

3

Pipesworld 1 e 2

Dentro da proposta desta tese de avaliar a utilização de técnicas de IA para resolver o problema de planejamento em oleodutos, a primeira abordagem foi analisar o desempenho das ferramentas de modelagem e de técnicas de planejamento de propósito geral disponíveis. O Pipesworld é o resultado desta pesquisa. Ele é um domínio de planejamento inspirado no problema de transporte de oleodutos, sendo o primeiro do gênero especificado em PDDL [11, 16], que é uma linguagem utilizada como padrão pela comunidade de planejamento em inteligência artificial. O Pipesworld foi divulgado tanto nesta comunidade [21, 40] quanto na comunidade de oleodutos [30], como forma de validá-lo como um domínio relevante tanto do ponto de vista de pesquisa acadêmica como de indústria.

A Seção 3.1 apresenta os conceitos básicos de PDDL. A Seção 3.2 apresenta a primeira versão proposta para o Pipesworld, enquanto a Seção 3.3 mostra a versão do domínio que foi utilizada no benchmark da quarta *International Planning Competition*, IPC-4 [40]. Este evento compara o desempenho do estado da arte em termos de planejadores automáticos de propósito geral. Finalmente, a Seção 3.4 comenta as lições aprendidas no processo de modelagem. A análise do desempenho dos planejadores sobre o domínio é mostrada no Capítulo 4.

3.1

A linguagem PDDL

A intenção da linguagem PDDL (*Problem Domain Definition Language*) é expressar a estrutura de um determinado domínio de planejamento, ou seja, como o estado do modelo é representado, quais são as ações possíveis em cada estado, com suas pré-condições e efeitos, e quais são os objetivos a serem atingidos. Por motivos históricos [11], PDDL deixa propositalmente de lado a especificação de qualquer tipo de informação que tenha como objetivo guiar os planejadores no processo de solução do domínio.

Do ponto de vista da construção de um sistema de planejamento, a grande vantagem da PDDL é permitir a especificação do domínio em uma linguagem neutra, podendo utilizar posteriormente vários programas de planejamento automático de propósito geral, que independem do domínio que está sendo analisado. O papel da PDDL neste cenário é análogo ao de linguagens como o GAMS [6] na área de programação linear, não linear e inteira mista.

A estrutura da PDDL é apresentada juntamente com os exemplos nas seções a seguir. Um aspecto importante da linguagem é que a mesma é bastante minimalista em termos das construções obrigatórias. Grande parte das construções permitidas é opcional. Um problema definido em PDDL deve destacar, através do uso da palavra-chave *requirements*, quais são as construções opcionais requeridas pelo modelo. Maiores detalhes sobre estas construções opcionais e seu impacto são fornecidas nas seções seguintes.

3.2

Pipesworld 1

Esta Seção apresenta uma listagem comentada da primeira versão do domínio de planejamento Pipesworld, apresentado em [21]. Este modelo leva em consideração algumas características básicas do problema de planejamento apresentado no Capítulo 2.

O nome Pipesworld escolhido para o modelo é derivado das similaridades entre o modelo proposto e o domínio clássico Blocksworld [22], no sentido de que cada duto é modelado como uma pilha de blocos que conecta as áreas. A principal diferença no caso é que o tamanho dos blocos no Pipesworld é fixo.

3.2.1

Conceitos do modelo

Esta Seção apresenta os conceitos utilizados no Pipesworld, relacionando-os com os definidos no capítulo 2.

Bateladas unitárias

O modelo proposto define o conceito de bateladas unitárias para representar as bateladas apresentadas na Seção 2.2.8. Bateladas unitárias

são porções indivisíveis do mesmo produto. Os volumes dos dutos e das tancagens são definidos em termos de bateladas unitárias.

O volume real, em m^3 por exemplo, associado a uma batelada unitária pode ser ajustado baseado nas características da instância real que desejamos modelar. Valores menores para o volume das bateladas unitárias reduzem o erro de arredondamento que ocorre quando o volume real dos dutos e dos tanques são convertidos para bateladas unitárias, mas também aumentam consideravelmente o tamanho da instância, já que mais objetos deste tipo são criados.

Ações PUSH e POP

Os dutos, conforme apresentado na Seção 2.2.1, sempre conectam duas áreas. No modelo Pipesworld, uma destas áreas tem um papel chamado “FROM”, enquanto outra tem um papel chamado “TO”. Estes papéis definem um sentido padrão para o duto, que é o sentido do fluxo da área “FROM” para a área “TO”.

