

FERNANDO MOREIRA SETTE

**PLANEJAMENTO DO SUPRIMENTO DE PETRÓLEO EM TERMINAIS
PORTUÁRIOS**

FICHA CATALOGRÁFICA

Sette, Fernando Moreira

**Planejamento do suprimento de petróleo em terminais portuários / F.M. Sette. -- São Paulo, 2007.
110 p.**

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

**1.Petróleo 2.Terminais de carga 3.UML 4.Inteligência artificial
I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos II.t.**

SUMÁRIO

LISTA DE FIGURAS	6
LISTA DE TABELAS	7
RESUMO.....	9
ABSTRACT.....	10
1. INTRODUÇÃO	11
1.1. Motivação do Estudo.....	11
1.2. Revisão Bibliográfica	12
1.3. Objetivos	14
1.4. Estrutura do Trabalho	15
2. DESCRIÇÃO DO PROBLEMA	17
2.1. Rede de Distribuição	17
2.2. Especificações Gerais.....	18
2.2.1. Tipos de Óleo Cru	19
2.2.2 Oleodutos	20
2.2.3. Tanques de Armazenamento	20
2.2.4. Recebimento de Óleo Cru no GEBAST	22
2.2.5. Armazenamento de Óleo Cru nas Refinarias	23
3. PLANEJAMENTO AUTOMÁTICO E UML	24
3.1. Planejamento Automático	24
3.1.1. Conceito	24
3.1.2. PDDL – Planning Domain Definition Language.....	27
3.1.2.1. Visão Geral da Sintaxe.....	28
3.1.2.2. Definição de Domínios	28
3.1.2.3. Definição de Ações	30
3.1.2.4. Definição do Problema de Planejamento	31
3.2. Modelagem com UML	32
3.2.1. UML – Unified Modeling Language	32
3.2.1.1. Diagramas de Caso de Uso	33
3.2.1.2. Diagramas de Classes	33

3.2.1.3. Diagramas de Estado.....	34
3.2.1.4. Diagramas de Objetos ou Snapshots.....	34
3.2.1.5. OCL – Object Constraint Language	35
3.2.2. Utilizando a UML em uma Abordagem de Planejamento	36
4. MODELAGEM DO PORTO.....	39
4.1. Visão Geral	40
4.2. Hipóteses do Modelo de Porto	41
4.3. Casos de uso.....	44
4.4. Classes	49
4.5. Diagrama de estados.....	52
4.5.1. Navio	52
4.5.2. Píer.....	54
4.5.3. Tanque	54
4.5.4. Oleoduto.....	55
5. RESULTADOS DO MODELO DE PORTO	59
5.1. Modelo Sequencial (sem tempo)	59
5.1.1. Escolha dos Planejadores.....	60
5.1.2. Cenários	61
5.1.2.1. Cenário 1.....	61
5.1.2.2. Cenário 2.....	62
5.1.3. Resultados do Modelo Sequencial	62
5.1. Modelo de Porto com Tempo.....	65
5.2.1. Escolha dos Planejadores.....	66
5.2.2. A Métrica Utilizada	67
5.2.2. Exemplo 1	69
5.2.2.1. Estado inicial e Dados de Entrada	70
5.2.2.1 Resultados do Exemplo 1.....	73
5.2.3. Exemplo 2	77
5.2.3.1. Estado inicial e Dados de Entrada	78
5.2.3.1. Resultados do exemplo 2.....	81
6. CONCLUSÃO E PROXIMOS PASSOS.....	87

ANEXO A	93
A.1 Atracar no píer.....	93
A.2 Desatracar no píer.....	94
A.3 Receber determinado óleo cru de um navio em um tanque	95
A.4 Conectar um tanque a um navio.....	96
A.5 Bombear óleo do navio para o tanque.....	97
A.6 Desconectar um tanque de um navio	98
A.7 Distribuir óleo cru para a refinaria a partir de um tanque.....	99
A.8 Conectar um tanque e uma refinaria através de um oleoduto.....	100
A.9 Bombear óleo cru através do oleoduto.....	101
A.10 Desconectar um tanque e uma refinaria de um oleoduto	102
A.11 Consumir óleo cru	103
ANEXO B.....	104
B.1 Modelo de Porto em PDDL.....	104

LISTA DE FIGURAS

Figura 2.1 - Rede de distribuição da Petrobrás no Estado de São Paulo.....	17
Figura 3.1 - Esquema de um sistema de planejamento automático dinâmico.....	24
Figura 3.2 - Sintaxe de definição de domínios em PDDL.....	29
Figura 3.3 - Sintaxe de definição de ações em PDDL.....	30
Figura 3.4 - Sintaxe de definição de problemas em PDDL.....	31
Figura 3.5 - Esquema geral do ambiente de planejamento.....	35
Figura 4.1 - Entradas e saídas do modelo proposto.....	39
Figura 4.2 - Caso de Uso <i>Atracar no píer</i>	44
Figura 4.3 - Diagrama de casos de uso.....	47
Figura 4.4 - Diagrama de classes.....	50
Figura 4.5 - Diagrama de estados do Navio.....	52
Figura 4.6 - Diagrama de estados do Píer.....	53
Figura 4.7 - Diagrama de estados do Tanque.....	55
Figura 4.8 - Diagrama de estados do Oleoduto.....	56
Figura 4.9 Diagrama de estados do Porto.....	57
Figura 5.1 - Classe <i>Domain_Metrics</i>	68
Figura 5.2 - Ilustração do exemplo 1.....	69
Figura 5.3 - <i>Snapshot</i> do estado inicial do exemplo 1.....	70
Figura 5.3 - <i>Snapshot</i> do estado inicial do exemplo 1 (cont.).....	71
Figura 5.5 - Plano gerado na segunda iteração para o exemplo 1.....	73
Figura 5.6 - Inventario dos tanques, em m ³ , para a solução do exemplo 1.....	74
Figura 5.7 - Inventario dos navios, em m ³ , para a solução do exemplo 1.....	75
Figura 5.7 - Inventário das refinarias, em m ³ , para a solução do exemplo 1.....	75
Figura 5.8 -Tempo que cada navio permaneceu atracado nos píeres na solução do exemplo 1 vs. Tempo permitido.....	76
Figura 5.9 - Ilustração do exemplo 2.....	77
Figura 5.10 - <i>Snapshot</i> do estado inicial do exemplo 1.....	78
Figura 5.11 - <i>Snapshot</i> do estado inicial do exemplo 1 (cont.).....	79
Figura 5.12 - Plano gerado na segunda iteração para o exemplo 2.....	81
Figura 5.13 - Plano gerado na segunda iteração para o exemplo 2 (cont.).....	82

