

## 8

### A segunda versão do Plumber

Este capítulo apresenta uma segunda versão do Plumber, desenvolvida utilizando o PLANSIM e com o objetivo de tratar problemas operacionais de transporte em oleodutos. A Seção 8.1 define o problema de planejamento tratado, considerando a terminologia introduzida no Capítulo 2. Nas seções seguintes, detalhamos como cada um dos *hot spots* definidos no PLANSIM foi instanciado no Plumber, apresentando as classes que implementam o simulador, ações, heurísticas e objetivos. Em particular, as seções 8.3 e 8.7 descrevem o papel que a definição dos eventos que discretizam o processo de transporte em oleodutos tem no projeto do planejador automático. Finalmente, na Seção 8.8 apresentamos os resultados obtidos pela versão corrente do Plumber, com as conclusões sendo apresentadas logo após.

#### 8.1

##### Definição do problema

Como visto no Capítulo 2, o problema de transporte em oleodutos possui vários aspectos. Os que são considerados no Plumber 2 são os seguintes:

1. Restrições de interface, sem limite no número de produtos (Seção 2.2.2);
2. Restrições de tancagem agregada (Seção 2.2.3);
3. Múltiplas rotas de fluxo, sem restrição no número de fontes e destinos. As rotas de fluxo podem ser ativadas simultaneamente. Não são consideradas operações de sangria (Seção 2.2.4);
4. Ordens de serviço de produção e demanda, que podem ser definidas com intervalos com a resolução de segundos e sem restrições de horizonte; (Seção 2.2.5);
5. Bateladas não definidas a priori (Seção 2.4);

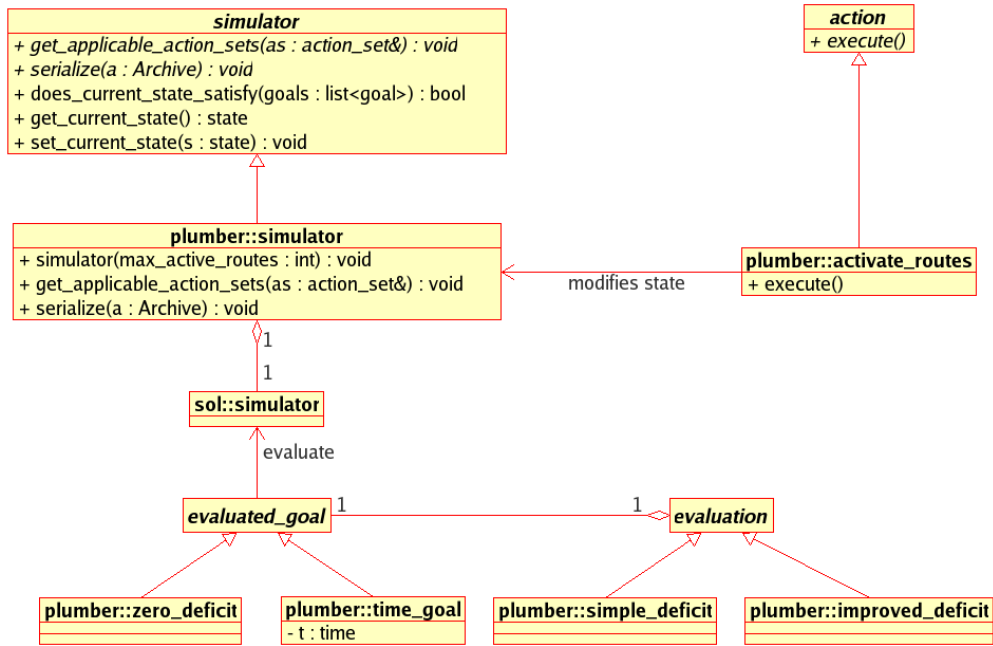


Figura 8.1: Instanciação do Plumber 2 a partir do PLANSIM

## 8.2

### Instanciação do PLANSIM

A Figura 8.1 apresenta a instanciação do PLANSIM para a implementação do Plumber. As classes instanciadas são descritas nas seções seguintes.

## 8.3

### O simulador a eventos discretos, SOL

Um aspecto crítico para o sucesso do sistema é a definição de como modelar um processo contínuo, como é o caso da operação de uma rede de oleodutos, em um processo discreto. A definição de como discretizar o processo faz parte efetiva da solução do problema.

