

Resultados Obtidos

[PROCURA CEGA] Procura Profundidade Primeiro em Árvore (PPPA)

A PPPA excedeu o limite máximo de tempo de execução disponível (10 segundos) em 4 instâncias de um total de 9.

Nas instâncias em que não excedeu o tempo máximo disponível a PPPA teve um desempenho relativamente bom mas pior que o das procuras informadas (PG e PAA).

Nas instâncias em que não excedeu o tempo máximo disponível a PPPA gerou/expandiu um número relativamente baixo de nós mas superior aos gerados/expandidos pelas procuras informadas (PG e PAA).

[PROCURA CEGA] Procura Largura Primeiro em Árvore (PLPA)

A PLPA excedeu o limite máximo de tempo de execução disponível (10 segundos) em 5 instâncias de um total de 9.

Nas instâncias em que não excedeu o tempo máximo disponível a PLPA teve um desempenho relativamente mau em comparação com todas as outras procuras.

Nas instâncias em que não excedeu o tempo máximo disponível a PLPA gerou/expandiu um número relativamente alto de nós quando comparado às outras procuras.

[PROCURA INFORMADA] Procura Gananciosa (PG)

A PG excedeu o limite máximo de tempo de execução disponível (10 segundos) em 3 instâncias de um total de 9.

Nas instâncias em que não excedeu o tempo máximo disponível a PG teve um desempenho muito bom não superando apenas a PAA.

Nas instâncias em que não excedeu o tempo máximo disponível a PG gerou/expandiu um número muito baixo de nós ficando atrás apenas da PAA.

[PROCURA INFORMADA] Procura A* (PAA)

A PAA nunca excedeu o limite máximo de tempo de execução disponível.

Em todas as instâncias a PAA teve um desempenho excelente.

Em todas as instâncias a PAA gerou/expandiu um número muito baixo de nós.

Análise dos Resultados Obtidos

[PROCURA CEGA] Procura Profundidade Primeiro em Árvore (PPPA)

A PPPA não é completa. Como é uma procura em árvore podem existir nós repetidos. Pode dar-se o caso de existir solução e a PPPA ficar num ciclo infinito (se escolher ações que se cancelam mutuamente - no nosso caso "X U" + "X D" ou "X L" + "X R", por exemplo).

A PPPA é muito pouco eficiente porque não tem nenhum mecanismo que evite a repetição de estados, gerando muitos nós. Além disso não tem conhecimento do domínio do problema portanto avança "cegamente".

A PPPA não é ótima. Não dá nenhuma garantia de, existindo solução, devolver a solução ótima (com menos jogadas).

[PROCURA CEGA] Procura Largura Primeiro em Árvore (PLPA)

A PLPA é completa. Ao contrário da PPPA se a primeira ação não levar a um estado solução a PLPA explora também a segunda, a terceira, até à enésima. Logo, se existir solução a PLPA encontrá-la-á.

A PLPA é muito pouco eficiente porque não tem nenhum mecanismo que evite a repetição de estados, gerando muitos nós. Além disso não tem conhecimento do domínio do problema portanto avança “cegamente”.

A PLPA é ótima. Começando no estado inicial, a PLPA explora todos os caminhos de comprimento 1 à procura de um estado solução, depois todos os caminhos de comprimento 2 e por aí em diante. Logo, se existir solução, a PLPA devolverá uma solução com o menor comprimento possível (no nosso caso menor número de jogadas) - que é uma solução ótima.

[PROCURA INFORMADA] Procura Gananciosa (PG)

A PG é não completa. Ao contrário da PAA a procura gananciosa só tem em conta o custo estimado de chegar ao objetivo. Logo, pode existir solução e a procura gananciosa ficar em ciclo em torno de uma solução “aparente” (porque de menor custo estimado).

A PG é eficiente porque não só tem um mecanismo que evita a repetição de estados mas também tem conhecimento (ainda que muito local/instantâneo) do domínio do problema.

A PG não é ótima. Como só tem conhecimento do domínio de um determinado instante só encontra “ótimos locais”. Precisaria de ter acesso à *big picture* como a PAA para poder encontrar ótimos globais (assumindo uma heurística admissível).

[PROCURA INFORMADA] Procura A* (PAA)

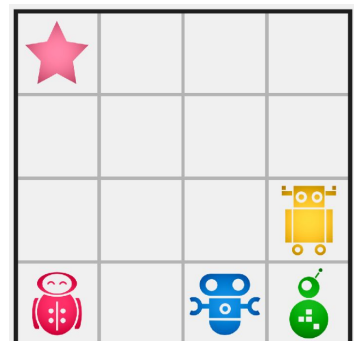
A PAA é completa. Como o espaço de estados é finito e a PAA é uma procura em grafo (logo não repete estados) é garantido que, existindo solução, a PAA devolverá uma solução.

