

6

A primeira versão do Plumber

Este capítulo descreve a primeira versão do planejador Plumber e apresenta os resultados obtidos. O Plumber é um planejador específico para os domínios não temporais do Pipesworld 2, ou seja, para as instâncias que não incorporam a noção de tempo e prazos descrita na Seção 3.3.1. O principal objetivo do Plumber é avaliar até que ponto o uso de heurísticas e regras de busca específicas para o domínio influenciam no desempenho dos planejadores.

A Seção 6.1 apresenta uma breve descrição da abordagem utilizada em planejamento clássico, que é a de tratar o problema de planejamento como busca em um espaço de estados. Em conjunto, é realizada uma revisão da literatura sobre as estratégias utilizadas pelos planejadores de propósito geral com melhor desempenho sobre as instâncias do Pipesworld. A Seção 6.2 apresenta a arquitetura do Plumber, as heurísticas e regras de controle implementadas. A Seção 6.3 apresenta e comenta os resultados obtidos pelo Plumber sobre as instâncias do benchmark utilizado na IPC-4, e a Seção 6.4 apresenta as conclusões obtidas pelo desenvolvimento do Plumber, incluindo a motivação para a construção de arcabouços de software para sistemas do gênero.

Os resultados apresentados neste capítulo foram publicados em [30, 36].

6.1

Planejamento como busca, a abordagem clássica

Esta Seção apresenta uma descrição breve de como problemas de planejamento em geral, e o Pipesworld mais especificamente, podem ser modelados como problemas de busca direta (*forward search*) em um espaço de estados. Esta é a estratégia básica adotada pelos resolvidores de propósito geral que apresentaram o melhor desempenho nas duas últimas edições da IPC.

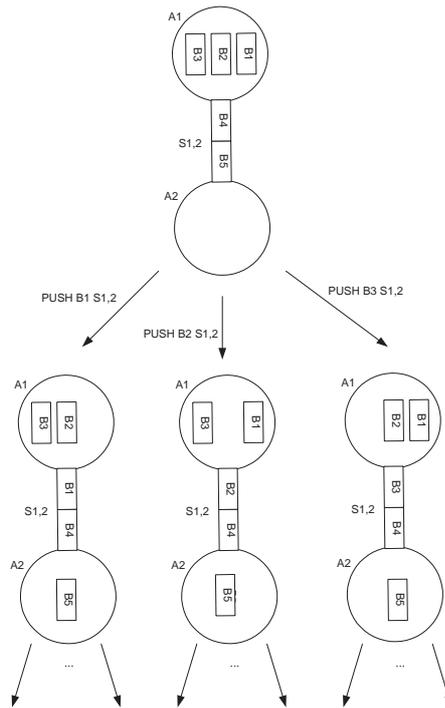


Figura 6.1: Exemplo de procedimento de busca direta no Pipesworld

6.1.1

Um exemplo utilizando o Pipesworld

Consideremos a instância do Pipesworld mostrada na Figura 6.1. Nela temos duas áreas, A_1 e A_2 , conectadas por um único duto $S_{1,2}$, com volume 2. A área A_1 exerce o papel “FROM”, e a área A_2 exerce o papel “TO”. No estado inicial as bateladas B_1 , B_2 e B_3 estão em A_1 , e as bateladas B_4 e B_5 estão dentro de $S_{1,2}$. O objetivo é posicionar B_4 em A_2 . Não existem, nesta instância de exemplo, restrições de interface, tancagem e prazos.

A partir do estado inicial, é possível aplicar três ações: $\text{PUSH } B_1 \ S_{1,2}$, $\text{PUSH } B_2 \ S_{1,2}$ e $\text{PUSH } B_3 \ S_{1,2}$. Estas ações são também mostradas na Figura 6.1, junto com o estado sucessor do inicial que resulta de sua aplicação. Cada novo estado tem um conjunto possível de ações possíveis de serem aplicadas. Este processo, se aplicado indefinidamente, gera um grafo no qual todos os estados alcançáveis são representados, e um caminho que conecta o estado inicial a qualquer estado que satisfaça todos os objetivos define um plano que soluciona o problema.