Em modelos de programação linear inteira mista, por exemplo, tipicamente o tempo é discretizado em intervalos iguais. Quanto maior este intervalo, menor o tamanho da formulação resultante e maior a facilidade para sua solução. No entanto, a granularidade também é maior, o que coloca em um mesmo “instante” de tempo eventos que na realidade não ocorrem simultaneamente. Isto aumenta a distância entre o processo e seu modelo, que leva aos problemas relatados nas seções 2.6 e 3.4.1.

Uma das principais vantagens de nossa abordagem é dar uma maior flexibilidade nesta discretização. O uso de simuladores a eventos discretos como forma de modelagem não impõe um intervalo fixo para discretização, mas pode utilizá-lo se for conveniente. Como veremos mais adiante, os eventos considerados pelo simulador podem ser modificados durante o processo de construção do sistema, com o objetivo de torná-lo mais adequado ao mecanismo de solução que é utilizado.

Em um simulador a eventos discretos, um evento é definido como uma ocorrência instantânea que pode modificar o estado do sistema. O simulador possui uma lista de eventos, com cada elemento desta lista contendo uma hora de ativação e uma rotina associada. O simulador, em cada passo de simulação, remove o(s) evento(s) mais iminentes desta lista, executa as rotinas associadas e avança o relógio de simulação para o tempo correspondente ao(s) evento(s).

Foi desenvolvido para o Plumber um simulador de oleodutos denominado SOL. O estado da simulação é representado pelo conteúdo dos dutos e dos tanques, e o tempo de simulação. As ações que podem ser aplicadas são operações de bombeamento, vistas na Seção 2.2.6.

### 8.3.1

#### Definição inicial dos eventos

Durante a definição dos eventos de simulação devemos procurar aqueles que representam pontos de decisão importantes na programação. Na construção do SOL, os seguintes eventos foram inicialmente considerados:

1. Chegada de alguma interface em uma área: este evento é importante pois neste momento o planejador pode optar por mudar a configuração de rotas de fluxo ativas;
2. Início e final de uma operação de demanda ou de produção;
3. Volume corrente da tancagem chegando ao máximo ou mínimo valor operacional;

Estes eventos foram definidos baseados nos eventos do LPSS, um sistema de simulação utilizado pelos programadores da Transpetro. A lista de eventos do simulador é atualizada em cada passo de simulação com estes eventos, cuja hora prevista é função das operações de bombeamento que estão ativas em um determinado instante e dos eventos exógenos representados pelas ordens de serviço.

## 8.4 Ações

Como visto na Seção 7.1.4, um simulador utilizado no PLANSIM deve prover o serviço de indicar, dado um determinado estado, uma lista com todos os conjuntos de ações, ou *actions*, que podem ser aplicados. Isto passa inicialmente pela definição do que é uma ação em nosso sistema.

No Plumber, a ação consiste de uma operação de bombeamento. Como parâmetros desta ação temos a rota, vazão utilizadas e o produto que é inserido no duto. Esta instanciação é feita pela classe *activate\_routes* mostrada na Figura 8.1.

O procedimento de gerar todas as ações possíveis em um determinado estado é realizado da seguinte forma. Para cada rota de fluxo definida, o simulador verifica o produto que está no primeiro duto na rota e gera ações para todos os produtos que podem fazer interface com este, e com estoque na área de origem da rota. Desta forma geramos apenas operações de bombeamento que respeitam a restrição de interface.

Para a ativação simultânea de rotas, devemos gerar todos os conjuntos de rotas não interferentes (Seção 2.2.4) possíveis. Como forma de agilizar este processo, o SOL realiza um pré-processamento das rotas definidas, de forma a informar rapidamente se duas rotas são interferentes ou não. O SOL também é configurável para definir a cardinalidade máxima do conjunto de ações a serem retornadas, de forma a facilitar o teste do sistema, como mostraremos na Seção 8.8.1.

Existem dois casos especiais que devem ser considerados. A não ativação de nenhuma rota e uma situação na qual qualquer ação aplicada gera uma violação nas restrições de tancagem. Estes casos são detalhados a seguir.