A PAA é muito eficiente porque tem um mecanismo que evita a repetição de estados. Além disso tem conhecimento do domínio do problema e avalia o tradeoff entre o custo do caminho percorrido até um estado e o custo estimado de chegar ao objetivo.

No nosso caso, a PAA não é ótima porque a nossa heurística não é admissível (explicação mais detalhada abaixo).

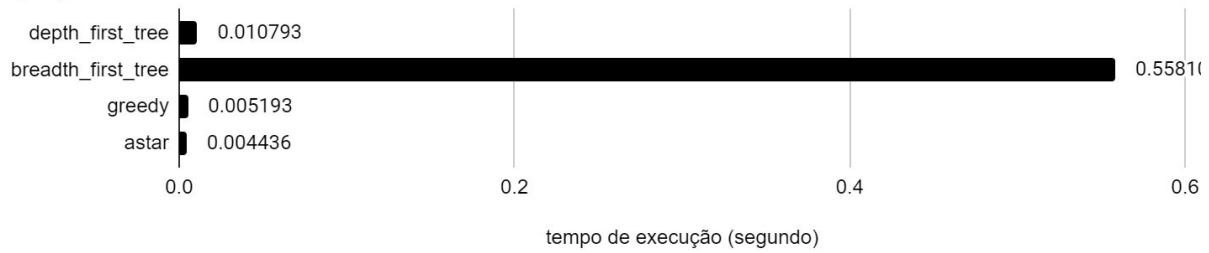
[HEURÍSTICA] Distância de Manhattan entre Posição do Robot-Objetivo e Posição-Objetivo

Veja-se o caso representado na imagem à direita. O número mínimo de movimentos para chegar a um estado solução é 1 (R U). A distância de manhattan entre o robot vermelho e o objetivo é $3+0=3$. Como a estimativa da heurística é superior ao valor real podemos afirmar que a nossa heurística é não admissível. (Como é não admissível então trivialmente a nossa heurística também é não consistente.)

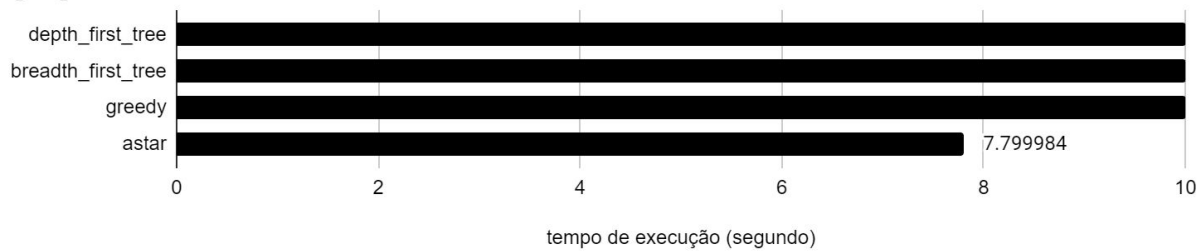


Tempo de Execução(s) em função do Algoritmo de Procura (o limite superior do tempo de execução é 10 segundos)

[01]



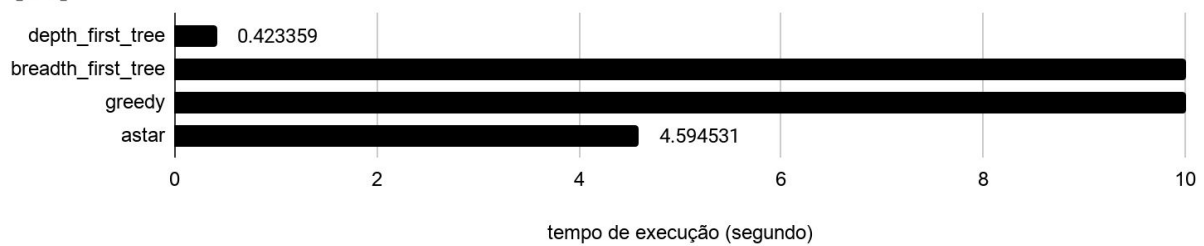
[02]