Desta forma, o problema de planejamento sobre qualquer instância do Pipesworld 1 pode ser visto como um problema de busca sobre o espaço de estados. O procedimento de busca inicia no estado inicial e para quando

um estado que satisfaz todos os objetivos é encontrado¹. No caso mostrado, qualquer operação “PUSH” aplicada a qualquer estado no grafo leva à uma solução, já que o resultado de todas elas é o posicionamento de B_4 em A_2 .

Esta é a chamada abordagem clássica para a solução de problemas de planejamento [29]. Ela não é utilizada tipicamente em problemas de planejamento temporais, como as instâncias do temporais e com prazo do Pipesworld descritas na Seção 3.3.1, embora isto seja possível. O Capítulo 8 mostra como esta abordagem é utilizada na construção de uma versão do Plumber que trata o problema temporal.

6.1.2

Controlando a explosão combinatória

O planejamento como busca é, em última instância, um exercício de controle de explosão combinatória [22]. Como a taxa de crescimento do espaço de estados é tipicamente exponencial, o processo de geração de todos os estados alcançáveis a partir do estado inicial até que um estado objetivo seja atingido está condenado a exaurir todos os recursos de memória disponíveis e/ou levar um tempo proibitivamente longo para achar uma solução. A utilização de heurísticas e estratégias de busca, quando executadas com sucesso, posterga esta explosão o suficiente para que uma solução seja encontrada.

Heurísticas

A maneira mais eficiente de lidar com os efeitos combinatórios no problema de busca é através da definição de uma boa heurística para guiar o processo. Uma heurística para a seleção do próximo nó a ser visitado ordena um conjunto de estados em função de seu potencial para a solução. Uma função heurística h é utilizada para calcular um valor numérico para cada estado u visitado $h(u)$.

Nos planejadores de propósito geral que utilizam PDDL como representação do modelo, a heurística não é fornecida. Isto não quer dizer que estes planejadores não trabalhem com elas, mas sim que eles devem derivar esta heurística automaticamente a partir da descrição do problema. Esta é uma das áreas de grande interesse em planejamento em inteligência artificial. De fato, o desenvolvimento recente de novas formas de derivação destas

¹Opcionalmente, o processo pode continuar em função da necessidade de um plano ótimo e da estratégia de busca utilizada.

heurísticas é o responsável pelo sucesso atual dos planejadores que utilizam busca direta em espaço de estados.

O primeiro passo neste sentido foi dado pelo planejador HSP [14], que deriva as heurísticas resolvendo um problema relaxado no qual as listas de deleção são ignoradas. As listas de deleção são todos os efeitos de uma ação PDDL que levam um predicado de verdadeiro para falso. Por exemplo, no caso do Pipesworld, a operação de bombeamento de uma batelada B_i de um tanque T_j em um duto $S_{1,2}$ tem como efeitos, entre outros, que B_i passa a ser a primeira batelada do duto e B_i passa a não estar mais no tanque de origem. Estas condições são representadas pelos predicados $(first\ B_i\ S_{1,2}) = true$ e $(on\ B_i\ T_j) = false$, respectivamente. O último predicado faz parte da lista de deleção da ação. Se forem ignoradas, B_i passa a poder estar simultaneamente no duto e no tanque de origem. O tamanho da solução para este problema relaxado é utilizado como função heurística para o processo de busca, que é completado com uma simples estratégia de *best first search*, ou seja, o próximo nó a ser expandido é mais promissor já visitado. Esta estratégia simples levou o HSP a superar diversos outros planejadores mais elaborados na IPC-2.

Uma abordagem semelhante é utilizada pelo FF [17], que em adição analisa a estrutura do plano resultante para realizar uma poda no espaço de busca e incrementa a estratégia de busca *hill climbing* para que esta ultrapasse mínimos locais, num procedimento denominado *enforced hill climbing*. O FF superou o HSP e foi um dos melhores planejadores do IPC-3.

Infelizmente, os algoritmos de derivação automática de heurísticas ainda não estão em um estágio que elimine a necessidade, para problemas mais práticos, da informação de heurísticas específicas para o domínio a ser planejado. Estas são definidas para o Pipesworld no Plumber, e mostradas na Seção 6.2.1.

Simetrias

Uma outra maneira promissora de adiar a explosão combinatória é a identificação de simetrias no problema, e levá-las em consideração para realizar a poda do grafo de espaço de estados. O tratamento adequado de simetrias reduz significativamente o fator de ramificação do grafo de estados, agilizando e algumas vezes viabilizando o processo de busca.