O modelo proposto define apenas duas ações, PUSH e POP. Ambas recebem como argumentos o duto a ser movimentado e a batelada a ser inserida no mesmo. A ação PUSH movimenta o duto em seu sentido padrão, enquanto a ação POP movimenta o duto no sentido oposto.

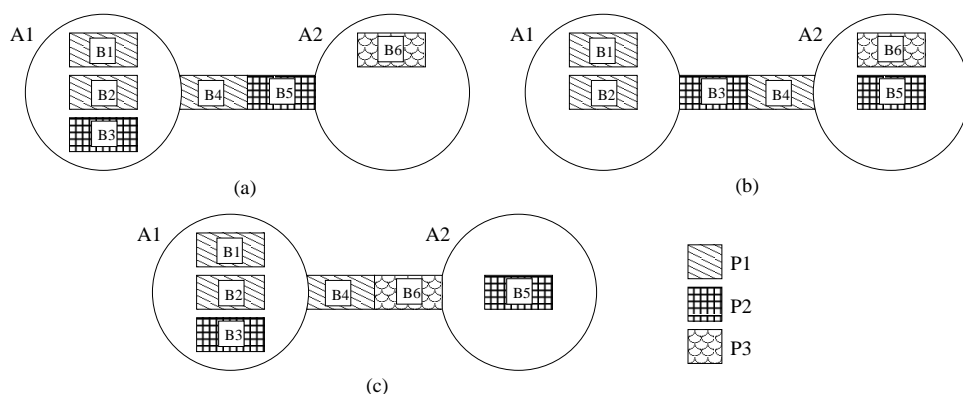


Figura 3.1: Exemplo de instância do Pipesworld

A Figura 3.1 mostra um exemplo das operações PUSH e POP. A área A1 exerce o papel “FROM”, enquanto a área A2 exerce o papel “TO”. O estado inicial é mostrado em (a), (b) mostra o estado da rede após uma ação “PUSH” com B3 sendo inserida no duto, (c) mostra o estado (b) após uma operação “POP” com B6 sendo inserida no duto.

Como as ações operam sobre um duto individualmente, a implementação do conceito de rotas de fluxo apresentado na Seção 2.2.4 é limitada

a rotas contendo um único segmento. Como consequência, não existe no Pipesworld a transferência de fluxo diretamente de um duto para outro, nem operações de sangria.

Tancagens

O Pipesworld suporta a definição de tancagens por produto e por área. Estas tancagens são implementadas através de construções especiais denominadas fluentes numéricos. Objetos deste tipo não são limitados a assumir os valores verdadeiro e falso, como os predicados em PDDL, podendo ser do tipo numérico.

O uso de fluentes numéricos permite que tanto a capacidade de cada tancagem quanto o valor corrente de seu volume seja modelado por números inteiros. Infelizmente, os fluentes numéricos são uma das construções opcionais citadas na Seção 3.1.

Para um exemplo simples de como a tancagem influencia o Pipesworld, considere novamente a Figura 3.1. Se a capacidade de armazenamento de P_2 em A_2 é igual a zero, então a transição do estado (a) para o estado (b) é proibida.

Interface

A restrição de interface, conforme descrita na Seção 2.2.2, é totalmente suportada no Pipesworld.

Objetivos

No Pipesworld, as bateladas podem opcionalmente ter uma área objetivo, para a qual devem ser transportadas pelo plano. Definimos as bateladas que tem um objetivo associado como **não proteláveis**, enquanto as bateladas que não possuem uma área de destino são **proteláveis**.

Esta definição explícita do destino de cada batelada difere do problema originalmente apresentado no capítulo 2, já que neste os objetivos são especificados em termos de demandas e ofertas de produtos, não especificando exatamente qual batelada deve ser transportada para cada área. Ou seja, as bateladas não proteláveis no Pipesworld não são fungíveis (Seção 2.1). Esta abordagem foi adotada para simplificar o modelo resultante.

3.2.2 Modelo PDDL comentado

O modelo PDDL parcial do Pipesworld 1 é apresentado a seguir, assim como uma instância do problema.

Uma instância de um problema em PDDL é definida em dois arquivos. O primeiro arquivo, denominado arquivo de domínio, descreve a estrutura e a dinâmica do problema. No caso do Pipesworld, por exemplo, este arquivo define os atributos obrigatórios das áreas e dos dutos, e quais são as pré e pós-condições das ações. Já o arquivo de problema descreve os dados específicos da instância a ser resolvida. No caso do Pipesworld, este arquivo descreve por exemplo as áreas de cada instância e como elas são conectadas, a localização inicial das bateladas e as áreas de destino das bateladas não proteláveis. Trechos comentados destes arquivos são apresentados em seções separadas a seguir.

