

Inteligência Artificial e Sistemas de Decisão

Projeto 1

Realizado por:

Henrique Ferreira, n.º 78654;

Manuel Rosa, n.º 78679

Ano letivo 2017-18

1.º Semestre

1 Descrição da representação do problema

Como primeira abordagem ao problema, definiu-se o conceito de estado para o caso em estudo, nomeadamente as variáveis que o definiriam. Neste sentido, assumiu-se que o estado é constituído por dois campos: o primeiro refere os componentes que já se encontram no espaço; o segundo indica o próximo lançamento a efetuar. Desta forma, o estado inicial é definido pela inexistência de componentes no espaço e pelo primeiro lançamento por ordem cronológica. Do mesmo modo, o *goal state* só é atingido quando todos os componentes estiverem no espaço, independentemente de qual seja o lançamento seguinte.

No que diz respeito ao *path cost*, este só é incluído no node, sendo continuamente somado ao longo de um certo caminho com o auxílio da função *launch_cost*. Basicamente, calcula o custo efetivo do lançamento de acordo com a massa dos componentes a enviar ($c_f + c_w \times comp_{weight}$).

Por fim, os operadores consistem no leque de ações possíveis realizar num determinado estado. Mais precisamente, cada ação consiste no lançamento de uma entre as diferentes combinações possíveis de componentes por enviar e que respeitam as condições de adjacência e *payload* máximo.

2 Algoritmo de procura implementado

O método de procura implementado para a procura não informada foi o **Uniform-cost search**, sendo expandido com o auxílio de uma função heurística para **A-star search** (A^*) no âmbito de procura informada.

Note que, o principal objetivo do trabalho consiste na determinação de uma sequência de lançamentos, por forma a minimizar o custo total das operações. Assim, faz todo o sentido usar o UCS, uma vez que expande sempre o nó da *open list* com o menor *path cost* - $g(n)$. Relativamente à procura informada, utilizou-se o algoritmo A^* , que resulta de uma expansão do UCS. Na realidade, o critério de escolha do próximo nó baseia-se no valor de $f(n) = g(n) + h(n)$, isto é, na soma do *path cost* e do valor de uma heurística $h(n)$.

3 Função heurística utilizada

Nesta secção, descreve-se e explica-se a implementação da função heurística utilizada, com base em diversos princípios. Ainda, pretende-se demonstrar que a mesma é consistente, condição necessária e suficiente para que a nossa solução seja ótima e completa. Por fim, analisa-se a qualidade da heurística e reflete-se sobre o seu desempenho.

Em primeiro lugar relaxam-se as restrições do problema. Considera-se que o peso a enviar é contínuo (*remaining weight*), ignorando assim a sua discretização devido ao peso de cada componente, e retira-se a restrição de apenas poderem ser enviados componentes adjacentes em cada lançamento, podendo então ser desprezada a sequência temporal dos lançamentos. Os lançamentos utilizados são escolhidos por ordem crescente do seu custo específico, dado por

$$C_{spe} = \frac{C_{fix} + C_{var} \times payload}{payload}.$$

O custo da heurística é calculado dentro de um ciclo enquanto houver peso a enviar ($remain_w > 0$). Em cada iteração do ciclo utiliza-se o lançamento com menor custo específico disponível, caso o peso a enviar seja maior ou igual à carga máxima que se pode enviar nesse lançamento soma-se à heurística o custo de um lançamento cheio, $h += C_{spe} \times payload$. Quando o peso a enviar é inferior à carga máxima do lançamento soma-se apenas o custo de enviar o peso em questão, $h += C_{spe} \times remain_w$, de realçar que este custo seria inferior ao real pois o custo fixo, tal como o nome indica, é constante para um lançamento e neste caso consideramos que quando o lançamento não está cheio apenas se paga um custo fixo proporcional à fração que se utiliza.

A heurística descrita é admissível pois o custo estimado é sempre inferior ou igual ao custo real até ao objetivo. Por vezes os lançamentos escolhidos não correspondem aos ótimos mas produzem um custo mais barato devido a apenas considerarem uma fração do custo fixo quando o lançamento não tem carga máxima. Mais ainda, a heurística é consistente pois, uma vez que nunca há introdução novos lançamentos disponíveis, a soma dos custos de um lançamento para um estado intermédio e desse estado intermédio para o estado final será sempre superior ao custo estimado.