Figura 5.14 - Inventario dos tanques, em m ³ , para a solução do exemplo 2.....	83
Figura 5.15 - Inventario dos navios, em m ³ , para a solução do exemplo 2.....	84
Figura 5.16 - Inventário das refinarias, em m ³ , para a solução do exemplo 2.....	85
Figura 5.17 -Tempo que cada navio permaneceu atracado nos píeres na solução do exemplo 2 vs. Tempo permitido.....	85

LISTA DE TABELAS

Tabela 2.1 - Associação entre classes e tipos de óleo cru.....	18
Tabela 2.2 - Descrição dos oleodutos do problema.....	19
Tabela 2.3 - Tanques de armazenagem de óleo cru do GEBAST.....	20
Tabela 5.1 - Problema realista sem métrica.....	63
Tabela 5.2 - Problema realista com métrica.....	64
Tabela 5.3 - Problema simples com variação no volume do inventário dos navios.....	64
Tabela 5.4 - Dados das soluções do Exemplo 1.....	72
Tabela 5.5 - Dados das soluções do Exemplo 2.....	81

RESUMO

Aumenta cada vez mais a importância de um cronograma de atividades eficiente das mais diversas operações industriais, como as de produção, transporte, armazenamento, entre outras. No contexto dos complexos de suprimento de petróleo, isto está evidenciado através do grande número de empresas importadoras de petróleo, das distâncias entre os complexos de suprimento e refino de petróleo e os respectivos poços de exploração e da necessidade de instalar refinarias em regiões distantes dos portos de recebimento. No entanto, ainda não existe nenhum algoritmo robusto e confiável que seja capaz de resolver este problema. O objetivo deste trabalho é desenvolver e analisar um modelo da operação de complexos de suprimento de petróleo utilizando UML (Unified Modeling Language) e técnicas de Planejamento Automático em IA, através da linguagem PDDL (Planning Domain Definition Language). Além disso, serão apresentados resultados de testes de performance dos planejadores automáticos atuais desenhados para avaliar a capacidade das técnicas mais recentes de planejamento automático de lidar com o desafio de resolver um problema real de alta complexidade.

ABSTRACT

The efficient planning of industrial activities, such as production, transportation, storage, among others, is becoming more and more important in modern times. In the context of oil supply complexes, this becomes clear through three facts: the great number of companies which import oil, that travels in oil tankers; the distance between supply complexes, refineries and exploration sites (even more for companies which make extensive use of offshore platforms); and the need to install refineries, for strategic or economic reasons, in locations far from the ports (justifying the use of pipelines to connect them). A great number of applications and mathematical models has been presented in the last decade to solve the planning problems involving these complexes. However, there is no algorithm that is robust and reliable enough to solve this problem. The objective of this work is to develop a model of the operations of oil supply complexes using UML and Automated Planning techniques (through the PDDL language). We will present the experience, results and issues that emerged from testing the performance of the recent planners when solving a real and complex problem such as the planning of daily activities of a petroleum plant for docking, storing and distributing oil.

1. INTRODUÇÃO

Este capítulo procura fornecer uma breve introdução ao escopo do trabalho, que, basicamente, utiliza técnicas de planejamento automático aplicadas a problemas de recebimento e distribuição de petróleo em complexos petroquímicos interligados por oleodutos.

Primeiramente, um breve relato das principais motivações do trabalho será apresentado; em seguida, procura-se resumir trabalhos e pesquisas anteriormente relacionadas à área e correlacionados com este trabalho. Finalmente, serão apresentados os principais objetivos, estratégias e ferramentas de resolução que serão utilizadas.

1.1. Motivação do Estudo

Aumenta cada vez mais a importância de um cronograma de atividades eficiente das mais diversas operações industriais, como as de produção, transporte, armazenamento, entre outras. Empresas competitivas, em geral, dominam tecnologias muito semelhantes em termos de eficiência e produtividade. Portanto, o grande diferencial estratégico entre empresas mais e menos bem sucedidas é a utilização inteligente de seus recursos para maximização de lucros.

As operações de transporte em uma empresa são consideradas, muitas vezes, como os principais gargalos na cadeia produtiva. Estas operações devem ocorrer em harmonia com as demais ações da empresa, ou podem incorrer em perdas significativas. Quando as operações de transporte atrasam, a cadeia produtiva fica sem matéria-prima e a empresa fica incapacitada de produzir e atender seus clientes. Por outro lado, se superdimensionada, estas operações podem levar à criação de estoques muito volumosos, o que significa capital imobilizado e improdutivo em excesso.

No contexto dos complexos de suprimento de petróleo, a necessidade de operações de transportes eficientes está evidenciada por três fatos: o grande número de empresas importadoras de petróleo, que provém através de navios petroleiros; a distância entre os

complexos de suprimento e refino de petróleo e os respectivos poços de exploração, fator relevante em empresas que extraem petróleo de poços em alto mar; e a necessidade de instalar refinarias, por razões econômicas ou estratégicas, em regiões distantes dos portos de recebimento, justificando a implementação de redes de oleodutos que as conectem.

O problema abordado neste trabalho considera o complexo de suprimento de petróleo como sendo constituído de terminais portuários de recebimento, refinarias de processamento e uma estrutura de oleodutos capaz de enviar diversas classes de petróleo dos portos para as refinarias. O terminal possui uma estrutura de recebimento composta por píeres e tanques de armazenagem.

Um grande número de aplicações de modelos matemáticos na área de programação foi apresentado na última década para resolver o problema do planejamento desses complexos. No entanto, ainda não existe nenhum algoritmo robusto e confiável que seja capaz de resolver este problema (LI *et al.* 2005). Além disso, o planejamento dessas operações é muito importante para o funcionamento das refinarias, além de constituir um problema de difícil modelagem matemática (DAHALL *et al.* 2003).

Diante destes fatos motivadores, este trabalho é apresentado, abordando o problema a partir de uma técnica até então pouco explorada nessa área: o planejamento automático. Esta abordagem permite a modelagem do problema em níveis mais altos, sem que seja necessário realizar simplificações matemáticas ou construir modelos matemáticos complexos. Além disso, os modelos gerados por esta nova abordagem podem ser facilmente adaptados a mudanças no sistema.