Para um exemplo podemos analisar novamente a Figura 6.1. Se assumimos que todas as bateladas são compostas de um mesmo produto, existe uma simetria envolvendo as bateladas B_1 , B_2 e B_3 no estado inicial.

Isto acontece pois nenhuma destas bateladas é atribuída a uma área de destino, são do mesmo produto e estão na mesma área. De fato, podemos simplesmente trocar os seus nomes e da mesma forma o grafo de estados teria a mesma estrutura. Assim sendo, ao invés de gerar os três estados sucessores a partir do estado inicial, podemos simplesmente gerar um estado único, escolhendo uma das bateladas B_1 , B_2 ou B_3 para ser bombeada.

A situação em relação à detecção de simetrias e poda do grafo de estados é semelhante à descrita na Seção 6.1.2 no que diz respeito aos planejadores de propósito geral. Estes devem detectar as simetrias de forma automática, baseados simplesmente na descrição do modelo.

6.1.3

Estratégia de busca

Um planejador automático deve também definir a estratégia a ser utilizada durante a busca. Quando falamos de busca heurística, esta estratégia define principalmente a política utilizada para a escolha do próximo nó a ser visitado em cada iteração, como os novos nós são gerados e como os nós visitados são armazenados. Esta Seção apresenta as estratégias de busca utilizadas pelos resolvedores de propósito geral com melhor desempenho nas últimas edições da IPC.

A estratégia de subida de encosta forçada (SEF), utilizada pelo FF [17] utiliza a função heurística para a avaliação de todos os sucessores do estado corrente, e escolhe como próximo estado a ser visitado o melhor sucessor, exigindo que este possua uma avaliação melhor do que a do estado atual. Se no processo encontramos um estado e_1 sem nenhum sucessor melhor, o algoritmo gera os sucessores de cada sucessor do estado atual, avaliando-os na medida em que são gerados. O procedimento é repetido até que um nó com avaliação melhor e_2 seja encontrado. O caminho percorrido de e_1 até e_2 é adicionado ao plano corrente, e o processo continua até que o objetivo seja encontrado.

O SGPlan foi um dos planejadores mais bem sucedidos no domínio na IPC-4. A estratégia do SGPlan é particionar o problema de planejamento original em vários sub-problemas, encontrando um plano viável para cada um individualmente e resolvendo eventuais conflitos entre as soluções parciais. O sistema procura ordenar os sub-problemas de forma que os mais “difíceis” sejam resolvidos antes. O sucesso do SGPlan no Pipesworld indica que este tipo de abordagem pode ser interessante para aplicação no domínio

real, onde várias rotas de fluxo podem estar ativas simultaneamente e, em muitos dos casos, sem interferências entre elas.

Já o YAHSP, que também teve um bom desempenho em relação ao número de instâncias resolvidas, adota a seguinte abordagem. Para um determinado estado corrente, o planejador gera os estados imediatamente vizinhos aplicando as ações possíveis, e também gera estados *look-ahead*, que são obtidos não a partir da aplicação de uma ação no estado corrente mas sim de uma seqüência de ações. Isto explica os planos muito longos retornados por este planejador.

O planejador FDD apresenta uma forma inovadora de combinação de funções heurísticas. Este planejador trabalha com duas funções heurísticas simultaneamente, e mantém duas listas com os estados ainda não explorados, que são expandidas alternadamente. Todos os novos estados gerados na expansão são avaliados pelas duas heurísticas e colocados em ambas as listas. O objetivo é que uma heurística auxilie a outra quando uma delas encontra um mínimo local.

O planejador Macro-FF introduz o conceito de macro-ações. Estas ações são compostas por uma seqüência de ações que podem ser aplicadas durante a expansão de um estado, em adição às ações simples. A estratégia é muito parecida com a utilizada pelo YAHSP.

6.1.4

O papel dos simuladores a eventos discretos

O procedimento de busca descrito até aqui demanda que as seguintes operações sejam efetuadas da forma mais eficiente possível: gerar as ações possíveis de serem aplicadas em cada estado, atualizar o estado após a aplicação da ação, calcular o resultado da função heurística de um determinado estado e identificar simetrias no modelo.