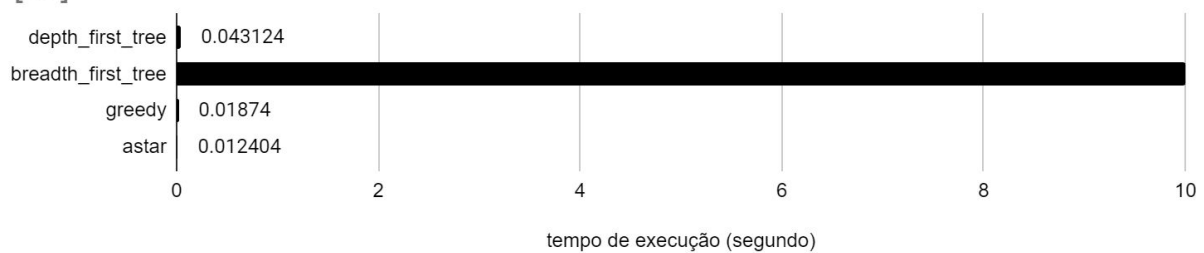
[03]



[04]

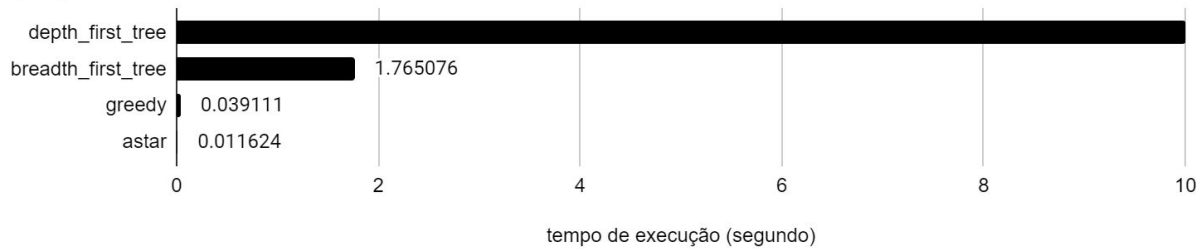


[05]



Tempo de Execução(s) em função do Algoritmo de Procura (o limite superior do tempo de execução é 10 segundos)

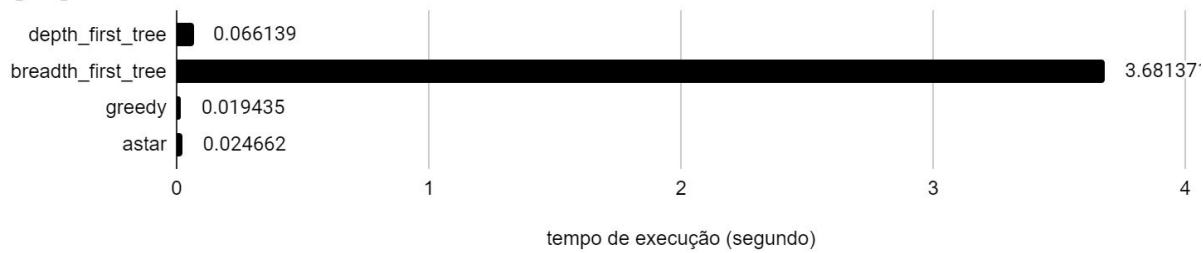
[06]



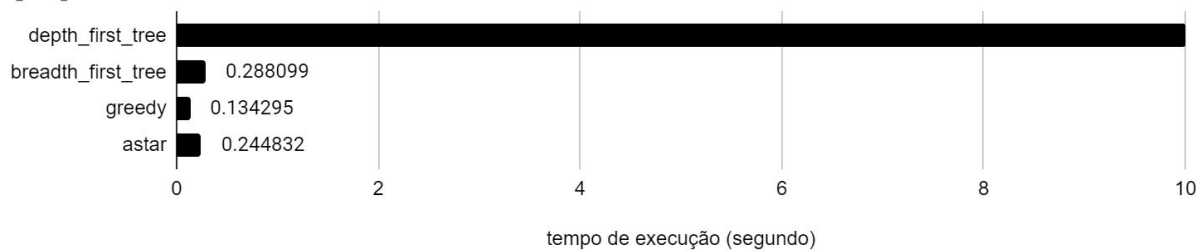
[07]



[08]

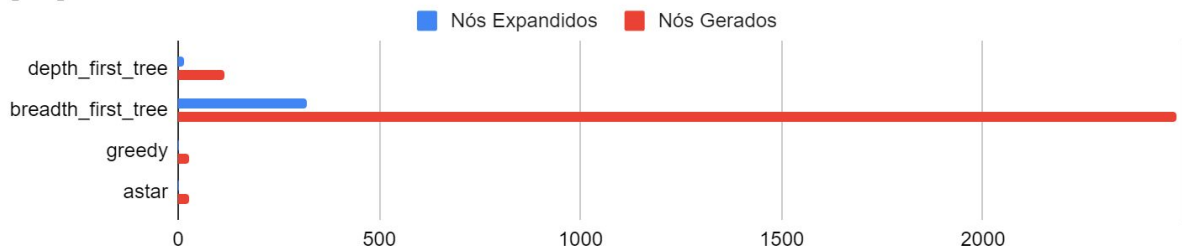


[09]