Arquivo de domínio

```
;; Pipesworld 1
(define (domain pipelines)

;; Note a declaração de que o modelo exige construção
;; ‘fluents’. Estas são utilizados para representação do volume
;; corrente das tancagens e sua capacidade
(:requirements :typing :fluents)

;; Tipos do domínio:
;; tank: tancagem de um produto
;; batch-atom: uma batelada unitária
;; pipe: um duto
;; area: uma área
;; product: produto transportado pela rede, como gasolina, querosene,
;;          etc
(:types tank batch-atom pipe area product)

;; Início da Seção de predicados binários
(:predicates

;; Predicado que define que duas áreas são conectadas por um duto.
;; Por exemplo, se existem objetos A1 e A2 do tipo area, e um objeto
;; D1 do tipo pipe, e se declararmos o predicado (connect A1 A2 D1)
```

```

;; como verdadeiro, então D conecta as áreas A1 e A2. Caso contrário
;; estas áreas não são conectadas por D
(connect ?from ?to - area ?pipe - pipe)

;; Localização dos tanques e produtos armazenados pelos mesmos
(tank-in-area ?tank - tank ?area - area)
(tank-product ?tank - tank ?product - product)

;; Estes predicados representam o conteúdo de um duto. São
;; definidas a primeira (mais próxima da área ‘FROM’) e a
;; última (mais próxima da área ‘TO’) bateladas unitárias
;; de um duto. A seqüência das bateladas no duto é capturada
;; pelo predicado ‘follow’
(last ?batch-atom - batch-atom ?pipe - pipe)
(first ?batch-atom - batch-atom ?pipe - pipe)
(follow ?next ?previous - batch-atom)

;; Associa um produto a cada batelada unitária
(is-product ?batch-atom - batch-atom ?product - product)

;; Bateladas unitárias situadas em tanques
(on ?batch-atom - batch-atom ?tank - tank)

;; Indica que dois produtos tem a sua interface permitida dentro de
;; um duto
(may-interface ?product-a ?product-b - product)

)

;; Fim dos predicados

;; Define os produtos que podem ser transportados
(:constants lco gasoleo rat-a oca1 oc1b - product)

;; Função que relaciona o volume corrente de um tanque com a
;; sua capacidade
(:functions (tank-capacity ?tank - tank)
            (tank-current-volume ?tank - tank))

;; Ação PUSH
;; Movimenta uma batelada unitária para um duto, no sentido
;; da área ‘FROM’ para a área ‘TO’

```

```

(:action PUSH
  ;; Parâmetros da ação
  :parameters(
    ;; Duto que será movimentado
    ?pipe - pipe
    ;; Batelada unitária inserida no duto
    ?batch-atom-in - batch-atom
    ;; Tanque que recebe a batelada que está saindo do duto
    ;; na área de destino
    ?destination-tank - tank
  )
  ;; Esta Seção define parâmetros extras que precisam ser utilizadas na
  ;; Seção de efeitos. Elas são 'grounded', ou seja, assumem o valor de
  ;; algum objeto do domínio de acordo com as condições explicitadas
  ;; na Seção :precondition, detalhada a seguir
  :vars(
    ;; Áreas de origem e destino
    ?from-area - area
    ?to-area - area

    ;; Tanque de origem
    ?from-tank - tank

    ;; Primeira e última bateladas unitárias no duto
    ?first-batch-atom - batch-atom
    ?last-batch-atom - batch-atom

    ;; Produto da batelada que está sendo inserida no duto
    ?product-batch-atom-in - product

    ;; Produto da primeira batelada do duto
    ?product-first-batch - product

    ;; Produto da última batelada do duto
    ?product-last-batch - product
  )
  ;; Início da Seção de pré-condições. Elas devem ser verdadeiras para
  ;; que a ação possa ser aplicada
  :precondition
  (and