Uma forma eficiente de implementar estas operações é através da construção de um simulador a eventos discretos do domínio a ser planejado. Um dos problemas encontrados quando a abordagem para construção do planejador utiliza técnicas de programação inteira, ou mesmo planejadores que trabalham sobre definições em PDDL, é que o problema analisado deve antes ser modelado em um problema de programação inteira ou em PDDL. Tipicamente, esta modelagem não é direta nem intuitiva, com a situação se agravando mais ainda quando tentamos representar cenários e aplicações reais. É comum, conforme relatado nas seções 2.5 e 3.4.1, que o problema

original tenha que ser bastante simplificado apenas por conta de restrições do modelo do resolvidor que está sendo utilizado.

Um fato agravante deste processo é que ele por vezes introduz simetrias que não fazem parte do problema original, sendo geradas unicamente pela adaptação na modelagem. Um exemplo disto é a situação de simetria de bateladas no Pipesworld, descrita na Seção 6.1.2. Neste caso a simetria é originada pelo fato de o modelo utilizar bateladas unitárias (Seção 3.2.1), que não existem no problema original.

Quando um simulador a eventos discretos é construído sob medida para o problema, é mais fácil aproximar o modelo do sistema que se deseja planejar. A construção do conjunto de ações aplicáveis em um determinado estado é na maioria das vezes direta, sem problemas como os descritos na Seção 3.3.1. Desta forma, é possível que o planejador fique focado na inteligência necessária para a solução do problema, como por exemplo, na definição das heurísticas e estratégias de busca utilizadas.

6.2 Arquitetura

Esta Seção apresenta a arquitetura do Plumber, um planejador dedicado aos domínios não temporais do Pipesworld 2. O Plumber incorpora um simulador de dutos e utiliza as mesmas técnicas de planejamento dos planejadores de propósito geral de melhor desempenho nas últimas versões da IPC. O Plumber também define heurísticas específicas para o domínio, e agiliza o processo de busca identificando e removendo simetrias no problema. Tanto as heurísticas quanto as estratégias de busca podem ser selecionadas pelo usuário.

Nas seções seguintes, descrevemos as heurísticas e estratégias de busca implementadas pelo Plumber, e como as simetrias são detectadas e tratadas.

6.2.1 Heurísticas

O Plumber define duas heurísticas bastante simples, descritas a seguir. Representamos a topologia da rede como um grafo G_t , onde cada área é representada por um nó e cada duto é representado por uma aresta. Cada aresta possui um peso que é igual ao volume do duto que ela representa, em termos de bateladas unitárias, mais 1. Esta adição é justificada mais tarde.

Inicialmente computamos, para cada par de áreas $(a_i, a_j) \in G_t$ o caminho mais curto entre elas. Chamamos o tamanho deste caminho $sp(a_i, a_j)$. Esta informação é então utilizada para calcular, para todas as bateladas b_k com uma área objetivo $g(b_k)$, a sua menor distância ao objetivo. Se b_k está na área a_k , esta distância é simplesmente

$$D(b_k) = sp(a_k, g(b_k)) \quad (6-1)$$

Se b_k está dentro de um duto $S_{i,j}$ que conecta as áreas a_i e a_j , deve-se levar em consideração também a distância entre b_k e a_i e a_j . Vamos chamar esta distância $d(b_k, a)$. Nós então temos que

$$D(b_k) = \min\{ \begin{aligned} &sp(a_i, g(b_k)) + d(b_k, a_i), \\ &sp(a_j, g(b_k)) + d(b_k, a_j) \end{aligned} \} \quad (6-2)$$

Heurística de somatório de distâncias

Seja BG o conjunto de todas as bateladas não proteláveis (Seção 3.2.1). A função heurística de somatório de distâncias HDS para um estado é definida como

$$HDS = \sum_{b_k \in BG} D(b_k) \quad (6-3)$$

Podemos verificar que HDS é zero apenas em um estado no qual todos as bateladas estão posicionadas em seu destino. Apesar de ser simples, a heurística definida na Equação 6-3 é bastante informativa. É razoável considerar estados com valores menores de HDS mais próximos a um estado objetivo, mas este não é sempre o caso.

