

Tarefa 04: Busca de Custo Uniforme x A*

1) Faça um Explique em linguagem natural de h1 juntamente com o código de h1.

Responda: a heurística h1 é admissível e consistente? Justifique.

- `float hn = (float) Math.sqrt(Math.pow((noAtual.getState().getLin() - problema.estObj.getLin()), 2) + Math.pow((noAtual.getState().getCol()-problema.estObj.getCol()), 2));`
- A heurística h1 é a distância euclidiana do nó atual até o objetivo. Ela é admissível pois a distância em linha reta sempre será menor ou igual a distância real de um nó a outro, pois a menor distância entre dois pontos é uma linha reta. Ela é consistente pois respeita a desigualdade triangular quase que perfeitamente, pois como o custo em linha reta é o menor possível, o custo para qualquer outro nó formará um triângulo e assim $h(n)$ sempre será menor que $c(n,a,n') + h(n')$.

2) Explique em linguagem natural de h2 juntamente com o código de h2.

Responda: a heurística h2 é admissível e consistente? Justifique.

- `hn = 0;`
- A heurística h2 é 0 para todos os pontos. Ela é admissível pois 0 sempre será menor ou igual do que a distância real, pois não há distâncias negativas. Além disso como todos os valores são 0 não há como ela desrespeitar a desigualdade triangular, pois como $h(n)=0$, temos que $h(n) < c(n,a,n') + h(n') \Rightarrow 0 < c(n,a,n') + 0$.

3) Qual heurística é dominante (h1 ou h2)? Explique o que é dominância e o efeito da dominância.

- Nesse caso é h1, pois como as duas heurísticas são admissíveis e h2 é sempre 0 e h1 sempre maior ou igual a zero, pela definição de dominância temos que:
 - Se $h1(n) \geq h2(n)$ para todo n (ambas admissíveis)
 - Então h1 domina h2
- O efeito da dominância é h1 expandir menos nós que h2

4) Faça uma tabela comparativa das três estratégias contendo os seguintes elementos:

a) Complexidade temporal medida pelo número de nós gerados: são todos os nós gerados, inclusive aqueles que não devem ser incluídos na árvore pelos motivos abaixo:

- i) é um nó que representa um estado já explorado (ct_ja_explorados)
- ii) é um nó que representa um estado que já está na fronteira com um f(n) pior ()
- iii) Ao final da execução, calcule o total e nós gerados como sendo:
total de nós na árvore + ct_ja_explorados + ct_descartados_front

b) Complexidade espacial: a cada iteração do algoritmo, guarde o número total de nós na árvore (observe que a árvore contém os já explorados e

a fronteira). Ao final da execução, o cálculo de desempenho temporal será dado pelo maior número de nós encontrado.

c) A solução encontrada

d) O custo da solução encontrada

Estratégia	Nós na árvore	ct_ já explorados	ct_ descarta dos_ front	Total nós gerados	Máx. núm. nós na memória	Solução	Custo
Custo Unif.	414	60	0	474	133	Sol	12.5
A* com h1	39	40	16	95	30	Sol	12.5
A* com h2	414	60	0	474	133	Sol	12.5

Sol: N, N, N, NE, L, L, L, L, NE, NE, L

5) Analise a tabela anterior em relação a verificação das propriedades de otimalidade e completude de cada implementação e, também, em relação aos resultados de desempenho temporal e espacial (se comparativamente estão dentro do esperado – utilize elementos como informação e dominância)?

- Otimalidade: na tabela os três custos são iguais como esperado, pois o algoritmo A* sempre encontra a solução ótima pois as heurísticas são admissíveis e consistentes e o algoritmo Custo Uniforme também pois expande os nós baseado em $g(n)$.
- Completude: todos os algoritmos são completos pois o máximo de sucessores de um nó é sempre finito.
- Desempenho espacial: o número máximo de nós na memória para o Custo Uniforme e o A* com h2 é o mesmo, mas o A* com h1, já que a informação da distância euclidiana corresponde à heurística dominante, explora menos nós que as outras soluções.
- Desempenho temporal: pelo mesmo motivo do item anterior, A* com h1 possui o melhor desempenho temporal.

6) Ao definir o estado objetivo como (8,1) – uma posição que contém uma parede – responda:

a) Qual a solução obtida pelos algoritmos?

- Nenhum dos algoritmos encontra solução.

b) Esta solução é compatível com a propriedade de completude?

- Visto que colocamos o objetivo dentro de uma parede, as estratégias nunca irão encontrar uma solução. Mas como o número total de nós é finito com $f(n) \leq C^*$, então os algoritmos conseguem chegar a conclusão que não existe um caminho para o objetivo e com isso nosso resultado é compatível com a propriedade de completude.

c) Quantos estados as estratégias exploraram?

- Todas as estratégias exploram todos os estados possíveis (53 estados).