```

```

;; Realiza o bind de todas as variáveis da Seção vars:, em função
;; dos objetos associados aos parâmetros da ação. Este próximo
;; trecho, por exemplo, associa objetos (realiza o ‘‘bind’’) às
;; variáveis ?first-batch-atom, ?last-batch-atom, ?from-area e
;; ?to-area a partir do parâmetro ?pipe
(first ?first-batch-atom ?pipe)
(last ?last-batch-atom ?pipe)
(connect ?from-area ?to-area ?pipe)
(on ?batch-atom-in ?from-tank)

;; Associa os valores corretos dos produtos da batelada
;; inserida no duto, da primeira batelada do duto e da
;; última batelada do duto
(is-product ?batch-atom-in ?product-batch-atom-in)
(is-product ?first-batch-atom ?product-first-batch)
(is-product ?last-batch-atom ?product-last-batch)

;; Verifica se produto que está sendo inserido no duto tem
;; interface permitida com o primeiro produto do duto
(may-interface ?product-batch-atom-in ?product-first-batch)

;; Os tanques devem estar nas áreas corretas
(tank-in-area ?from-tank ?from-area)
(tank-in-area ?destination-tank ?to-area)

;; Verifica se tanque na área de destino pode receber o produto que
;; está deixando o duto
(tank-product ?destination-tank ?product-last-batch)

;; Verifica se existe espaço disponível no tanque de destino
(< (tank-current-volume ?destination-tank)
   (tank-capacity ?destination-tank) )
)

;; Esta Seção descreve o efeito das ações
:effect
  (and

    ;; Atualiza o estado do duto

    ;; A batelada inserida é a nova primeira batelada do duto

```



```

(first ?batch-atom-in ?pipe)
(not (first ?first-batch-atom ?pipe))

;; Atualiza a relação de seqüenciamento no duto e define o
;; a nova última batelada no duto
(forall (?batch - batch-atom)
  (when (follow ?last-batch-atom ?batch)
    (and (last ?batch ?pipe)
         (not (follow ?last-batch-atom ?batch))
         (follow ?first-batch-atom ?batch-atom-in)))
  )

;; Caso especial para duto com uma única batelada, neste caso os
;; predicados 'last' e 'first' referem-se à mesma batelada
;; unitária
(when (= ?first-batch-atom ?last-batch-atom)
  (last ?batch-atom-in ?pipe))
(not (last ?last-batch-atom ?pipe))

;; A batelada unitária inserida no duto é removida do tanque de
;; origem
(not (on ?batch-atom-in ?from-tank))

;; A batelada removida do duto é inserida no tanque de destino
(on ?last-batch-atom ?destination-tank)

;; Ajusta o volume dos tanques de origem e de destino
(increase (tank-current-volume ?destination-tank) 1)
(decrease (tank-current-volume ?from-tank) 1))
)

(:action POP
  ;; Esta ação é análoga à ação PUSH, a única diferença é no sentido
  ;; no qual o duto é movimentado. A ação POP movimenta o duto da
  ;; no sentido da "to-area" definida pelo predicado "connect"
  ;; ...
)
)

```

Arquivo de problema

```
(define (problem problem001)

  (:domain pipelines)

  ;; Define os objetos desta instância específica
  (:objects

    ;; Bateladas da instância
    B1 B2 B3 B4 B5 B6 B7 - batch-atom

    ;; Áreas da instância
    A1 A2 - area

    ;; Duto da instância
    P1 - pipe

    ;; Tanques, definimos um tanque por produto, por área
    TA1LCO TA1GASOLEO TA1RAT-A TA1OCA1 TA1OC1B
    TA2LCO TA2GASOLEO TA2RAT-A TA2OCA1 TA2OC1B - tank

  )

  ;; Define os predicados que são verdadeiros no estado
  ;; inicial. Assume-se a hipótese do ‘‘mundo fechado’’, ou seja, o que
  ;; não pode ser provado como verdadeiro é falso
  (:init

    ;; Definição da topologia da rede, neste caso um único duto que
    ;; liga as duas áreas definidas
    (connect A1 A2 P1)

    ;; Restrições de interface
    (may-interface lco lco)
    (may-interface gasoleo gasoleo)
    (may-interface gasoleo lco)
    (may-interface rat-a rat-a)
    ;; ...

    ;; Especifica a localização dos tanques
```

```
(tank-in-area TA1LCO A1)
(tank-in-area TA1GASOLEO A1)
(tank-in-area TA1RAT-A A1)
;; ...

;; Especifica a capacidade máxima de cada tanque
(= (tank-capacity TA1LCO) 10)
(= (tank-capacity TA1GASOLEO) 10)
(= (tank-capacity TA1RAT-A) 10)
;; ...

;; Especifica o produto de cada tanque
(tank-product TA1LCO lco)
(tank-product TA1GASOLEO gasoleo)
;; ...

;; Especifica o produto de cada batelada
(is-product B1 lco)
(is-product B2 gasoleo)
;; ...

;; Especifica o volume corrente de cada tanque
(= (tank-current-volume TA1RAT-A) 0)
(= (tank-current-volume TA1OCA1) 0)
(= (tank-current-volume TA1OC1B) 2)
(= (tank-current-volume TA2LCO) 1)
;; ...

;; bateladas inicialmente localizada nas áreas (tanques)
;; A1
(on B5 TA1OC1B)
(on B4 TA1OC1B)
;; A2
(on B6 TA2OCA1)
(on B7 TA2LCO)

;; Bateladas inicialmente localizadas no duto P1,
;; que são B1, B2 e B3, nesta ordem, da área ‘‘FROM’’ para
;; a área ‘‘TO’’
(first B1 P1)
(last B3 P1)
```

```

    (follow B2 B1)
    (follow B3 B2)

)
;; Objetivo: posicionar batelada B2 na área A2, e a
;; batelada B3 na área A1
(:goal (and
  (on B2 TA2GASOLEO)
  (on B3 TA1RAT-A)
))
)