Para um exemplo vamos analisar a Figura 6.1 novamente. Digamos que, a partir do estado inicial, desejemos posicionar B_4 em A_1 . Para este caso teríamos $HDS = 1$. Mas B_4 não pode ser posicionada em A_1 em uma única operação pois não existem bateladas em A_2 para movimentar o duto. Claramente, se tivéssemos uma batelada em A_2 teríamos uma situação melhor, mas este fato não é capturado pela heurística de somatório de distâncias.

Heurística de somatório de distâncias - MP

Para endereçar este problema, definimos uma nova função heurística denominada somatório de distâncias - MP. Inicialmente definimos, para cada batelada b_k em um duto $S_{i,j}$ que conecta a_i a a_j , um déficit $f(b_k, a_l)$ para a batelada b_k chegar à área a_l . Seja $n(a_l)$ o número de bateladas em a_l . Definimos $f(b_k, a_i)$ como:

$$f(b_k, a_i) = \begin{cases} 0 & \text{if } n(a_j) \geq d(b_k, a_i) \\ 2 * (d(b_k, a_i) - n(a_j)) & \text{caso contrário} \end{cases}$$

A Equação 6-2 é então reescrita para levar este déficit em consideração, e obtemos

$$D_{mp}(b_k) = \min \left\{ \begin{aligned} &sp(a_i, g(b_k)) + d(b_k, a_i) + f(b_k, a_i), \\ &sp(a_j, g(b_k)) + d(b_k, a_j) + f(b_k, a_j) \end{aligned} \right\} \quad (6-4)$$

A função heurística somatório de distâncias - MP, $HDSMP$, é definida como

$$HDSMP = \sum_{b_k \in BG} D_{mp}(b_k) \quad (6-5)$$

6.2.2 Estratégias de busca implementadas

O Plumber permite ao usuário selecionar uma das seguintes estratégias de busca: subida de encosta forçada, apresentada na Seção 6.1.3, subida de encosta com movimentos laterais ou A* com aprendizado em tempo real.

O algoritmo de subida de encosta com movimentos laterais (SEL) difere do SEF no sentido que se um mínimo local é encontrado o algoritmo escolhe como próximo estado qualquer um que tenha ao menos o mesmo valor para a função heurística. Isto possibilita ao algoritmo tratar de forma efetiva os platôs, ou seja, regiões do espaço com o mesmo valor para a função heurística.

O algoritmo de A* com aprendizado em tempo real (ATR) [8] é semelhante ao SEL, mas com uma diferença importante: o valor da função heurística de cada estado é atualizado durante a execução do algoritmo, daí o aprendizado no nome do algoritmo.

6.2.3 Simetrias

O Plumber evita a geração de alguns estados simétricos utilizando o seguinte procedimento. Quando gerando todos os possíveis sucessores de um estado, o sistema inicialmente verifica se a batelada b_k que está sendo bombeada em um duto d_i é protelável ou não. Se b_k não é protelável, um estado sucessor é gerado. Caso contrário, o Plumber gera um sucessor apenas se nenhuma outra batelada do mesmo produto e localizada na mesma área que b_k foi bombeada em d_i na geração de algum sucessor. Este procedimento garante que estados simétricos como os mostrados na Figura 6.1 não são gerados durante o procedimento de busca.

6.3 Resultados

Nesta Seção reportamos os resultados experimentais obtidos pelo Plumber. Inicialmente, comparamos uma versão do Plumber com o FF, para avaliar como a inclusão das heurísticas específicas descritas na Seção 6.2.1 comparam com funções heurísticas derivadas automaticamente. Mais tarde as heurísticas e estratégias de busca do Plumber são comparadas entre si.

A configuração de hardware utilizada para a execução de todos os testes foi um PC com processador AMD XP de 2.4 GHz, e com 512MB de RAM.

6.3.1 Comparação com planejador de propósito geral

Nesta Seção comparamos o desempenho do Plumber e do FF sobre o conjunto *no-tankage* descrito na Seção 4.2.1. O Plumber é executado com a mesma estratégia de busca utilizada pelo FF, a SFE, e com a heurística de somatório de distâncias. O objetivo é avaliar como heurísticas simples e específicas do domínio comparam-se com as derivadas automaticamente. Os planos resultantes do Plumber e do FF são comparados tanto em termos de tempo total de execução e tamanho do plano resultante.

