Universidade Federal do Rio Grande do Sul

Instituto de Informática

INF05023 - Planejamento em Inteligência Artificial

**Lista de Exercícios Três**

Felipe Zorzo Pereira

João Vitor de Camargo

Porto Alegre, Junho de 2019

**Exercício 1**

* 1. Suponha que temos 3 buracos (A, B e C) um do lado do outro e apenas uma bola. As únicas ações possíveis são mover a bola para o buraco da direita (A é o buraco mais à esquerda, B é o do meio e C o mais à direita). A bola inicia no buraco A. O objetivo de nossa tarefa de planejamento Π1 é ter uma bola em cada um dos buracos, o que é obviamente impossível, já que possuímos apenas uma bola. Abaixo a descrição em STRIPS da tarefa de planejamento Π1, que é insolúvel.

V = {ATA, ATB, ATC}

I = {ATA}

G = {ATA, ATB, ATC}

A = {MOVEAB, MOVEBC}

* pre(MOVEAB) = {ATA}; add(MOVEAB) = {ATB}; del(MOVEAB) = {ATA}; cost(MOVEAB) = 1.
* pre(MOVEBC) = {ATB}; add(MOVEBC) = {ATC}; del(MOVEBC) = {ATB}; cost(MOVEBC) = 1.

A tarefa de planejamento relaxada, entretanto, não possui os efeitos de deleção que indicam que a bola foi retirada do buraco da esquerda. Com isso, ao movermos a bola para a direita, é como se, magicamente, aparecesse uma nova bola no buraco de onde acabamos de tirar a bola. Portanto, com a sequência de ações {MOVEAB, MOVEBC}, atingimos o estado objetivo {ATA, ATB, ATC} com custo 2, já que cada movimento tem custo unitário. A tarefa relaxada Π1+ está abaixo:

V = {ATA, ATB, ATC}

I = {ATA}

G = {ATA, ATB, ATC}

A = {MOVEAB, MOVEBC}

* pre(MOVEAB) = {ATA}; add(MOVEAB) = {ATB}; del(MOVEAB) = ∅; cost(MOVEAB) = 1.
* pre(MOVEBC) = {ATB}; add(MOVEBC) = {ATC}; del(MOVEBC) = ∅; cost(MOVEBC) = 1.
  1. Dizemos que um estado s' domina um estado s caso o conjunto das variáveis *true* em s esteja contido no conjunto das variáveis *true* em s'. Uma consequência de relaxarmos a tarefa de planejamento Π2 é que seus estados dominarão os seus respectivos estados na tarefa original. Isso ocorre pois a diferença entre as duas tarefas é que, em Π2+, não temos mais os efeitos de deleção. De acordo com o *relaxation lemma*: dada uma sequência π de operadores aplicáveis em s, a mesma sequência π+, mas com os operadores relaxados, é aplicável a s' (estado que domina s). Além disso, se π leva a um estado-objetivo a partir do estado s, então π+ leva a um estado objetivo a partir de s'. Portanto, é impossível que Π2 tenha uma solução de tamanho 2, mas Π2+ não tenha qualquer solução.
  2. Podemos usar o mesmo problema descrito no exercício i, mas adicionando um novo buraco D, à direita de C, que também deve estar ocupado ao fim de nossa tarefa. Além dessa nova variável D que aparece no conjunto de variáveis V e no *goal* G, temos um novo operador MOVECD, de custo 2, já que o buraco C é mais profundo e requer mais esforço para que consigamos retirar a bola dele. Com isso, a tarefa Π3 é:

V = {ATA, ATB, ATC, ATD}

I = {ATA}

G = {ATA, ATB, ATC, ATD}

A = {MOVEAB, MOVEBC, MOVECD}

* pre(MOVEAB) = {ATA}; add(MOVEAB) = {ATB}; del(MOVEAB) = {ATA}; cost(MOVEAB) = 1.
* pre(MOVEBC) = {ATB}; add(MOVEBC) = {ATC}; del(MOVEBC) = {ATB}; cost(MOVEBC) = 1.
* pre(MOVECD) = {ATC}; add(MOVECD) = {ATD}; del(MOVECD) = {ATC}; cost(MOVECD) = 2.

A relaxação Π3+ dessa tarefa, cuja sequência de ações {MOVEAB, MOVEBC, MOVECD} atinge o estado-objetivo e tem custo ótimo igual a 4, é dada por:

V = {ATA, ATB, ATC, ATD}

I = {ATA}

G = {ATA, ATB, ATC, ATD}

A = {MOVEAB, MOVEBC, MOVECD}

* pre(MOVEAB) = {ATA}; add(MOVEAB) = {ATB}; del(MOVEAB) = ∅; cost(MOVEAB) = 1.
* pre(MOVEBC) = {ATB}; add(MOVEBC) = {ATC}; del(MOVEBC) = ∅; cost(MOVEBC) = 1.
* pre(MOVECD) = {ATC}; add(MOVECD) = {ATD}; del(MOVECD) = ∅; cost(MOVECD) = 2.
  1. Tomemos com base a tarefa de planejamento proposta em i. Os buracos, agora, são nomeados com H0...Hi, com H0 sendo o buraco mais à esquerda e Hi, com i > 1, o buraco mais à direita. Nosso objetivo, agora, é ter uma bola em H0 e outra em H1. Quando conseguimos levar a nossa bola até Hi, ganhamos outra bola no buraco H0 e podemos movimentar a nossa bola antiga para H1. Cada movimento de bola custa 1. Com isso, obtemos a família de tarefas de planejamento que segue o modelo abaixo:

V = {H0, H1, H2, H3, ..., Hi}

I = {H0}

G = {H0, H1}

A = {MOVE0,1, MOVE1,2, MOVE2,3, …, MOVE(i-1),i, GetNewBall}

* Para cada k indo de 1 a i:

pre(MOVE(k-1),k) = {Hk-1}; add(MOVE(k-1),k) = {Hk}; del(MOVE(k-1),k) = {Hk-1}; cost(MOVE(k-1),k) = 1.