```

3.2.3 Exemplos

Esta Seção apresenta seis instâncias ilustrativas do Pipesworld 1, com soluções. Cada instância explora um aspecto específico do domínio. Os resultados obtidos são comentados de forma a explicar a dinâmica da solução.

Todos os exemplos trabalham com um conjunto comum de produtos. Existe uma tancagem para cada produto definido em todas as áreas. O nome para as tancagens segue a regra T[nome da área][nome do produto]. Por exemplo, TA1OC1B é a tancagem para o produto OC1B na área A1.

O programa de planejamento FF [17], com parâmetros padrão de funcionamento, é utilizado para resolver as instâncias de exemplo. Este planejador foi escolhido pois foi o que apresentou o melhor desempenho na IPC-3 [23]. Todos os tempos reportados são relativos ao tempo de CPU gasto pelo FF em um PC de 1GHz, Pentium III, com 1GB de memória.

3.2.4 Reversão

Esta Seção apresenta uma instância na qual um plano viável deve reverter o duto por pelo menos duas vezes. A Figura 3.2 mostra a configuração de rede utilizada e o estado inicial do duto. O objetivo é movimentar a batelada B_2 para A_2 e a batelada B_3 para A_1 .

A definição PDDL desta instância é a mostrada resumidamente na Seção 3.2.2.

O planejador FF retornou um plano válido para a instância em menos de um segundo, o plano é mostrado na Figura 3.3. Os argumentos logo após

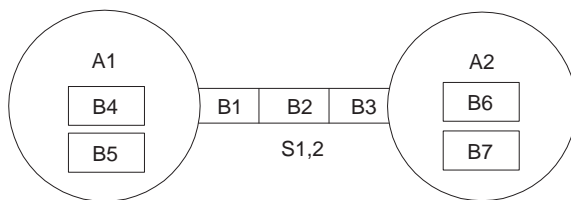


Figura 3.2: Exemplo de instância com reversão

as ações PUSH e POP são o duto movimentado, a batelada que é inserida no duto em cada ação e o tanque de destino, nesta ordem. Como existe apenas um único duto nesta rede de exemplo, o primeiro argumento é sempre S12.

```

step  0: PUSH S12 B4 TA2RAT-A
      1: PUSH S12 B5 TA2GASOLEO
      2: POP  S12 B3 TA10C1B
      3: POP  S12 B7 TA10C1B
      4: POP  S12 B6 TA1LCO
      5: POP  S12 B2 TA1RAT-A
      6: PUSH S12 B1 TA2GASOLEO

```

Figura 3.3: Solução para a instância de reversão

3.2.5 Escalabilidade

Este experimento ilustra como o tamanho da instância afeta o desempenho do planejador. A mesma instância da Seção anterior é modificada da seguinte forma: cada batelada unitária B_n é substituída por duas bateladas unitárias, B_{n1} e B_{n2} .

Esta nova instância admite o mesmo plano obtido para a instância original, com uma transformação apropriada das ações. Por exemplo, pode-se obter um plano válido substituindo a ação “PUSH S12 B4 TA2RAT-A” no plano original por duas novas ações, “PUSH S12 B41 TA2RAT-A” e “PUSH S12 B42 TA2RAT-A”, e assim por diante.

O planejador FF levou cinco horas para resolver esta instância modificada, com uma grande variação em relação ao tempo utilizado para solução da instância anterior. Isto indica que o domínio pode ser desafiador para os planejadores automáticos. O motivo deste impacto no desempenho do planejador é analisado na Seção 3.3.1.