Os tempos de execução medidos são mostrados na Tabela 6.1. Os valores reportados são termos de tempo de CPU, em segundos. O tempo máximo dado para resolução de cada instância foi de 10 minutos. Instâncias não resolvidas dentro deste intervalo aparecem com um traço na tabela.

Tabela 6.1: Comparação do tempo de execução, Plumber x FF

Instância	FF	Plumber	Instância	FF	Plumber
01	0.00	0	26	-	13.35
02	0.00	0.09	27	34.47	3.91
03	0.01	0.05	28	-	4.99
04	0.01	0.17	29	26.67	3.06
05	0.00	0.12	30	8.77	1.68
06	0.01	0.2	31	12.52	3.83
07	0.00	0.01	32	8.03	4.8
08	0.01	0.02	33	-	21.22
09	0.02	0.02	34	-	6.07
10	0.03	0.17	35	-	-
11	0.20	0.29	36	-	86.74
12	0.04	0.48	37	-	322.24
13	1.91	0.06	38	-	278.51
14	0.34	0.39	39	-	11.9
15	0.30	0.15	40	-	38.25
16	2.29	1.29	41	-	0.24
17	38.99	0.23	42	-	-
18	0.27	0.21	43	-	-
19	0.04	0.29	44	-	-
20	0.06	0.03	45	-	-
21	0.86	0.31	46	-	-
22	-	-	47	-	-
23	0.32	4.03	48	-	-
24	10.46	251.63	49	6.21	6.19
25	0.98	1.81	50	11.96	15.48

Podemos ver pela tabela que o desempenho dos planejadores é praticamente equivalente para as primeiras 20 instâncias. A diferença fica aparente para as instâncias de 21 a 41, onde o Plumber exibiu um desempenho melhor. Nesta faixa, o Plumber reportou uma solução em menos tempo do que o FF em 16 das 19 instâncias. Podemos também verificar que as instâncias 42 a 48 foram difíceis para ambos os planejadores.

A Tabela 6.2 mostra o número de ações do plano reportado por cada um dos resolvedores. O Plumber foi melhor do que o FF também neste aspecto, retornando um plano maior do que o fornecido pelo FF apenas para 4 das 41 instâncias. No caso específico da instância 24 o Plumber encontrou uma solução mais longa, o que justifica o maior tempo utilizado para a obtenção da solução.

Tabela 6.2: Comparação do tamanho do plano, FF x Plumber

Instância	FF	Plumber	Instância	FF	Plumber
01	5	5	26	-	33
02	12	12	27	28	18
03	9	9	28	-	26
04	11	11	29	25	22
05	9	9	30	43	25
06	13	11	31	27	20
07	12	8	32	32	28
08	13	10	33	-	28
09	24	14	34	-	32
10	28	21	35	-	-
11	12	12	36	-	28
12	16	16	37	-	26
13	14	8	38	-	31
14	21	15	39	-	23
15	23	13	40	-	29
16	43	32	41	-	6
17	13	11	42	-	-
18	31	19	43	-	-
19	15	12	44	-	-
20	18	16	45	-	-
21	11	9	46	-	-
22	-	-	47	-	-
23	19	26	48	-	-
24	26	34	49	27	28
25	24	25	50	38	38

6.3.2

Comparação das heurísticas e estratégias de busca

Inicialmente, comparamos o número de instâncias resolvidas pelo Plumber para cada uma das combinações das estratégias de busca e heurísticas definidas. O Plumber foi executado por no máximo 10 segundos para resolver cada instância do benchmark do IPC-4. Como o Plumber não suporta domínios temporais, foram utilizadas apenas os conjuntos sem tempo, descritos nas seções 4.2.1 e 4.2.2.

A Tabela 6.3 mostra o número de instâncias sem tancagem resolvidas pelo Plumber em menos de 10 segundos, para as três estratégias de busca e as duas heurísticas. Podemos observar que a heurística de somatório de distância - mp resultou em um desempenho melhor que a de somatório de distâncias, independente da estratégia de busca selecionada. Além disso, o estratégia de busca SEL foi melhor que a SEF, e a ATR nunca foi pior que

Tabela 6.3: Instâncias sem tancagem resolvidas

som. de distâncias			som. distâncias mp		
sef	sel	atr	sef	sel	atr
38	38	42	41	47	47

Tabela 6.4: Instâncias com tancagem resolvidas

som. de distâncias			som. de distâncias mp		
sef	sel	atr	sef	sel	atr
24	33	33	26	34	38

a SEL.