1.2. Revisão Bibliográfica

Na literatura, praticamente todos os trabalhos relacionados ao suprimento de petróleo encontram-se na área da pesquisa operacional. Esta área é tradicionalmente associada a problemas de otimização e tomada de decisão. A seguir serão descritos alguns trabalhos dessa área que lidam com este problema.

O problema do Terminal Portuário de São Sebastião (GEBAST) foi descrito em (MÁS, 2003) que utiliza uma abordagem de programação mista inteira linear (MILP)

para encontrar soluções. Neste artigo é apresentada uma proposta para o modelo matemático e a solução para um caso real.

Outros autores também utilizaram a abordagem de MILP para resolver problemas relacionados ao suprimento de petróleo: (MORO, 2003) estudou o planejamento da operação de uma refinaria que recebe diferentes tipos de óleo através de oleodutos utilizando a programação inteira linear. Neste caso trabalhou-se com um horizonte de, no máximo quatro dias, pois o algoritmo não conseguia gerar soluções eficientes para horizontes de planejamento maiores. O exemplo apresentado neste trabalho é o da refinaria de São José dos Campos (REVAP) que é abastecida pelo Terminal Petrolífero de São Sebastião (GEBAST). No entanto, não existe preocupação com o funcionamento do porto, trata-se unicamente do problema da refinaria.

Já (JIA, 2003) lida com o problema de distribuição de óleo cru de forma completa: desde o descarregamento dos navios até sua chegada nas refinarias, passando por tanques de armazenamento e distribuição. Mais uma vez, buscam-se maneiras de resolver o problema através de algoritmos de programação matemática, especialmente MILP. Aqui, os autores buscaram formular o problema utilizando técnicas de tempo contínuo, evitando a discretização do tempo que havia sido utilizada em outras formulações. A principal vantagem deste método é a redução do número de restrições e variáveis binárias que permite uma resolução mais eficiente através da MILP.

Quando modelos MILP são utilizados para resolver o problema do suprimento de petróleo, algumas simplificações devem ser feitas para tornar o sistema linear. Isto pode fazer com que o modelo não se distancie muito do sistema real em alguns casos. Para superar esta limitação, alguns trabalhos utilizam a programação mista inteira não-linear (MINLP). (NEIRO 2004) descreve um método geral para a modelagem dos elementos da cadeia de suprimento da indústria de petróleo (tanques, oleodutos, refinarias etc) que transforma o planejamento da operação em um problema de MINLP.

(NEIRO 2005) também busca propor um modelo baseado na programação não-linear. No entanto são utilizadas funções probabilísticas para a previsão de preços e demandas que afetam a tomada de decisão. O problema que foi resolvido como exemplo foi o da refinaria REVAP da Petrobrás. Novamente, só foi levado em conta o abastecimento da refinaria e não o funcionamento do terminal petrolífero.

Outros trabalhos também buscam inserir incertezas no modelo, pois estas aproximam o modelo da realidade onde existem flutuações nos preços do petróleo e incertezas quanto às datas de chegada dos navios (que podem se atrasar ou adiantar).

(LI 2006) Lida com o problema de distribuição de óleo cru: desde o descarregamento dos navios até sua chegada nas refinarias, passando por tanques de armazenamento e distribuição. No entanto, trata de inserir as incertezas relacionadas aos preços e as chegadas de navios na procura de um Schedule ótimo para a operação. Para isto são adicionadas variáveis probabilísticas ao modelo.

(ZHENYA, 2005) Lida com todo o problema do processamento de óleo cru: desde sua chegada nos navios até o processamento nas refinarias. São propostos modelos matemáticos para cada componente do sistema e utilizados algoritmos *branch and bound* para a resolução. Também são levadas em consideração incertezas com relação aos preços do petróleo.

Na área de planejamento automático, poucos trabalhos relacionados a este problema têm sido realizados. No entanto, na ICAPS'04 (*International Conference of Automated Planning and Scheduling*) um dos domínios propostos na quarta competição internacional de planejamento IPC-4 (Hoffman *et al.* 2004) estava relacionado ao problema de chaveamento das linhas de transmissão de petróleo internas ao terminal petrolífero (que levam o óleo do píer ao tanque, de um tanque para o outro etc.).

No entanto, o problema de chaveamento das linhas possui um foco muito operacional, enquanto este trabalho esta preocupado com um problema estratégico que é o planejamento das operações de suprimento de petróleo de forma a maximizar os lucros.

1.3. Objetivos

Este trabalho propõe uma abordagem de planejamento automático para a resolução do problema de planejamento de suprimento de petróleo em complexos contendo portos (compostos de píeres e tanques de armazenagem), refinarias e uma infra-estrutura de oleodutos. Esta abordagem possui algumas vantagens com relação

aos modelos matemáticos propostos hoje: modelagem do sistema em um nível mais alto e geração de modelos facilmente extensíveis e adaptáveis.

Será utilizada uma linguagem de modelagem chamada UML em uma Abordagem de Planejamento proposta por (VAQUERO *et al.* 2005) que utiliza elementos da UML e da OCL para modelagem de domínios de planejamento. Esta técnica permite uma melhor visualização do modelo proposto e a incorporação do conhecimento de diversos especialistas nas fases iniciais da modelagem (por exemplo, nos *Diagramas de Casos de Uso*), além de facilitar a análise e verificação do modelo desenvolvido. Também serão técnicas de Planejamento Automático em IA, através da linguagem PDDL (Planning Domain Definition Language).

Além disso, este trabalho busca avaliar a capacidade dos planejadores atuais de resolver problemas reais de grandes dimensões e alta complexidade. Este objetivo se torna muito importante quando consideramos o crescente desejo da comunidade de planejamento automático em aplicar as técnicas desenvolvidas até agora em problemas reais (VAQUERO *et al.* 2005).

1.4. Estrutura do Trabalho

A seguinte estrutura foi proposta para este trabalho:

- No capítulo dois, o problema que motivou o trabalho é apresentado em detalhes. Aqui, são incluídas informações qualitativas e quantitativas sobre o sistema.
- No capítulo três, as técnicas e abordagem propostas para resolução do problema são descritas. Uma breve descrição dos principais conceitos de planejamento e da linguagem de modelagem utilizada é apresentada.
- O modelo desenvolvido para a operação do terminal de São Sebastião é descrito no capítulo quatro. São apresentados os *Diagramas de Casos de Uso*, os *Diagramas de Classe* e os *Diagramas de Estados* para o sistema.
- O capítulo cinco apresenta os resultados dos experimentos realizados no modelo utilizando planejadores gerais disponíveis atualmente. Aqui também é realizada

uma análise do desempenho desses planejadores ao resolver o problema proposto.