3.2.6 Interface

Esta instância é obtida a partir da definida na Seção 3.2.4, adicionando-se uma restrição que proíbe a interface entre os produtos LCO e OC1B. Estes são os produtos das bateladas B_1 e B_4 , respectivamente. Esta restrição adicional torna a primeira ação do plano mostrado na Figura 3.3 inválida.

O planejador FF foi capaz de resolver esta nova instância em aproximadamente 10 segundos, retornando o plano mostrado na Figura 3.4, que possui quatro vezes mais ações do plano da Figura 3.3 e realiza sete reversões de fluxo no duto.

```

step    0: POP S12 B7 TA1LCO
        1: PUSH S12 B4 TA2LCO
        2: PUSH S12 B5 TA2RAT-A
        3: PUSH S12 B1 TA2GASOLEO
        4: POP S12 B6 TA1LCO
        5: POP S12 B7 TA1OC1B
        6: POP S12 B2 TA1OC1B
        7: POP S12 B3 TA1OCA1
        8: PUSH S12 B1 TA2RAT-A
        9: PUSH S12 B6 TA2GASOLEO
       10: PUSH S12 B4 TA2LCO
       11: POP S12 B2 TA1OC1B
       12: POP S12 B7 TA1OCA1
       13: POP S12 B3 TA1LCO
       14: PUSH S12 B4 TA2RAT-A
       15: PUSH S12 B1 TA2LCO
       16: PUSH S12 B6 TA2GASOLEO
       17: PUSH S12 B5 TA2OC1B
       18: POP S12 B3 TA1OC1B
       19: POP S12 B7 TA1OCA1
       20: POP S12 B2 TA1LCO
       21: POP S12 B4 TA1RAT-A
       22: PUSH S12 B6 TA2OC1B
       23: PUSH S12 B1 TA2GASOLEO

```

Figura 3.4: Solução para a instância de interface

3.2.7 Tancagem

Esta instância define uma nova rede, mostrada na Figura 3.5. Por simplicidade, apenas o produto LCO é definido. O objetivo é mover B_2 para A_3 .

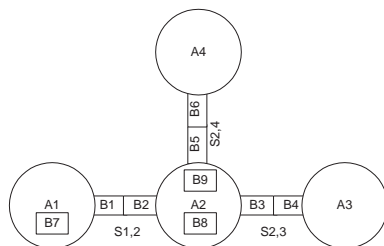


Figura 3.5: Instância com restrição de tancagem apertada

Sem a restrição de tancagem, a solução é trivial e é mostrada na Figura 3.6.

```

step    0: PUSH S12 B7 TA2LCO
        1: PUSH S23 B2 TA3LCO
        2: PUSH S23 B8 TA3LCO
        3: PUSH S23 B9 TA3LCO

```

Figura 3.6: Solução trivial para a instância de tancagem

Se a tancagem para LCO for limitada a 2 na área A_2 , o plano deve inicialmente liberar algum espaço em A_2 de forma a poder movimentar B_2 from $S_{1,2}$ to A_2 .

Um plano válido para esta situação foi retornado pelo FF em menos de cinco segundos, e é mostrado na Figure 3.7. O plano criou espaço na área A_2 bombeando produtos para o duto $S_{2,4}$ e $S_{2,3}$, utilizando este último duto como um tanque de “segundo nível” para o armazenamento de produtos.

```

step    0: PUSH S23 B8 TA3LCO
        1: PUSH S12 B7 TA2LCO
        2: PUSH S24 B9 TA4LCO
        3: POP  S23 B4 TA2LCO
        4: PUSH S23 B2 TA3LCO
        5: POP  S24 B6 TA2LCO
        6: PUSH S23 B8 TA3LCO
        7: PUSH S23 B9 TA3LCO

```

Figura 3.7: Solução para instância com tancagem apertada

3.2.8 Roteamento

Outro aspecto importante é o do roteamento escolhido para o transporte das bateladas em um oleoduto. A Figura 3.8 mostra uma instância na

qual as bateladas podem chegar ao seu destino através de três rotas diferentes. Esta decisão é inspirada na decisão de qual rota de fluxo, apresentada na Seção 2.2.4, deve ser utilizada na operação.

Nesta instância, a batelada B_8 deve ser posicionada em A_3 . Isto pode ser conseguido através de três caminhos distintos: $P_1 = \{S_{1,4}, S_{4,3}\}$, $P_2 = \{S_{1,2}, S_{2,4}, S_{4,3}\}$ ou $P_3 = \{S_{1,2}, S_{2,3}\}$. Se estamos interessados em um plano que minimiza o número de operações de bombeamento, P_1 é a melhor escolha, já que requer apenas 4 operações, enquanto P_2 e P_3 requerem 6 operações.

