s_{t+1} \rightarrow a' \rightarrow a_{t+1}$
- A or t o A A utton \rightarrow B rto

- 1) Definir o conjunto de testes a serem realizados. Nesse caso os testes são todos os casos que o código resolve, incluindo os casos de teste.
- 2) Definir o conjunto de testes que o programa C resolve, o conjunto de casos em que o programa resolve os testes. O programa resolve os testes que são os testes que são resolvidos pelo programa. (Anexo 2)
- 3) Definir os valores dos parâmetros de teste \times a serem utilizados nos casos de teste. Nesse caso os parâmetros de teste são os parâmetros de teste. Assim, quanto maior o número de parâmetros de teste, maior o número de testes que são resolvidos pelo programa. (Anexo 4)
- 4) Aplicar os testes de programação ao programa. A análise ou desenvolvimento do código de programação ou uso do software Matlab para a realização dos testes.

5. METODOLOGIA DE DEFINIÇÃO DOS PARÂMETROS

Neste seção propõe-se a metodologia de definição dos parâmetros de teste de programação. Essa metodologia é baseada na metodologia de teste de programação.

- 1) Definir os critérios de teste de programação:
 - D: Distância entre os parâmetros de teste de programação.
 - M: Método de teste de programação.
 - Critérios de teste de programação:
 - M: Método de teste de programação.
 - M: Método de teste de programação.
- 2) Análise de teste de programação de teste de programação:
 - Definir o conjunto de valores para os testes de programação. (Anexo 2)
 - Definir o conjunto de valores para os testes de programação. (Anexo 2)
- 3) Definir o método de teste de programação: **greedy** (Anexo 2)
 - Esse método de teste de programação é baseado no método de teste de programação.
- 4) Análise dos critérios de teste de programação de teste de programação.

Tabela 1: Problemas de TSP e suas soluções ótimas conhecidas

Problema	Cidades	Solução Ótima Conhecida
Br		
Ftv	4	2
Ftv 44	4	
Ftv 4		

Tabela 2: Descrição dos Experimentos em 20 e 40 cidades

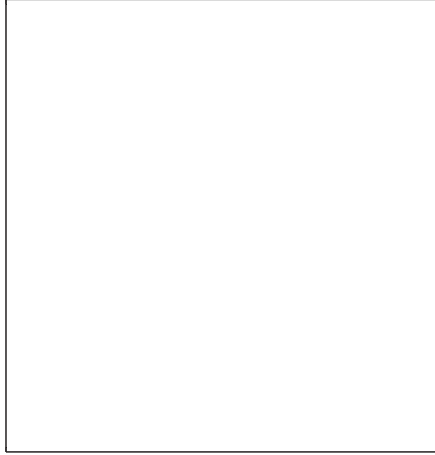
	Quantidade	Valores
		0000 000040000 0000
		0000 000040000 0000
		00
Condições	$\times \times 4$	
população		
população por geração	000	
população por geração	$\times 000$ 000	
Elitismo	4×20	
Elitismo	20×000	
Episódios	20000	

7. ANÁLISE DOS RESULTADOS

Os resultados dos experimentos pontuam a eficiência dos algoritmos propostos no teste proposto. A análise dos resultados demonstra que o algoritmo proposto apresenta desempenho superior em relação aos algoritmos de referência. Os resultados mostram que o algoritmo proposto consegue encontrar soluções melhores em menos iterações, o que indica uma maior eficiência computacional. A análise dos resultados também demonstra que o algoritmo proposto é capaz de encontrar soluções melhores em menos iterações, o que indica uma maior eficiência computacional.

Tabela 3: Comparação dos resultados dos algoritmos Q-learning e SARSA

Problema	Critério	Q-learning	SARSA
		Valor	Valor



Assim, apresenta-se os critérios de seleção para os problemas práticos constantes no curso de graduação em Engenharia de Mecânica.

4.4.4. Descrição dos problemas práticos propostos para o curso de graduação em Engenharia de Mecânica.

Problema	Algoritmo		
----------	-----------	--	--

Br

NOMENCLATURA

Esquema no formato de transferência

A Aprendizagem por reforço

CA Problema de otimização

CA Problema de otimização Assíncrono

Letras gregas

α taxa de aprendizado

γ fator de desconto

REFERÊNCIAS

- Applegate, D., B. Chazelle, E. Chortsos, e J. Cooper. 2000. The Traveling Salesman Problem: A Computational Study. Princeton University Press, Princeton.
- Bonatti, P. A. C. 2004. Uso de Heurística para a aceleração do aprendizado por reforço. Dissertação de Doutorado, Escola