Em algumas situações, o plano deve conter ações “vazias”, que tem como efeito simplesmente a evolução do simulador até o próximo evento. Estas ações são representadas por um conjunto vazio de ações, incluído na lista retornada pelo método *plumber::simulator::get\_applicable\_action\_sets*.

Já em outros casos, as restrições de tancagem são violadas independente da ação aplicada. Isto acontece por exemplo quando existe uma demanda ativa em uma área sem o produto, e nenhuma quantidade deste produto está na adjacência da área. No estado mostrado na Figura 8.2, se houver uma demanda ativa por P3 na área A3 e o estoque de P3 em A3 for zero, não existe forma de esta demanda ser atendida, e uma violação na restrição de tancagem é sempre gerada. Este estado, para efeitos de pla-

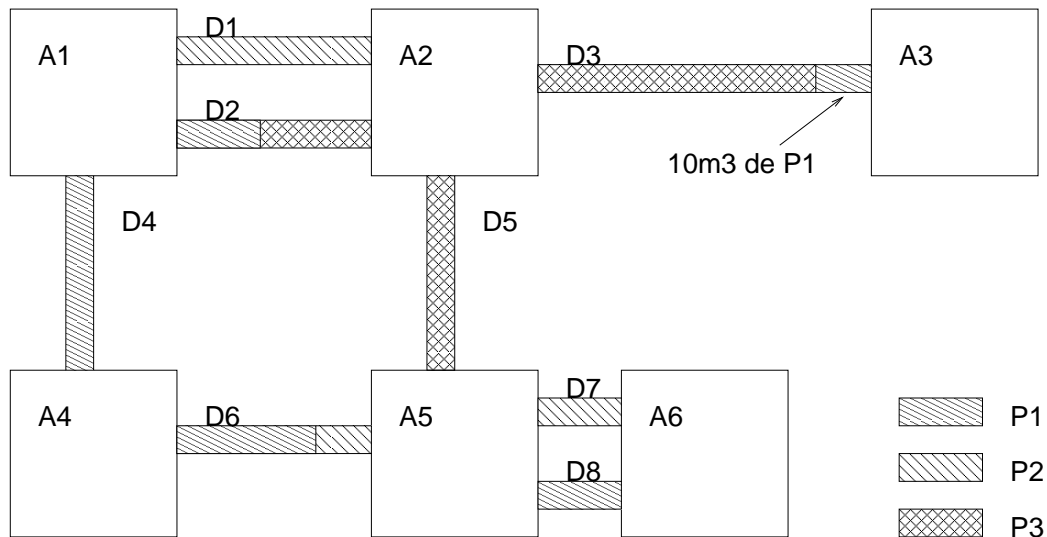


Figura 8.2: Exemplo de rede de oleodutos

nejamento, é terminal. Em função disto, nestas situações uma lista vazia é retornada pelo método *plumber::simulator::get\_applicable\_action\_sets*, indicando ao PLANSIM que o estado não deve ser expandido no processo de busca.

## 8.5 Heurísticas

As heurísticas utilizadas pelo Plumber 2 são baseadas em estimativas do volume mínimo de produto a ser deslocado de forma a atender todas as ordens de serviço programadas. Para isto, são criados os conceitos de déficit global simples e melhorado, detalhados a seguir.

### 8.5.1 Déficit global simples

O déficit global simples é calculado da seguinte forma. Inicialmente, geramos um perfil de volume projetado para cada tancagem onde existem ordens de serviço programadas, como mostra a Figura 8.3. Caso o perfil gerado permaneça no intervalo entre zero e a capacidade da tancagem, não é necessária nenhuma movimentação do produto nesta área no horizonte de tempo dado.