- Finalmente, o capítulo seis apresenta as conclusões do trabalho e descreve os próximos passos a serem tomados no projeto.

2. DESCRIÇÃO DO PROBLEMA

Este capítulo introduz as características relevantes para a modelagem do problema proposto neste trabalho: o planejamento das operações do terminal petrolífero de São Sebastião.

A operação do sistema envolve o carregamento e descarregamento de navios petroleiros nos píeres, a distribuição dos óleos crus em tanques de armazenamento e o envio de óleo para suprir as necessidades das refinarias. O plano deve garantir que, em todos os momentos, o estoque nas refinarias será mantido acima de um determinado nível mínimo e, ao mesmo tempo, minimiza o custo da operação.

A descrição apresentada nesta seção é baseada em informações obtidas junto à Petrobrás e em (MÁS, 2001).

2.1. Rede de Distribuição

No Estado de São Paulo, a Petrobrás realiza o refino e o processamento de óleo cru nas refinarias de Paulínia (REPLAN), São José dos Campos (REVAP), Cubatão (RPBC) e Capuava (RECAP). Uma rede de oleodutos que se inicia no terminal de São Sebastião (GEBAST) é utilizada para abastecer estas refinarias com o petróleo necessário para suas operações. Também devemos mencionar a existência de duas estações intermediárias, uma em Cubatão (SEBAT) e outra em Guararema (SEGUA), e as estações de bombeamento em Rio Prado e Guaratuba. As estações SEGUA e SEBAT funcionam como pulmões do sistema, absorvendo a diferença de vazão entre os oleodutos. Todo óleo cru que é processado no Estado de São Paulo é recebido, através de navios petroleiros, pelo GEBAST, e distribuído para as refinarias através de dois oleodutos: OSVAT e OSBAT.



Figura 2.1 - Rede de distribuição da Petrobrás no Estado de São Paulo

Como pode ser visto a partir da Figura 2.1, a rede de distribuição é muito complexa e de difícil modelagem. Dessa forma, como mencionado anteriormente, este trabalho visa somente à modelagem das operações do terminal de São Sebastião. Assim, as estações de bombeamento, as subestações e refinarias não serão incluídas no modelo.

2.2. Especificações Gerais

O sistema descrito acima possui diversas restrições que devem ser incluídas no modelo para que o plano seja coerente com a realidade do terminal. Estas englobam a locação de óleos em classes, descrição do funcionamento de tanques e oleodutos e restrições relacionadas a navios e píeres. Estas características serão apresentadas a seguir.

2.2.1. Tipos de Óleo Cru

A Petrobrás trabalha com diversos tipos de petróleo: cerca 42 tipos de óleo cru distintos, que denominaremos oc-01 a oc-42, compõe praticamente toda a produção de derivados de petróleo. Como o terminal possui 44 tanques (vinte e um deles dedicados ao recebimento de petróleo, dezessete a derivados e seis tanques de serviço) existe uma restrição importante no problema quanto à limitação física de estoque. A solução mais simples, que seria a segregação de todos os tipos de óleo cru, não é implementável.

Para resolver este problema, dividem-se os diversos tipos de óleos crus em sete classes, que denominaremos cl-1 a cl-7. As classes englobam diversos tipos de óleo cru e, embora existam classes mais e menos nobres, qualquer tipo de óleo que pertença a uma classe pode ser misturado no tanque que a contém, sem prejuízo das propriedades do óleo como um todo. Deve-se mencionar que um mesmo tipo de óleo cru pode pertencer a mais de uma classe.

A tabela 2.1 mostra a associação entre alguns tipos de óleo cru e as diferentes classes. Um “X” em uma determinada posição indica que o óleo *oc-i* pertence à classe *cl-j*.

Tabela 2.1 - Associação entre classes e tipos de óleo cru

	cl-1	cl-2	cl-3	cl-4	cl-5	cl-6	cl-7
oc-01	X		X	X	X	X	
oc-02	X		X	X	X	X	
oc-03	X		X	X	X	X	
oc-04	X	X		X	X		X
oc-05	X		X	X	X	X	
oc-06			X	X			
oc-07			X	X			
oc-08			X	X			
oc-09			X	X			
oc-10	X		X	X	X	X	

2.2.2 Oleodutos

Os oleodutos levam óleos crus do terminal de São Sebastião até as refinarias que irão processar-los. Um oleoduto pode transportar mais de um tipo de óleo cru ao mesmo tempo e, neste caso, a capacidade máxima de transporte é dada pela vazão máxima do óleo cru mais viscoso.

Outra consideração importante sobre o transporte de petróleo em oleodutos está relacionada à mistura que ocorre internamente. Quando óleos diferentes são transportados em um único oleoduto, ocorre mistura na interface entre os óleos transportados. Isto gera perdas importantes de propriedades (e, portanto, valor econômico), dos mesmos nessa região.

Quando isto ocorre, a quantidade misturada é armazenada em tanques distintos para evitar a contaminação dos óleos puros, cujas propriedades precisam ser preservadas para evitar perdas de valor.

Tabela 2.2 - Descrição dos oleodutos do problema

	Trecho	Diâmetro (pol)	Comprimento (km)	Volume (1000 m ³)
OSVAT	GEBAST/ESPAR	42"	34,4	29,6
	ESPAR/SEGUA	38"	48,6	34,8
	SEGUA/REPLAN	30"	152,7	65,2
	SEGUA/REVAP	34"	35,1	21
OSBAT	GEBAST/ESGUAR	24"	70,5	18
	ESGUAR/RPBC	24"	50,5	15,8
	RPBC/SEBAT	24"	3,6	1,5
	SEBAT/RECAP	12"	30,0	2,3

2.2.3. Tanques de Armazenamento

Para prevenir o acúmulo de voláteis dentro dos tanques e, portanto, aumentar a segurança no processo de estocagem a Petrobrás utiliza o sistema de teto flutuante. Dessa forma, o teto do tanque sobe ou desce de acordo com o nível do óleo. Isto cria a

necessidade de um estoque mínimo nos tanques para evitar danos à estrutura da base do teto. Esta quantidade de óleo cru, necessária para manter o teto a uma determinada altura, é chamada de *lastro*. No sistema da Petrobrás adota-se um lastro de dois metros (cerca de 15% da capacidade total de cada tanque). Assim, cada tanque possui uma capacidade máxima e mínima de armazenamento que devem ser respeitada em toda a operação.