3.1 Análise crítica e comentários

Como forma de análise faz-se a comparação entre a heurística implementada, outra heurística (disponível em anexo) e o método de procura não informado, no que diz respeito ao número de nós expandidos e ao critério denominado *Effective Branching Factor*. Com esse intuito utilizam-se cinco exemplos diferentes: *simple1*, *Teste2*, *mir*, *Teste4*, *Teste5* (dispostos em anexo).

A heurística 2 ordena separadamente em duas listas os lançamentos por custo fixo e por custo variável. Assim, para os lançamentos com o c_w mais baixo, somará à heurística somente o custo variável de envio, até que $remain_w = 0$. Para a lista ordenada de custos fixos, o procedimento é análogo, considerando que se o $remain_w$ for menor que o $payload$ é apenas adicionada uma fração do c_f .

Relativamente ao número de nós expandidos (ver figura 1a), verifica-se que para ambas as heurísticas existe uma melhoria significativa quando comparado com o método de procura não informado. De facto, para esse caso o número de nós expandidos foi de: 297, 70, 1220, 664, 6376 para *simple1*, *Teste2*, *mir*, *Teste4*, *Teste5*, respetivamente. Então, verifica-se a vantagem do método informado e da presença de uma heurística, desde que seja bem formulada. Um dos critérios mais importantes para avaliar a qualidade de uma heurística consiste no *Effective Branching Factor* - B^* . Na verdade, representa o *branching factor* que uma árvore uniforme de *depth* L teria, de modo a incluir o número total de nós expandidos $T + 1$. Note que L consiste na dimensão do caminho até ao goal state. Com base na equação 1 calcula-se este critério para ambas as heurísticas (ver 1b). Assim, quanto mais próximo o *Effective Branching Factor* estiver de 1, melhor será o desempenho ou a qualidade da heurística.

$$T + 1 = 1 + B^* + (B^*)^2 + \dots + (B^*)^L \quad (1)$$

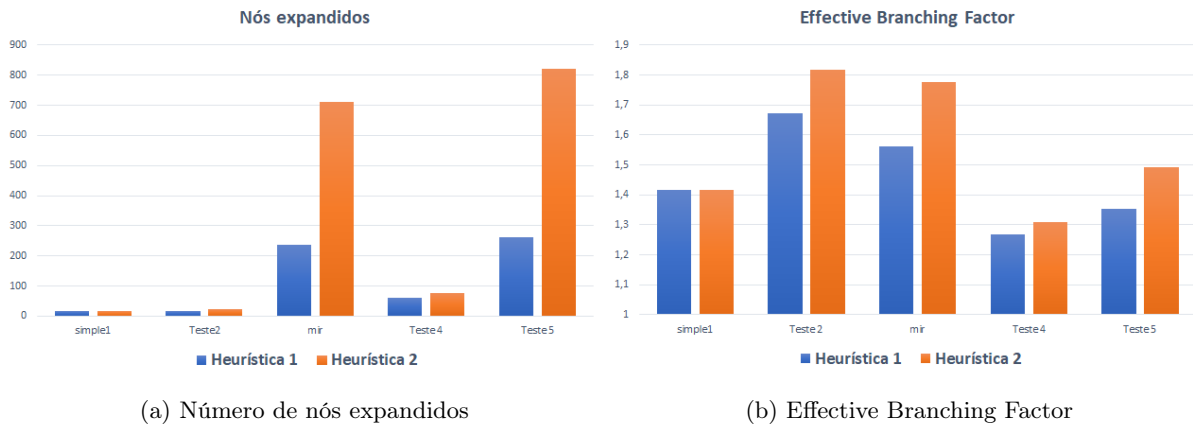


Figura 1: Comparação entre duas heurísticas

Por análise dos gráficos conclui-se que a heurística 1 possui uma melhor performance, ou seja, h_1 domina h_2 , daí a sua utilização no âmbito deste projeto.