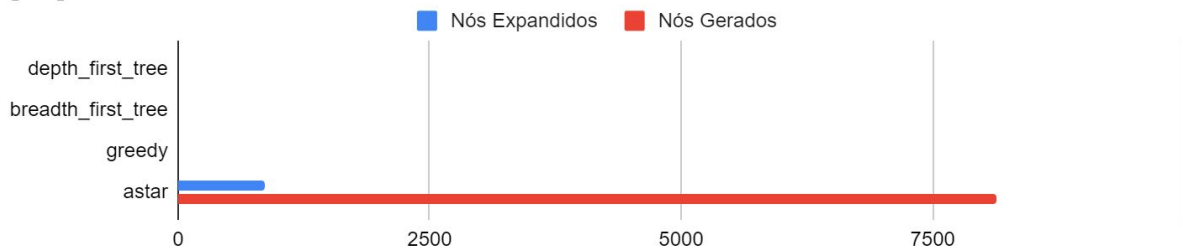


Nº de Nós Expandidos/Gerados em função do Algoritmo de Procura (não têm nº de nós as buscas cujo tempo de execução chegou ao limite)

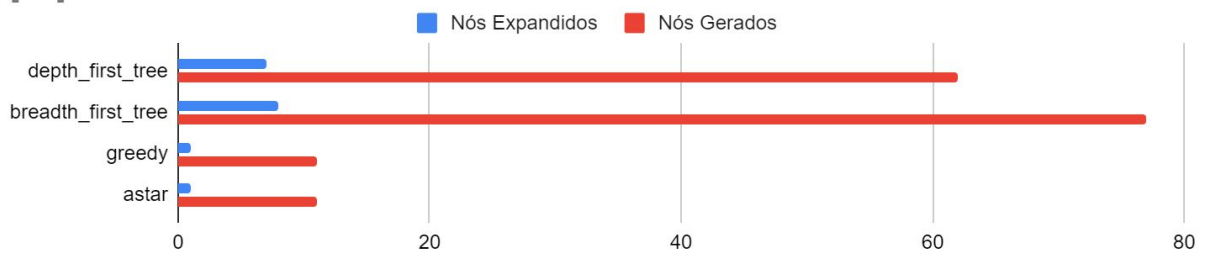
[01]



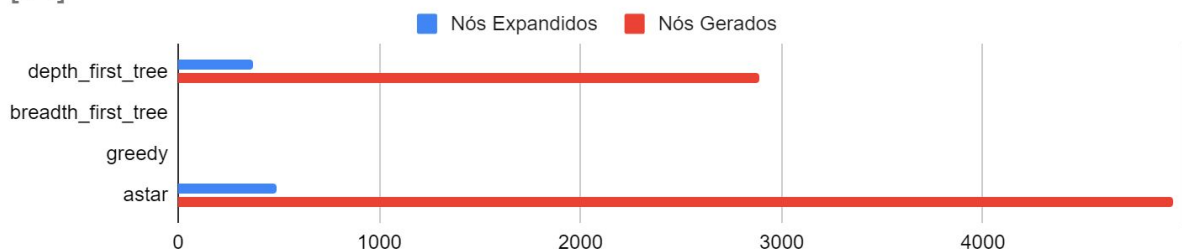
[02]



[03]



[04]



[05]

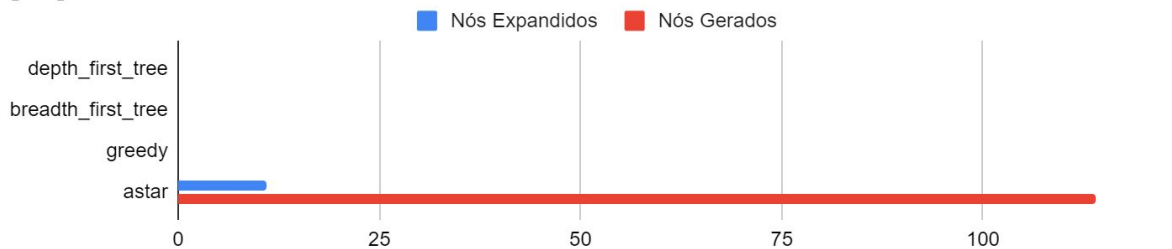


Nº de Nós Expandidos/Gerados em função do Algoritmo de Procura (não têm nº de nós as buscas cujo tempo de execução chegou ao limite)

[06]



[07]



[08]



[09]