Tabela 2.3 - Tanques de armazenagem de óleo cru do GEBAST

Tanque	Est. Min (x 1000 m ³)	Est. Max (x 1000 m ³)
TQ3208	5,192	30,605
TQ3209	5,192	30,626
TQ3210	5,192	31,322
TQ3214	10,353	66,004
TQ3215	10,353	64,826
TQ3216	10,353	69,977
TQ3217	10,353	70,747
TQ3218	10,353	68,582
TQ3219	10,353	68,576
TQ3233	11,780	73,786
TQ3234	11,780	75,969
TQ3235	11,780	76,979
TQ3236	11,780	74,315
TQ3237	11,780	72,713
TQ3238	11,780	73,382
TQ3239	11,780	77,019
TQ3240	11,780	74,074
TQ3241	11,780	73,928
TQ3242	11,780	73,355
TQ3243	11,780	77,400
TQ3244	11,780	75,790

Outras restrições importantes da operação dos tanques estão relacionadas aos processos de carga e descarga. Um tanque pode se encontrar em três estados diferentes: inoperante, carregando ou descarregando. Um tanque nunca pode ser carregado e descarregado ao mesmo tempo. Além disso, como todos os tipos de óleo descarregados no terminal de São Sebastião contêm salmoura (mesmo após a separação em plataformas de petróleo), os tanques devem passar por um período de

decantação (durante o qual se mantém inoperante), após terem recebido óleo de um navio, antes de enviar esse óleo para uma refinaria. Isto é feito nos tanques do terminal de São Sebastião, pois não é interessantes transportar salmoura pelos oleodutos e enviar-la para as refinarias.

2.2.4. Recebimento de Óleo Cru no GEBAST

Os tipos de óleo cru chegam ao GEBAST em navios petroleiros (tanto os nacionais quanto os importados) e são descarregados nos quatro píeres de recebimento de petróleo, que serão denominados de P1 a P4. Os píeres possuem limitações relacionadas ao tamanho (calado) do navio que podem receber: o píer P1 é dito de grande calado, pois comporta petroleiros de até 300.000 m³, enquanto os píeres P2, P3 (ambos com capacidade de 120.000 m³) e P4 (com capacidade de 60.000 m³) podem descarregar apenas petroleiros menores.

Vale destacar que, no planejamento de óleo cru da Petrobras, os inventários de cada petroleiro já possuem alocação pré-definida, ou seja, os tipos de óleo cru que chegam em cada navio têm refinarias estabelecidas a priori. No modelo a ser desenvolvido, entretanto, pretende-se dar maior flexibilidade ao problema, pois se acredita que a pré-alocação é uma restrição simplificadora imposta pela programação da produção. Desta forma, os inventários de cada petroleiro poderão ser transportados para qualquer refinaria do sistema.

Os petroleiros são descarregados através de bombas de capacidade máxima entre 6.400 m³/h (oc-26) a 8.000 m³/h (oc-13), e os óleos crus armazenados em tanques no GEBAST.

Os inventários dos tanques não têm destino totalmente livres, ou seja, há tanques cujos estoques são destinados sempre ao oleoduto OSBAT (TQ3208 e TQ3210). Outros armazenam petróleo que, preferencialmente pelo oleoduto OSVAT (TQ3214/15/17/18/19/33) ou pelo OSVAT (TQ3234/35/37/38/39/40/41/42/44). Novamente, esta restrição operacional não será considerada no modelo por ser simplificadora.

Um aspecto crítico do problema refere-se à operação de descarga dos navios. De fato, para cada petroleiro é determinado o número de dias em que este fica disponível para descarregamento sem custos de sobreestadia. Se for necessário que um dado petroleiro fique por mais tempo no porto, a Petrobrás deverá pagar pela sobreestadia. O valor a ser pago varia entre US\$ 20.000 (para petroleiros de 60.000 m³) a US\$ 40.000 (para navios de 300.000 m³) por dia.

2.2.5. Armazenamento de Óleo Cru nas Refinarias

No modelo desenvolvido, as refinarias serão modeladas como consumidoras de óleo cru do sistema. Será admitido que, no planejamento de curto prazo do suprimento de óleo cru, as refinarias já tenham preestabelecido o consumo médio no período, e caberá ao modelo apresentado, dentro do horizonte de planejamento especificado, definir a operação de transporte das classes de óleo que cada refinaria aceita para reposição dos respectivos estoques.

3. PLANEJAMENTO AUTOMÁTICO E UML

3.1. Planejamento Automático

Nesta seção será apresentada uma breve descrição dos principais conceitos, e linguagens associadas ao processo de design de sistemas, principalmente dos domínios de planejamento. Inicialmente, serão apresentados os conceitos da área de Planejamento Automático onde este trabalho está inserido. Em seguida, é apresentada a linguagem utilizada para modelagem de domínios de Planejamento Automático PDDL – *Planning Domain Definition Language* (MCDERMOTT, 1998).

3.1.1. Conceito

Planejamento automático é um ramo da inteligência artificial que se preocupa com a realização de estratégias ou seqüências de ações, que tipicamente são executadas por agentes inteligentes, robôs autônomos e veículos não tripulados. Diferentemente dos problemas clássicos de controle e classificação, as soluções são complexas, desconhecidas e devem ser descobertas e otimizadas em um espaço multidimensional.

A pesquisa na área de Planejamento Automático esta focada, principalmente, no desenvolvimento de sistemas computacionais de planejamento, chamados de planejadores, que são capazes de procurar por uma seqüência de ações a serem aplicadas em um domínio para atingir um conjunto de metas ou objetivos a partir de um determinado estado inicial.

Um planejador típico recebe três entradas: uma descrição do estado inicial do sistema, uma descrição do objetivo desejado e um conjunto de possíveis ações, codificados em uma linguagem formal como a PDDL. O planejador produz uma

seqüência de ações que levam o sistema de um estado inicial para o estado final desejado.

Em ambientes conhecidos com modelos disponíveis, o planejamento pode ser realizado *offline*. Soluções podem ser encontradas e avaliadas antes da execução. Em sistemas dinâmicos desconhecidos a estratégia deve ser revisada *online*. Modelos e políticas precisam ser adaptadas. As soluções normalmente se utilizam de processos iterativos de *tentativa e erro*, comumente vistos como inteligência artificial. Estes incluem, programação dinâmica, otimização combinatória, etc. A Figura 3.1 mostra o modelo conceitual de Planejamento Automático.