No exemplo da Figura, no entanto, o volume total gerado pela produção programada no intervalo de 02:00 até 04:00 excede a capacidade máxima de armazenamento em  $30m^3$ . Isto significa que, no mínimo, esta

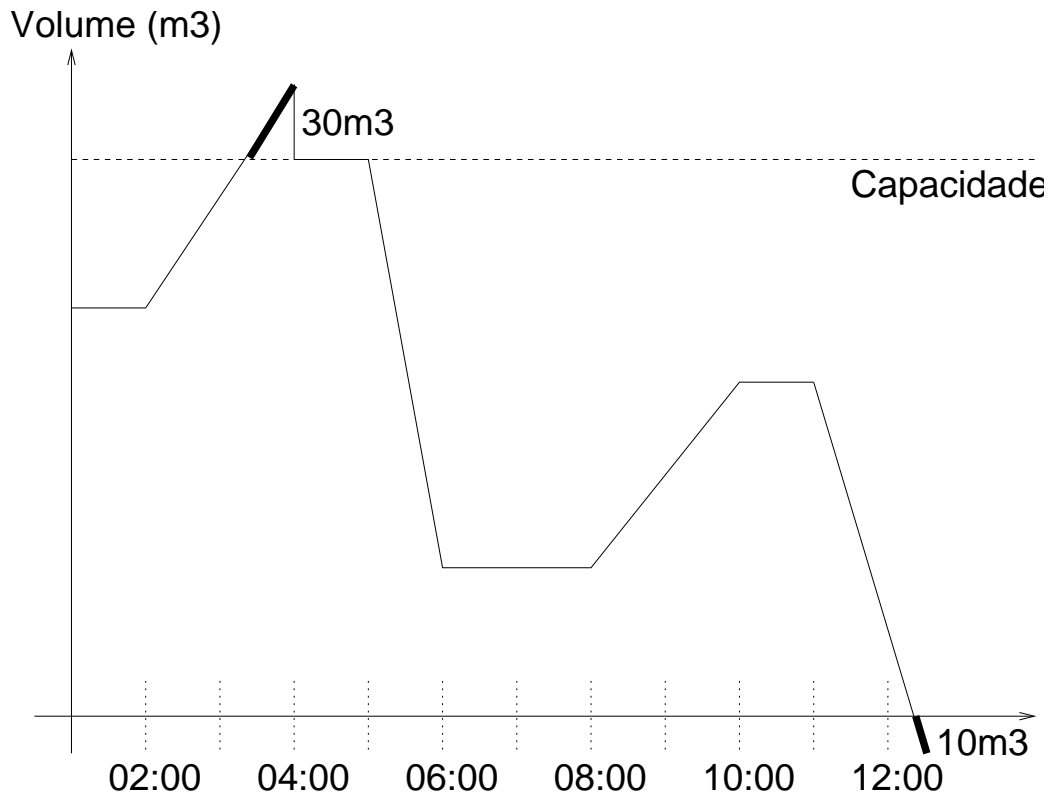


Figura 8.3: Exemplo de perfil de tancagem

quantidade de produto deve ser movimentada pela rede, através de uma operação de bombeamento deste produto, de forma a não gerar violação na restrição de tancagem. Dizemos que neste caso existe um **déficit de capacidade** de  $30m^3$ .

Continuando no exemplo da Figura 8.3, há uma demanda programada do produto de 05:00 a 06:00hs. Assumindo que o déficit anterior foi resolvido, o estoque do produto é capaz de atender esta demanda. Logo, não existe nenhum déficit associado a esta ordem de serviço. Da mesma forma, a tancagem é capaz de suportar a produção programada para ocorrer entre 08:00 e 10:00.

Já a demanda iniciada às 11:00 horas não será atendida, gerando um **déficit de estoque** de  $10m^3$ . De forma análoga, podemos dizer que este volume também deve ser movimentado pela rede, na forma de recebimento deste produto por esta área, para que a restrição de tancagem não seja violada.

O somatório destes déficits, para todos os produtos em todas as áreas, define uma função heurística inicial para o problema. Estados com déficit global simples menores são, provavelmente, mais promissores. Esta heurística é admissível, já que nenhum plano viável pode movimentar uma