A Tabela 6.4 reporta o número de instâncias com tancagem resolvidas. A primeira observação é que o número total de instâncias resolvidas é menor que o reportado na Tabela 6.3. Isto é esperado já que nenhuma das heurísticas propostas leva a restrição de tancagem em consideração. De maneira geral, as mesmas observações feitas para o conjunto sem tancagem valem para o conjunto com tancagem, em relação ao desempenho das heurísticas.

Agora comparamos a qualidade das soluções encontradas pelas diferentes heurísticas e estratégias de busca, em termos de tamanho total do plano. Já que o maior custo na operação de uma rede de oleodutos é o custo de bombeamento, é desejável que obtenhamos planos com o menor número possível de ações.

A Tabela 6.5 e a Tabela 6.6 mostram o tamanho do plano retornado para cada uma das instâncias sem tancagem, utilizando as duas heurísticas definidas. Para cada instância, o menor plano retornado é destacado. Podemos observar que o ATR exibe um desempenho melhor também neste aspecto, retornando um plano pior que o do SEF e do SEL para apenas três casos na Tabela 6.5 e seis casos na Tabela 6.6. Nesta última, o ATR exibiu um desempenho equivalente ao do SEL.

Podemos também notar que o plano retornado para as instâncias 42, 43 e 44 na Tabela 6.6 possuem um comprimento que é na média bem maior do que os planos encontrados para as outras instâncias. Isto indica que estas instâncias são mais difíceis, e pode haver espaço para melhoria da

Tabela 6.5: Tamanho do plano, sem tancagem, som. de distâncias

Instância	sef	sel	atr	Instância	sef	sel	atr
1	5	5	5	26	-	21	21
2	12	-	12	27	16	14	14
3	9	8	8	28	23	19	19
4	11	11	11	29	22	20	20
5	9	9	9	30	28	22	23
6	11	11	11	31	15	14	14
7	8	8	8	32	22	20	20
8	10	10	10	33	-	-	49
9	14	13	13	34	-	21	21
10	21	18	18	35	15	15	16
11	14	-	10	36	-	43	33
12	16	14	14	37	-	-	-
13	8	8	8	38	-	32	38
14	16	15	15	39	20	18	18
15	13	13	13	40	24	20	20
16	29	33	28	41	6	6	6
17	11	11	11	42	-	-	-
18	21	19	19	43	-	-	-
19	12	12	12	44	-	-	-
20	16	14	14	45	26	50	26
21	9	9	9	46	39	-	-
22	28	-	-	47	-	-	-
23	14	12	12	48	-	-	-
24	-	-	39	49	21	21	21
25	21	20	18	50	31	30	30

solução. De fato, nenhuma destas instâncias foi resolvida quando a heurística somatório de distâncias foi utilizada, independente da estratégia de busca selecionada.

A Tabela 6.7 e a Tabela 6.8 reportam o tamanho do plano encontrado para cada uma das instâncias com tancagem. De forma geral, os mesmos comentários feitos para as instâncias sem tancagem se aplicam.

6.4 Conclusões

Os resultados obtidos pela primeira versão do Plumber demonstram que heurísticas específicas do domínio, mesmo bem simples, já são capazes de superar as heurísticas derivadas pelos planejadores automáticos quando utilizamos a mesma estratégia de busca.

Tabela 6.6: Tamanho do plano, sem tancagem, som. de distâncias mp

Instância	sef	sel	atr	Instância	sef	sel	atr
1	5	5	5	26	29	21	21
2	12	-	12	27	21	14	14
3	9	8	8	28	28	19	19
4	11	11	11	29	22	17	17
5	9	9	9	30	26	22	22
6	11	11	11	31	14	13	13
7	8	8	8	32	20	18	19
8	10	10	10	33	-	56	45
9	14	13	13	34	31	21	21
10	20	18	18	35	13	13	13
11	12	10	10	36	-	29	31
12	16	14	14	37	-	22	22
13	12	9	9	38	22	26	22
14	16	15	15	39	20	19	18
15	13	13	13	40	24	20	20
16	29	35	28	41	6	6	6
17	11	11	11	42	-	83	-
18	21	19	19	43	-	68	66
19	12	12	12	44	-	77	88
20	16	14	14	45	29	27	27
21	7	7	7	46	34	30	33
22	28	18	18	47	-	-	-
23	15	12	12	48	-	-	-
24	-	17	18	49	21	21	21
25	21	18	18	50	31	30	32