Nessa figura, podemos observar os três principais elementos que estão presentes em um sistema real de planejamento: o planejador, que recebe como entrada a descrição do sistema, uma situação inicial e os objetivos desejados; o controlador recebe o estado atual do sistema real através de observações e atua de acordo com o plano encontrado pelo planejador; e o sistema a ser controlado que evolui de acordo com as ações e eventos que recebe do controlador e do meio externo.

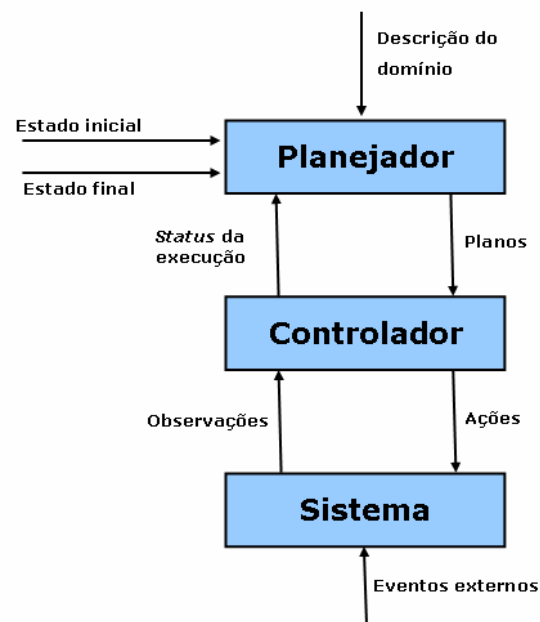


Figura 3.1 – Modelo conceitual de Planejamento Automático.

Fonte (GHALLAB; NAU; TRAVESSO, 2004).

O Planejamento Automático aborda os diversos aspectos presentes no modelo conceitual apresentado, mas nem todos os conceitos são facilmente tratados. Durante muito tempo, foram aplicadas diversas restrições aos problemas e domínios tratados pelo Planejamento Automático, para que fosse possível desenvolver planejadores capazes de apresentar soluções satisfatórias. Este conjunto de restrições caracteriza o chamado Planejamento Clássico. Essas simplificações e suposições são descritas a seguir (GHALLAB; NAU; TRAVESSO, 2004):

- A0. Σ **Finito**: O sistema estado transição Σ possui um conjunto finito de estados.
- A1. Σ **Totalmente Observável**: O sistema Σ é totalmente observável, isto é, tem-se total conhecimento do estado de Σ .
- A2. Σ **Determinístico**: O sistema Σ é determinístico.
- A3. Σ **Estático**: O sistema Σ é estático, isto é, o conjunto de *Eventos* E é vazio.
- A4. **Objetivos Restritos**: Os planejadores lidam apenas com *objetivos restritos* que são especificados explicitamente no estado objetivo.
- A5. **Planos Seqüenciais**: Um plano-solução de um problema de planejamento é uma seqüência finita de ações linearmente ordenada.
- A6. **Tempo Implícito**: Ações e eventos não possuem duração. Elas são transições de estado instantâneas.
- A7. **Planejamento Offline**: O planejador não percebe mudanças do sistema Σ enquanto está planejando. Ele planeja apenas com as condições iniciais e os objetivos, independentemente da dinâmica que ocorre em Σ . O planejador não recebe status de execução.

Este trabalho lida apenas com o planejador, ou seja, o planejamento offline. Mais especificamente, esta abordagem está focada na descrição do domínio, definição do problema (estado inicial e estado final) e no plano gerado pelo planejador. No entanto, os axiomas do planejamento clássico não são suficientes para tratar o problema descrito neste trabalho (planejamento do suprimento de petróleo) de uma forma satisfatória. Por isso, algumas das restrições descritas acima devem ser relaxadas (GHALLAB; NAU; TRAVESSO, 2004):

- **Relaxando o Axioma A0 (Σ Finito).** Como o modelo utiliza variáveis numéricas (por exemplo, volume dos tanques), um conjunto de estados infinitos se torna necessário para descrever as ações do domínio.
- **Relaxando o Axioma A4 (Objetivos Restritos).** No problema tratado neste trabalho, a única informação que temos sobre o estado final diz respeito ao estado dos navios (que devem ter seu inventário vazio e estarem desatracados). Por tanto, o estado final não está completamente definido. Além disso, o este problema exige a utilização de otimização de determinadas variáveis.
- **Relaxando o Axioma A5 (Planos Seqüenciais).** No problema estudado muitas ações podem (e devem) ser realizadas em paralelo – por exemplo, um tanque pode ser abastecido enquanto outro é descarregado.
- **Relaxando o Axioma A6 (Tempo Implícito).** No problema do suprimento de petróleo, muitas ações possuem duração. Assim, o modelo estado-transição não é suficiente para tratar este problema.

As considerações descritas acima permitem que o problema do suprimento de petróleo em terminais petrolíferos possa ser modelado de uma forma satisfatória. Além disso, essas considerações já foram, em maior ou menor grau, incorporadas às técnicas atuais de planejamento.

3.1.2. PDDL – *Planning Domain Definition Language*

Esta linguagem tem como objetivo representar os domínios do mundo real através de uma estrutura que possa ser entendida e interpretada pelo planejador. Hoje, a grande maioria dos planejadores é capaz de lidar com domínios e problemas descritos em PDDL (VAQUERO, 2007). A representação do modelo deve ser aproximar consideravelmente do sistema real, contendo a descrições das ações possíveis no domínio, suas pré e pós condições e as informações necessárias sobre os estados iniciais e final (objetivo).

A representação de um modelo de domínio de planejamento é separada em duas partes. A primeira possui a descrição do domínio onde se encontram, principalmente, as ações possíveis de serem realizadas no domínio, assim como a declaração dos tipos de objetos existentes. A segunda parte possui a definição do problema que deverá ser resolvido pelo planejador que constitui de uma descrição do estado inicial e do estado final (objetivo a ser atingido). O lado positivo dessa separação é que um mesmo modelo de domínio pode ser utilizado para resolver diversos problemas.

A seguir o formalismo da PDDL para definições dos domínios e dos problemas será apresentado.