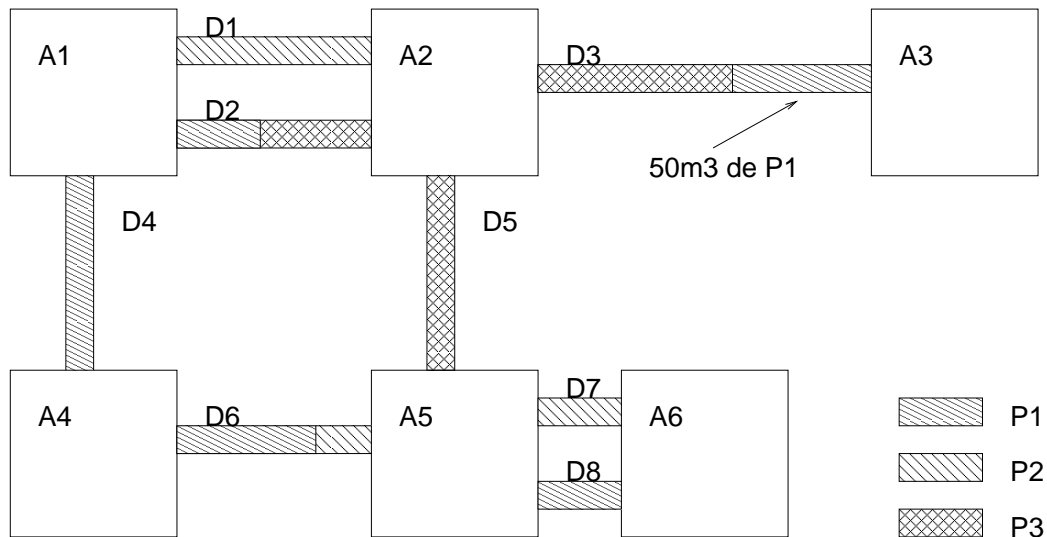


Figura 8.4: Outro estado para a rede de oleodutos

quantidade menor de produtos do que este déficit global.

### 8.5.2 Déficit global melhorado

Um problema na heurística apresentada na Seção anterior é que ela não leva em consideração os produtos que estão no duto em um determinado estado, restringindo-se a analisar o perfil da tancagem em função das ordens de serviço.

Para um exemplo, vejamos a Figura 8.4. Ela representa um estado diferente da mesma rede mostrada na Figura 8.2. Supondo que exista um déficit de estoque de de P3 em A3, o estado da Figura 8.4 é mais promissor do que o mostrado na Figura 8.2, já que na primeira P3 está mais próximo da área onde existe a demanda. No entanto o déficit global simples é o mesmo em ambas as situações.

A função heurística de déficit global melhorado procura capturar estas informações. A idéia por trás de sua definição é que o volume mínimo a ser movimentado para que o déficit de estoque seja atendido é igual ao próprio déficit mais o que está entre a ocorrência mais próxima do produto e a área do déficit.

Para calcular o déficit melhorado para um déficit de estoque de um produto P em uma área A, computamos a quantidade de produtos diferentes de P entre A e o estoque de P mais próximo de A, esteja ele situado em uma área na origem da rota de fluxo ou em um duto. Para isso, todas as

seqüências de rotas de fluxo que chegam em A são percorridas, até a área de origem, em busca de P.

Caso uma ou mais ocorrências de P sejam encontradas neste processo, a ocorrência mais próxima é considerada, e a distância em  $m^3$  desta ocorrência até A é adicionada ao déficit. Em nossos exemplos, este valor adicionado seria  $10m^3$  para a Figura 8.2 e  $50m^3$  para a Figura 8.4, assumindo a rota  $A6 - D7 - D5 - D3$ .

Caso nenhuma ocorrência de P seja encontrada, o valor adicionado é igual ao volume em  $m^3$  da maior rota que chega em A. É importante notar que esta heurística, como qualquer outra, pode gerar avaliações ruins em algumas situações, como detalharemos na Seção 8.8.2. No processo de desenvolvimento do Plumber 2, o refinamento das heurísticas é feito sob demanda, em função do resultado da análise de exemplos reais nos quais a heurística não teve o funcionamento adequado.

## 8.6

### Objetivos

O Plumber 2 define dois tipos de objetivos. O primeiro, representado pela classe *plumber::time\_goal* na Figura 8.1, é levar o simulador até um instante de tempo igual ou maior a um instante  $t$ . Este tipo de objetivo é utilizado quando desejamos um plano que atenda as ordens de serviço programadas entre o instante inicial de simulação e  $t$ , que formam o horizonte de programação.