Em função disto, os planejadores automáticos tem mais a contribuir para a construção de nosso resolvidor no que diz respeito às estratégias de busca utilizadas. Este é um campo fértil na área de planejamento em IA, com novas estratégias sendo propostas regularmente. Esta Seção analisou alguma das estratégias mais bem sucedidas nas instâncias do Pipesworld durante a IPC-4. É desejável, durante o desenvolvimento do planejador específico, que várias destas estratégias sejam experimentadas.

Como o objetivo da tese é a construção de planejadores capazes de lidar com problemas na escala operacional, neste ponto optou-se pela construção de outro planejador, cujo modelo do processo fosse mais próximo do problema real.

Durante o desenvolvimento do Plumber, no entanto, a realização dos testes necessários com diversas funções heurísticas e estratégias de busca se mostrou mais caro na medida em que o sistema crescia. Sem

Tabela 6.7: Tamanho de plano, tancagem, som. de distâncias

Instância	sef	sel	atr	Instância	sef	sel	atr
1	5	5	5	26	-	21	22
2	16	-	16	27	16	14	14
3	9	8	8	28	-	-	-
4	11	11	11	29	22	17	17
5	9	9	9	30	28	23	23
6	11	11	11	31	-	-	-
7	8	8	8	32	-	26	24
8	15	11	11	33	-	-	56
9	-	-	-	34	-	21	21
10	24	-	21	35	15	18	15
11	-	-	-	36	-	36	29
12	-	-	-	37	-	-	-
13	8	8	8	38	-	24	-
14	-	-	-	39	20	18	18
15	-	-	40	40	24	20	20
16	29	34	29	41	7	7	7
17	-	-	-	42	-	-	-
18	21	19	19	43	-	-	-
19	-	14	15	44	-	-	-
20	16	14	14	45	-	37	35
21	9	9	9	46	-	-	90
22	28	-	-	47	-	-	-
23	-	-	-	48	-	-	-
24	-	-	-	49	23	23	22
25	21	19	19	50	-	29	30

um planejamento cuidadoso, a tendência é que aspectos relacionados ao simulador do processo, funções heurísticas e estratégias de busca fiquem acoplados no software, o que dificulta a troca dos mesmos. Por exemplo, não seria trivial no Plumber a mudança do simulador para um mais aderente ao problema operacional, no qual as ações não fossem PUSH/POP e as bateladas não fossem unitárias.

Neste momento detectou-se a necessidade de um arcabouço de software que nos permitisse realizar testes com heurísticas, estratégias de busca e diferentes modelos para os simuladores de uma forma mais fácil. Esta é a motivação para o PLANSIM, apresentado no capítulo seguinte.

Tabela 6.8: Tamanho do plano, tancagem, som. de distâncias mp

Instância	sef	sel	atr	Instância	sef	sel	atr
1	5	5	5	26	31	21	21
2	16	-	16	27	21	14	14
3	9	8	8	28	-	-	-
4	11	11	11	29	22	17	17
5	9	9	9	30	26	23	22
6	11	11	11	31	-	-	-
7	8	8	8	32	-	32	26
8	15	11	11	33	-	-	43
9	-	-	-	34	31	21	21
10	24	-	29	35	13	13	13
11	-	-	-	36	-	31	30
12	-	-	-	37	-	27	22
13	12	9	9	38	-	22	23
14	-	-	-	39	20	18	19
15	-	-	26	40	24	20	20
16	29	38	28	41	7	7	7
17	-	-	-	42	-	79	-
18	21	19	19	43	-	58	59
19	-	15	14	44	-	-	-
20	16	14	14	45	-	29	29
21	7	7	7	46	-	38	56
22	28	18	18	47	-	-	84
23	-	-	-	48	-	-	-
24	-	-	-	49	23	24	23
25	21	20	18	50	-	30	30