3.1.2.1. Visão Geral da Sintaxe

Neste trabalho foi utilizada a PDDL 2.1 para modelagem do terminal de São Sebastião. A sintaxe dessa linguagem segue o seguinte formato (VAQUERO, 2007):

- Cada regra está descrita no formato <elemento sintático> ::= expansão.
- Os símbolos menor e maior (< e >) delimitam os nomes dos elementos sintáticos.
- Colchetes ([e]) representam informações opcionais.
- O asterisco (*) representa zero ou mais elementos, enquanto o sinal positivo (+) representa um ou mais elementos.
- Parênteses são apenas delimitadores dos elementos sintáticos, não possuindo nenhuma interpretação semântica.
- Informações extras e regras de expansão podem aparecer sobrescritas por um marcador.

Utilizando as regras descritas acima podemos apresentar as sintaxes para definição de domínios e problemas.

3.1.2.2. Definição de Domínios

O formalismo e a estrutura para definir os domínios de planejamento seguem a BNF apresentada na figura 3.2. As palavras chaves devem aparecer na ordem em que são apresentadas na figura.

A seguir são descritos alguns dos elementos apresentados na estrutura da figura 3.2:

name: Esta categoria consiste de uma palavra iniciada por uma letra, podendo conter caracteres alfanuméricos, hífen e traços subscritos. Os nomes definidos nesta categoria devem ser únicos.

requirements: Indicam características específicas que estarão presentes no modelo. Alguns planejadores suportam apenas algumas características declaradas nesta categoria. Neste caso, o planejador sinalizaria que não possui a capacidade de lidar com o modelo em questão.

types: É uma lista usada para declarar tipos de entidades/classes que estão presentes no modelo do domínio.

constants: as informações declaradas neste campo são tomadas como constantes em todo o modelo do domínio, bem como em todos os problemas do domínio em questão.

predicates: este campo consiste de uma lista de declarações de predicados.

functions: este campo consiste de uma lista de declarações de predicados que serão associados a valores numéricos.

```

<domain> ::= (define (domain <name>)
               [<required - def>]
               [<types - def>]:typing
               [<constants - def>]
               [<predicates - def>]
               [<functions - def>]:fluents
               <structure - def>*)

<require - def>      ::= (:requirements <require - key> +)
<require - key>      ::= Podendo ser :strips :adl :typing :fluents e outros
<types - def>        ::= (:types <typed list (name)> )
<constants - def>    ::= (:constants < typed list (names) >)
<predicates - def>   ::= (:predicates < atomic formula > +)
<atomic formula skeleton> ::= (<predicate> <typed list
                               (variable)>)
<predicate>          ::= <name>
<variable >          ::= ?<name>
<atomic function skeleton> ::= (<function-symbol> <typed list
                               (variable)>)
<function-symbol>    ::= <name>
<functions-def>      ::= :fluents (:functions <function typed list
                               (atomic function skeleton)>)
<structure-def>      ::= <action-def>

```

Figura 3.2 – Sintaxe de definição de domínios em PDDL (FOX; LONG, 2003).

3.1.2.3. Definição de Ações

A representação de ações é uma das principais partes dentro da definição do domínio de planejamento. A sintaxe utilizada na declaração das ações demonstrada na Figura 3.3 depende do conjunto de tipos declarados na cláusula *:requirements*.

A condição para que uma ação seja realizada em um estado depende da instanciação dos predicados da pré-condição e seus correspondentes valores verdade. Uma vez que os predicados da pré-condição sejam instanciados e seus valores habilitem a execução da ação, os predicados contidos em *<atomic - eff>* são totalmente instanciados e adicionados ao estado seguinte e removidos do estado anterior.

```

<action - def> ::= (:action <name>
                    :parameters (<typed list (variable)>))
<action -def body> ::= [:precondition <GD>]
                    [:effect <eff - formula>]
<GD> ::= <literal term>
<GD> ::= not <GD>
<GD> ::= (and <GD> <GD>+)
<GD> ::= (or <GD> <GD>+)
<GD> ::= (imply <GD> <GD>)
<GD> ::= (exist (<typed list (variable)>+) <GD>)
<atomic - eff> ::= <literal term>
<atomic - eff> ::= (and <literal term> <literal term>+)
<eff - formula*> ::= <atomic -eff>
<eff - formula*> ::= (when <GD> <atomic -eff>)
<eff - formula*> ::= (forall (<typed list (variable)>+) <atomic -
                        eff>))
<eff - formula*> ::= (forall (<typed list (variable)>+)
                        (when <GD> <atomic -eff>))
<eff - formula*> ::= <eff - formula*>
<eff - formula*> ::= (and <eff - formula*> <eff - formula*>+)

```

Figura 3.3 – Sintaxe de definição de ações em PDDL (FOX; LONG, 2003).

3.1.2.4. Definição do Problema de Planejamento

O problema de planejamento é definido basicamente por dois estados, o estado inicial do problema e o estado objetivo. O problema submetido ao agente planejador, também conhecido como arquivo de fatos, é definido em relação a um determinado domínio, descrevendo os fatos verdadeiros no estado inicial, e os fatos que constituem o estado objetivo. A sintaxe para definição de problemas é definida, de forma simplificada, conforme a Figura 3.4.

```

<problem>      ::= (define (problem <name>)
                    (: domain <name>)
                    [<require-def>]
                    [<object declaration>]
                    <init>
                    <goal>
                    [<metric-spec>]
<object declaration> ::= (:objects <typed list (name)>)
<init>            ::= (:init <init-el*>)
<init-el>         ::= <literal (name)>
<init-el>         ::= 'fluents (= <f-head> <number>)
<goal>           ::= (:goal <GD>)
<metric-spec>    ::= (:metric <optimization> <função a ser otimizada>)
<optimization>   ::= minimize
<optimization>   ::= maximize

```

Figura 3.4 – Sintaxe de definição de problemas em PDDL (FOX; LONG, 2003).

O estado inicial é uma lista de fórmulas atômicas consideradas verdadeiras no estado inicial e são definidas na seção *:init*. Deve-se mencionar que, qualquer predicado não declarado no estado inicial é presumido como falso. O estado objetivo é declarado na seção *:goal*. A solução do problema, como mencionado anteriormente, corresponde a aplicação de uma sequência de ações que levem o estado inicial ao estado objetivo.