Outro objetivo possível, representado pela classe *plumber::zero\_deficit* na Figura 8.1, é chegarmos a um estado cujo déficit global seja igual a zero. Neste estado, a rede encontra-se em uma situação capaz de acomodar todas as produções e atender todas as ofertas programadas.

## 8.7

### Refinamento dos eventos do simulador

Durante os testes realizados com as heurísticas apresentadas nas seções anteriores, percebemos que um ponto de decisão importante é o momento no qual um déficit, de estoque ou de capacidade, é zerado. Como a escolha das ações é guiada por estes eventos, é natural que reanaliseemos esta escolha na medida em que estes déficits são anulados.

Para contemplar isto, a lista de eventos que discretiza a operação do simulador foi estendida, em relação à apresentada na Seção 8.3.1, para gerar



também eventos quando um déficit é anulado. Por exemplo, na Figura 8.3, caso estejamos no instante 00:00 e bombeando o produto P a uma vazão de  $30m^3/h$ , um evento de déficit é programado para ocorrer a 01:00, já que nesta hora o volume excedente já foi bombeado e o déficit de capacidade não existe mais. O planejador então retoma o controle, e pode avaliar se é interessante ou não continuar o bombeamento do produto.

Este é um exemplo de como os eventos que discretizam a simulação podem ser estendidos de acordo com a heurística utilizada, conforme apresentado na introdução deste Capítulo.

## 8.8 Resultados

Esta Seção apresenta os resultados obtidos durante o processo de construção do Plumber. Neste momento, o Plumber é capaz de gerar um plano para um horizonte de 8 dias na rede de claros da Transpetro, com 13 áreas, 25 dutos, 16 produtos e 49 rotas de fluxo. As limitações correntes estão no número de ordens de serviço e na quantidade de rotas que podem ser ativadas simultaneamente.

O processo de testes iniciou com a definição da topologia completa da rede, com todos os seus produtos, áreas, dutos e rotas de fluxo. O estado inicial da rede foi colocado de acordo com uma instância real. De forma a permitir que a avaliação seja feita incrementalmente, optou-se por testar o Plumber sempre com um horizonte de planejamento de 8 dias, acrescentando as ordens de serviço uma a uma e verificando se o Plumber era capaz de gerar um plano.

Como nesta fase o interesse maior está na definição das heurísticas, fixamos a estratégia de busca como sendo a *best first search*. A utilização do PLANSIM no entanto nos garante que outras estratégias podem ser testadas mais tarde sem nenhum impacto no Plumber.

### 8.8.1 Análise do fator de ramificação

O fator de ramificação de um processo de busca é definido pelo número de novos estados gerados na expansão de cada nó. No caso da instância testada, o fator de ramificação ficou na faixa de 140 sucessores em cada estado explorado, quando permitimos a ativação de uma rota de fluxo

por vez. É importante notar que este valor, na arquitetura do Plumber, independe do número de ordens de serviço da instância.

Já com duas rotas podendo ser ativadas por vez, este número sobe para a faixa de 10000. Este alto valor é função do grande número de rotas disponíveis e produtos que podem ser bombeados em cada uma delas. Neste ponto foi detectado que qualquer abordagem que expanda todos os sucessores de um estado está condenada ao fracasso, mesmo com uma excelente função heurística. Não é por acaso que a maioria das abordagens apresentadas na literatura ativam apenas uma rota de fluxo por vez, como vimos na Seção 2.5. Em função deste alto fator de ramificação, os resultados relatados nesta Seção estão restritos à ativação de uma rota de fluxo por vez. Formas de contornar este problema são apresentadas na Seção 8.9.

## 8.8.2 Plano gerado

Esta Seção apresenta o plano de maior complexidade gerado pela versão atual do Plumber. O tempo utilizado na geração do plano foi da ordem de 1 minuto. A heurística utilizada em todos os casos foi a de déficit global melhorado.

No decorrer do desenvolvimento do sistema foi formada um base de testes com instâncias cada vez mais complexas. A cada modificação realizada nas funções heurísticas ou na discretização do processo, estes testes eram executados novamente verificando se a nova configuração do Plumber conseguia resolver todas as instâncias. Desta forma garantimos a não regressão do planejador durante o desenvolvimento.