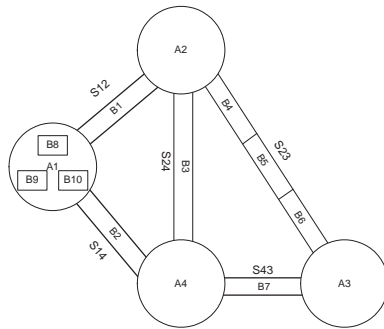


Figura 3.8: Instância com várias opções de roteamento

O planejador FF retornou a solução P_1 em menos de 12 segundos. O plano resultante é mostrado na Figura 3.9.

```

step    0: PUSH S14 B8 TA4LC0
        1: PUSH S14 B9 TA4LC0
        2: PUSH S43 B8 TA3LC0
        3: PUSH S43 B2 TA3LC0

```

Figura 3.9: Solução da instância de roteamento

3.2.9 Ciclos

Uma variação interessante da instância mostrada na Seção 3.2.8 é remover as bateladas B_9 and B_{10} da mesma. Pode parecer a princípio que isto torna a instância inviável, mas este não é o caso.

A solução mostrada na Figura 3.10 foi encontrada pelo FF em menos de 24 segundos.

Pode-se observar que os passos $\{0, 1, 2\}$ formam um ciclo no fluxo das bateladas, $C = \{S_{1,4}, S_{2,4}, S_{1,2}\}$. Ao final destes passos, B_8 é posicionada


```

step    0: PUSH S14 B8 TA4LC0
        1: POP  S24 B2 TA2LC0
        2: POP  S12 B3 TA1LC0
        3: PUSH S14 B1 TA4LC0
        4: PUSH S43 B8 TA3LC0
        5: POP  S23 B7 TA2LC0
        6: PUSH S24 B4 TA4LC0
        7: PUSH S43 B2 TA3LC0

```

Figura 3.10: Solução que utiliza ciclos

em $S_{1,4}$, e B_1 é posicionada em A_1 . Esta batelada é então utilizada para movimentar B_8 para A_4 , no passo 3. O passo 4 move B_8 para $S_{4,3}$. Uma operação similar é então utilizada para movimentar B_2 para A_4 , com esta batelada sendo então utilizada para movimentar B_8 para o seu destino final.

3.3

Pipesworld 2

O Pipesworld 2 é o resultado da modificação do Pipesworld 1 para sua utilização como benchmark do IPC-4. O domínio foi incluído na competição como consequência do compromisso dos organizadores do evento em benchmarks que se aproximem de aplicações reais, ao contrário dos domínios clássicos utilizados em versões anteriores do evento [24].

O Pipesworld 2 foi desenvolvido em conjunto com Jörg Hoffmann, participante do comitê organizador da competição.

3.3.1

Descrição das modificações no Pipesworld 1

As modificações efetuadas no Pipesworld 2 foram: inclusão de componente temporal no problema, remodelagem da forma com que as capacidades dos tanques são representadas e desmembramento das ações PUSH e POP. Estas modificações são detalhadas nas seções seguintes.

Ações temporais e prazos

No Pipesworld 2, os dutos podem possuir vazões distintas. Esta característica foi implementada através das construções temporais introduzidas na versão 2.1 da PDDL [16]. Se a vazão de um duto é s , associamos às ações “PUSH” e “POP” deste duto uma duração proporcional a $\frac{1}{s}$.

São também definidos prazos para a entrega das bateladas não proteláveis. Para cada uma destas bateladas, existe um literal que indica se ela pode ser removida de qualquer duto na rede. No estado inicial, este literal é verdadeiro para todas as bateladas não proteláveis, e torna-se falso no prazo final associado à batelada.

Nova modelagem para os tanques

Como forma de tornar o domínio acessível ao maior número de planejadores, a comissão organizadora do evento decidiu não utilizar domínios com fluentes numéricos. Em função disto, a modelagem utilizada para os tanques, apresentada na Seção 3.2.1, foi modificada para trabalhar com o conceito de *slots* de tancagem. Um *slot* é associado a um produto e uma área e possui um estado que pode ser ocupado ou não ocupado. Cada *slot* tem capacidade de armazenar uma batelada unitária. Os *slots* são então criados nas áreas em quantidade igual à capacidade da mesma de armazenamento dos produtos.