* pre(GetNewBall) = {Hi}; add(GetNewBall) = {H0, H1}; del(GetNewBall) = {Hi}; cost(GetNewBall) = 1.

Com esse modelo para a família de tarefas de planejamento, o custo ótimo para a tarefa Pi será de i+1. Se relaxarmos o mesmo modelo, entretanto, o custo ótimo de Pi+ será sempre 1, já que bastará que utilizemos o operador MOVE0,1, de custo 1, para chegarmos no estado objetivo {H0, H1}. O modelo para a família de tarefas de planejamento relaxada é dado por:

V = {H0, H1, H2, H3, ..., Hi}

I = {H0}

G = {H0, H1}

A = {MOVE0,1, MOVE1,2, MOVE2,3, …, MOVE(i-1),i, GetNewBall}

* Para cada k indo de 1 a i:

pre(MOVE(k-1),k) = {Hk-1}; add(MOVE(k-1),k) = {Hk}; del(MOVE(k-1),k) = ∅; cost(MOVE(k-1),k) = 1.

* pre(GetNewBall) = {Hi}; add(GetNewBall) = {H0, H1}; del(GetNewBall) = ∅; cost(GetNewBall) = 1.

1. h∗(I) = 6  
   h+(I) = 4

O *Relaxed Task Graph* criado para o exercício se encontra no arquivo Relaxed\_Task\_Graph.png. Nele, a cor **vinho** representa hmax, enquanto a cor **azul** representa hadd. Com base nesse grafo, temos:

hmax(I) = 2

hadd(I) = 6

A relaxação da tarefa de planejamento dada é descrita por:  
V = {AT0, AT1, AT2, AT3, AT4, VISITED0, VISITED1, VISITED2, VISITED3, VISITED4}

I = {AT2, VISITED2}

G = VISITED0 Λ VISITED1 Λ VISITED2 Λ VISITED3 Λ VISITED4

O = {MOVEi,i+1, MOVEi+1,i} para todo i de 0 a 3.

Para todo MOVEj,k:

* pre(MOVEj,k) = ATj
* eff(MOVEj,k) = ATk Λ VISITEDk
* cost(MOVEj,k) = 1

No *Relaxed Task Graph*, renomeamos os operadores da seguinte forma:

* O0 = MOVE0,1
* O1 = MOVE1,0
* O2 = MOVE1,2
* O3 = MOVE2,1
* O4 = MOVE2,3
* O5 = MOVE3,2
* O6 = MOVE3,4
* O7 = MOVE4,3

1. A implementação se encontra no arquivo /fast-downward/src/search/planopt\_heuristics/h\_greedy\_relaxed\_plan.cc.
2. Abaixo se encontram os resultados obtidos para cada instância do diretório *castle*. Quando uma solução para o plano não existe, utilizamos o símbolo “-” no campo de custo. Quando pode ser que uma solução exista, mas ela não foi descoberta dentro do limite de tempo e memória definidos, utilizamos um *lower bound* (lb) para o custo. Marcamos o campo “*relaxed solvable*” com “*can’t answer*” quando o plano que utiliza a heurística implementada no exercício 1.c estava procurando por uma solução, mas foi interrompido pelo limite de tempo ou memória. Nesse caso nós não podemos afirmar se uma solução para o plano existe ou não.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instance | Relaxed solvable | Solved within resource limits | Heuristic value of the initial state | Discovered plan cost | Optimal relaxed plan cost | Optimal plan cost |
| 2-2-8 | Yes | Yes | 4 | 4 | 4 | 4 |
| 3-3-8 | Yes | Yes | 16 | 9 | 9 | 9 |
| 4-3-5 | Yes | Yes | 19 | 16 | 14 | 15 |
| 5-4-7 | No | No | 59 | - | lb = 14 | - |
| 5-4-9 | Yes | Yes | 77 | 19 | 19 | 19 |
| 5-4-10 | No | No | ∞ | - | - | - |
| 6-4-7 | Yes | Yes | 45 | 27 | lb = 15 | 26 |
| 7-5-4 | Yes | Yes | 124 | 38 | lb = 12 | 35 |
| 8-5-9 | Yes | Yes | 89 | 51 | lb = 16 | 39 |
| 9-6-5 | No | No | 173 | - | lb = 12 | - |
| 10-6-7 | Yes | Yes | 139 | 72 | lb = 16 | lb = 28 |
| 12-7-3 | Can’t answer | No | 410 | lb = 22 | lb = 17 | - |
| 16-9-1 | Can’t answer | No | 616 | lb = 12 | lb = 8 | - |