Cada linha do plano especifica a data de início da operação, a rota que é ativada, o produto bombeado e a vazão utilizada. A ação vazia, apresentada na Seção 8.4, é representada pela operação NOP.

```

1: [ 20041001T000000: activating REVAP-OSVAT22(D)OSPLANRP(R), DIESEL-D, 3600 m3/h ], C: 1
2: [ 20041001T000001: activating REVAP-OSVAT22(D)OSPLANRP(R), DIESEL-D, 3600 m3/h ], C: 2
3: [ 20041001T021820: activating REVAP-OSVAT22(D)OSPLAN18(D), DIESEL-D, 2400 m3/h ], C: 3
4: [ 20041001T105326.081000: activating S_SEBASTIAO-OSPLANTB(D)OSVAT22(R), DIFESP, 3600 m3/h ], C: 4
5: [ 20041001T111926.102000: activating RPBC-R5(D)A6(D), DIESEL-D, 2700 m3/h ], C: 5
6: [ 20041001T161000.716000: activating RPBC-R5(D)A6(D), DIESEL-D, 2700 m3/h ], C: 6
7: [ 20041001T192606.102000: activating S_SEBASTIAO-OSPLANTB(D)OSVAT22(R), DIFESP, 3600 m3/h ], C: 7
8: [ 20041001T195742.101000: activating S_CAETANO-LINHAC(R)R6(R), NAPPQU, 1500m3/h ], C: 8
9: [ 20041001T201003.700000: activating S_CAETANO-LINHAC(R)R6(R), NAPPQU, 1500m3/h ], C: 9
10: [ 20041001T234942.100000: activating S_SEBASTIAO-OSPLANTB(D)OSPLAN18(D), QAV-1, 2700 m3/h ], C: 10
11: [ 20041001T235155.433000: activating S_CAETANO-LINHAC(R)R1(R), NAPPQU, 1500 m3/h ], C: 11
12: [ 20041001T235853.033000: activating S_CAETANO-LINHAC(R)R1(R), NAPPQU, 1500 m3/h ], C: 12
13: [ 20041002T000000: activating S_CAETANO-LINHAC(R)R1(R), NAPPQU, 1500 m3/h ], C: 13
14: [ 20041002T052029.033000: activating S_SEBASTIAO-OSPLANTB(D)OSPLAN18(D), DIFESP, 2700 m3/h ], C: 14
15: [ 20041002T061810.294000: activating S_SEBASTIAO-OSPLANTB(D)OSVAT22(R), DIFESP, 3600 m3/h ], C: 15
16: [ 20041002T070322.293000: activating REVAP-OSVAT22(D)OSPLAN18(D), DIESEL-B,2400 m3/h ], C: 16
17: [ 20041002T074410.293000: activating S_SEBASTIAO-OSPLANTB(D)OSPLAN18(D), DIFESP, 2700 m3/h ], C: 17
18: [ 20041002T100810.292000: activating REVAP-OSVAT22(D)OSPLAN18(D), DIESEL-B,2400 m3/h ], C: 18
19: [ 20041002T105010.291000: activating RPBC-R1(D)A2(D), DIESEL-D, 1500 m3/h ], C: 19
20: [ 20041002T105707.891000: activating RPBC-R1(D)A2(D), DIESEL-D, 1500 m3/h ], C: 20