Desmembramento das ações PUSH e POP

Conforme mostrado na Seção 3.2.2, a definição original das ações “PUSH” e “POP” recebem, além do duto a ser movimentado e da batelada a ser inserida, vários outros parâmetros relacionados com produtos, tanques, etc. Em particular, para que seja possível a atualização do conteúdo do duto, as ações precisam de parâmetros associados à primeira, última e penúltima batelada do duto. Desta forma, esta ação possui 4 parâmetros sobre as bateladas, levando a pelo menos n^4 instanciações possíveis para a ação quando existem n bateladas unitárias. Foi observado que este fato tornava o domínio completamente inviável para qualquer planejador que instancia todas as ações possíveis já na fase de pré-processamento. Como muitas bateladas unitárias são necessárias mesmo para a codificação de domínios simples, estes planejadores abortavam a sua execução já nesta fase.

Este fenômeno foi contornado dividindo as ações “PUSH” e “POP” em duas partes, uma ação de “início” que recebe como parâmetros apenas a batelada inserida no duto e a primeira batelada do mesmo, e outra ação de “finalização” que recebe como parâmetros apenas a última e a penúltima batelada do duto. Estas ações são ligadas através de alguns predicados auxiliares, garantindo que são executadas na ordem correta. Esta divisão

não é necessária para dutos de tamanho 1, logo foram também criadas ações específicas para a movimentação destes dutos, denominadas “PUSH-UNITARY” e “POP-UNITARY” [40].

3.4

Conclusões

Esta Seção apresenta as conclusões obtidas de nossa experiência na utilização de planejadores de propósito geral para o tratamento de um problema real, descrito no capítulo 2. Neste ponto nos restringimos aos aspectos de modelagem, ou seja, na transformação do problema original em um modelo PDDL. A análise detalhada dos resultados obtidos pelos planejadores é mostrada no capítulo 4.

3.4.1

Limitações da PDDL

Como não é possível em PDDL a criação e remoção de um objeto no domínio, nenhuma batelada é inserida ou removida da rede durante o processo de planejamento, sendo simplesmente movimentadas no sistema.

Uma das principais vantagens da utilização da PDDL seria, a princípio, o desacoplamento entre o processo de modelagem e o de solução do problema propriamente dito. Isto permitiria que, no processo de solução, fossem realizadas experimentos com diversas estratégias disponíveis¹, implementadas por diferentes programas de planejamento.

A construção *requirements*, comentada na Seção 3.1, é o principal empecilho para que isto aconteça. Como grande parte das construções da linguagem tem sua implementação opcional, o modelo resultante deve utilizar apenas o conjunto obrigatório das construções de forma a avaliar o maior número possível de planejadores. Isto aconteceu por exemplo na modelagem de tancagem utilizada no Pipesworld 2, descrita na Seção 3.3.1. Neste caso específico, e como tipicamente acontece, a utilização das construções mais básicas aumenta a diferença entre o problema real e o problema que é modelado de fato.

No caso do Pipesworld, estas limitações no modelo são: bateladas unitárias, destino explicitado para as mesmas, rotas de fluxo com apenas um segmento, impossibilidade de sangria e bateladas que não são removidas nem inseridas no sistema.

¹Estas opções de estratégias são comentadas no Capítulo 4

A impossibilidade de incorporação de qualquer informação específica do domínio que auxilie o planejamento também é frustrante para o desenvolvedor. A incorporação deste tipo de informação ao sistema é fundamental para um bom desempenho do mesmo, e por vezes está facilmente acessível.

3.4.2

Aderência do processo de modelagem e solução

O exemplo relatado na Seção 3.3.1 é emblemático da aderência que acaba existindo de fato entre o modelo que é proposto e a forma de solução para o mesmo. A opção feita pelo desmembramento das ações PUSH e POP foi feita exclusivamente em função das características dos planejadores. Ela não é intuitiva em termos de modelagem, dividindo artificialmente uma ação que é única no problema real.

3.4.3

Dificuldade de testes e manutenção

Na medida em que o modelo PDDL fica maior e mais complexo, cresce a preocupação de verificar se as definições das ações estão corretas. Por exemplo, é interessante conferir se a ação PUSH de fato está atualizando o duto da maneira correta, respeitando as restrições de tancagem.

Não existe nenhum arcabouço para testes de domínios especificados em PDDL. Tipicamente isto é feito gerando pequenos exemplos nos quais deve-se verificar manualmente, a partir do plano gerado, se o efeito das ações sobre o sistema foi o esperado e se nenhuma ação foi executada quando não poderia. Em alguns casos, apenas a criação destes exemplos já pode ser um processo complicado.