```

21: [ 20041002T113317.491000: activating SANTOS-A2(R)R6(R), GASOEXP, 3000 m3/h ], C: 21  
 22: [ 20041002T113928.290000: activating RPBC-R1(D)A2(D), DIESEL-D, 1500 m3/h ], C: 22  
 23: [ 20041002T115149.888000: activating RPBC-R9(D)LINHAA(D)LINHAARU(D), GASO-A, 1140 m3/h ], C: 23  
 24: [ 20041002T115749.888000: activating RPBC-R9(D)LINHAA(D)LINHAARU(D), DIESEL-D, 1140 m3/h ], C: 24  
 25: [ 20041002T120349.888000: activating RPBC-R9(D)LINHAA(D)LINHAARU(D)OBATICL(D), DIESEL-D, 1050 m3/h ], C: 25  
 26: [ 20041002T144756.745000: activating RPBC-R9(D)LINHAA(D)LINHAARU(D), DIESEL-D, 1140 m3/h ], C: 26  
 27: [ 20041002T145356.745000: activating RPBC-R9(D)LINHAA(D)LINHAARU(D), DIESEL-D, 1140 m3/h ], C: 27  
 28: [ 20041002T145956.745000: activating RPBC-R9(D)LINHAA(D), DIESEL-D, 1140 m3/h ], C: 28  
 29: [ 20041002T160237.675000: activating RPBC-R9(D)LINHAA(D), DIESEL-D, 1140 m3/h ], C: 29  
 30: [ 20041002T234340.832000: activating RPBC-R9(D)LINHAA(D)LINHAARU(D), DIESEL-D, 1140 m3/h ], C: 30  
 31: [ 20041002T235959.999000: activating RPBC-R9(D)LINHAA(D), DIESEL-D, 1140 m3/h ], C: 31  
 32: [ 20041003T034728.202000: NOP ], C: 31  
 33: [ 20041003T235959.929000: NOP ], C: 31  
 34: [ 20041005T000000: NOP ], C: 31  
 35: [ 20041006T000000: activating RPBC-R9(D)LINHAA(D)LINHAARU(D), NAFPQU, 1140m3/h ], C: 32  
 36: [ 20041006T000600: activating RPBC-R9(D)LINHAA(D)LINHAARU(D), NAFPQU, 1140m3/h ], C: 33  
 37: [ 20041006T001741.052000: NOP ], C: 33  
 38: [ 20041006T235959.907000: NOP ], C: 33

Em um determinado momento identificamos que a incapacidade da versão corrente do Plumber em gerar o plano está ligada a uma fraqueza na heurística de déficit global melhorado. Esta heurística leva em consideração, no caso de um déficit de estoque, a situação corrente do duto e das tancagens nas áreas de origem das rotas analisadas. Ela não vislumbra que, apesar de eventualmente não existir nenhum volume do produto em uma determinada área de origem, a sua produção pode estar programada para breve.

## 8.9 Conclusões

Apresentamos o processo de desenvolvimento do Plumber 2, e seu estágio atual. Entre as principais vantagens do processo apresentado está o seu desenvolvimento incremental, que permite que tanto heurísticas como a definição dos simuladores seja ajustada no decorrer do processo. A utilização do PLANSIM permite que estes experimentos sejam feitos de maneira simples. A criação de novas heurísticas é feita através de uma especialização da classe *evaluation*, e a implementação de novos eventos de simulação, conforme descrito na Seção 8.7, fica restrita ao simulador e não afeta os demais componentes do sistema.

A utilização de um simulador a eventos permite que heurísticas sofisticadas, que inclusive levem em consideração informação temporal, sejam implementadas com relativa facilidade. Existe ainda bastante espaço para a melhoria das heurísticas definidas. Isto somado ao fato de trabalharmos com um processo iterativo, que nos garante uma evolução contínua do planejador e não permite situações de regressão, nos leva a crer que estamos no caminho certo para atingir o objetivo de construir um planejador que escale o suficiente para tratar instâncias reais.

Para contornar a questão do alto fator de ramificação, detalhada na Seção 8.8.1, podemos pensar na extensão do PLANSIM para a incorporação de estratégias como a utilizada pelo SGPlan, analisada na Seção 6.1.3. Podemos inicialmente considerar uma ordem de serviço por vez, procurando encontrar um plano que atenda apenas esta ordem e utilize uma única rota. O plano seria refinado em passos sucessivos, adicionando novas ordens de serviço e rotas, sempre uma por vez, resolvendo eventuais conflitos.

Outra alternativa seria na linha de definição de macro-ações, utilizada pelo Macro-FF. É conhecido que algumas combinações de bombeamentos são mais comumente utilizadas pelos programadores. Esta informação pode ser fornecida de antemão ao Plumber, que geraria os próximos estados apenas para estas bombeamentos, reduzindo desta forma drasticamente o fator de ramificação. Esta abordagem pode ser implementada sem modificações no PLANSIM, apenas redefinindo o conjunto de ações retornado em cada chamada ao método *get\_applicable\_action\_sets*. Esta abordagem é semelhante à utilizada em [4], onde a solução pode ser formada apenas por concatenações de seqüências de bombeamento pré-definidas.