3.2. Modelagem com UML

3.2.1. UML – *Unified Modeling Language*

A UML – *Unified Modeling Language* é uma das linguagens mais utilizadas para modelar uma grande variedade de aplicações e foi definida pela primeira vez na *OMG Unified Modeling Language Specification* entre 1996 e 1997 (OMG 2001). Além disso, a UML tem flexibilidade para atender diferentes tipos de modelo de uma forma orientada a objetos. Hoje, a UML é vista como o padrão para análise e design orientados a objeto (VAQUERO, 2007).

A UML é baseada em diagramas. Alguns diagramas desta linguagem são: o *Diagrama de Casos de Uso*, o *Diagrama de Classes*, o *Diagrama de Estados*, o *Diagrama de Objetos*, entre outros. A UML também possui uma linguagem predefinida

de representação de restrições chamada OCL – *Object Constraint Language* (OMG, 2003). A OCL é a linguagem formal utilizada para definir restrições complementares aos diagramas.

3.2.1.1. Diagramas de Caso de Uso

O *Diagrama de Casos de Uso* é geralmente o primeiro diagrama desenvolvido. Com este diagrama o projetista pode: decidir e descrever os requisitos funcionais do sistema; fornecer uma descrição clara e consistente do que o sistema deve fazer; e refinar os requisitos. Neste diagrama é possível descrever o sistema de uma forma abstrata fazendo com que haja uma visão geral e simplificada do domínio.

Este diagrama mostra o relacionamento entre os *Casos de Uso* e os *Atores* (também chamados de Agentes) do sistema. Um *Caso de Uso* é basicamente uma descrição de um caso ou processo do sistema. Vale ressaltar que o *Diagrama de Casos de Uso* é muitas vezes utilizado tanto para a especificação do sistema quanto para a modelagem.

3.2.1.2. Diagramas de Classes

O *Diagrama de Classes* representa, principalmente, a estrutura estática do modelo do domínio. Considerado um dos diagramas mais importantes e conhecidos, ele representa as classes (entidades e tipos) existentes, suas propriedades, as ações que cada uma delas pode realizar e as associações entre elas, assim como restrições nestas associações (OMG, 2001).

Classes e *Objetos* são os conceitos fundamentais na modelagem orientada a objeto. As *Classes* representam um conjunto de *Objetos* que possuem estrutura, comportamento e relacionamentos similares. Qualquer relacionamento entre classes é chamado de associação.

As associações criam relacionamentos semânticos entre as classes. Existem basicamente três tipos de associações: Associação Simples, Agregação e Composição. As associações simples apenas conectam as classes, identificando algum significado

apropriado pelo nome da associação e sua direção. Agregação e Composição são casos especiais de associações que já trazem algum significado semântico. A associação Agregação de o significado “possui”, enquanto a associação Composição tem o significado “é feito de”. Assim, pode-se notar que a Composição representa uma dependência mais forte que a Agregação.

Este diagrama também permite a definição de grupos de elementos com comportamentos similares no modelo. Este conceito é chamado de *stereotype* e é muito útil, por exemplo, para distinguir elementos passivos e ativos do modelo. Já existem alguns *stereotypes* pré-definidos: *actor*, que classifica atores do modelo; *entity*, que classifica os elementos passivos do modelo; *utility*, que define um comportamento global no modelo (os métodos e atributos desta classe se tornam variáveis e procedimentos globais).

3.2.1.3. Diagramas de Estado

A UML possui alguns diagramas que definem características dinâmicas das entidades existentes no modelo do sistema, como o *Diagrama de Estados*. Este diagrama pode ser utilizado para descrever o comportamento de instâncias das classes do domínio através da representação dos possíveis estados, das pré e pós-condições das ações que as afetam (D'SOUZA; WILLS, 1999). Os estados representam situações nas quais um objeto pode se encontrar e as ações representam a transição de um estado para outro (OMG, 2001).

Cada *Diagrama de Estados* descreve as possíveis seqüências de estados nos quais um único objeto passa durante a sua vida no sistema em resposta a eventos e ações. Esta vida é baseada nos resultados das ações e das reações ocorridas. Basicamente, cada diagrama de estados representa o comportamento de uma determinada classe (OMG, 2001).

3.2.1.4. Diagramas de Objetos ou *Snapshots*

O *Diagrama de Objetos* representa uma situação, um estado, ou momento do domínio. Este diagrama é também chamado de *Snapshot* e é, de fato, uma instância do diagrama de classes. O *Snapshot* permite ao usuário instanciar classes, dar valores aos atributos de cada instância das classes e associar os objetos de acordo com a situação que o projetista deseja construir. Basicamente, o *Snapshot* é uma foto de um estado do sistema em um momento específico. Uma seqüência de diagramas de objetos nos dá a evolução do sistema, formando um “filme” da evolução do sistema. É importante lembrar que os valores e associações realizadas no *Diagrama de Estados* devem respeitar as restrições definidas no *Diagrama de Classes*.

3.2.1.5. OCL – *Object Constraint Language*

Um diagrama UML geralmente não é refinado o suficiente para modelar todos os aspectos relevantes de uma modelagem (VAQUERO, 2007). Existe a necessidade de expressar restrições adicionais sobre os objetos no modelo em desenvolvimento. Tais restrições são geralmente descritas em linguagem natural, que tendem a causar ambigüidade (VAQUERO, 2007). Para lidar com estas questões, foi desenvolvida a linguagem formal OCL – *Object Constraint Language* (OMG, 2003) – para especificar restrições.

A OCL é muito similar à lógica de predicados de primeira ordem. As restrições são geralmente expressões booleanas, que podem ser utilizadas com conectores lógicos, que devem ser verdade durante todo o ciclo de vida do objeto no sistema. As expressões de restrições podem complementar a descrição das características e dos comportamentos dos objetos no sistema. É possível representar expressões sobre classes e associações, estados, ações, diagramas, entre outros.

Esta linguagem se mostra muito útil na fase de modelagem na área de Planejamento Automático já que esta é uma linguagem formal de especificação de restrições. Restrições como invariantes são muito úteis durante o processo de planejamento para que os planejadores evitem buscas exaustivas em caminhos que não chegam a soluções plausíveis.

As restrições podem, de fato, enriquecer o modelo do domínio fazendo com que as técnicas de planejamento possam usufruir desta informação. É importante citar que muitas das capacidades de expressão das recentes evoluções da PDDL já estavam presentes na OCL (VAQUERO, 2007).

3.2.2. Utilizando a UML em uma Abordagem de Planejamento

Como a UML é uma linguagem de modelagem de propósito geral, algumas de suas características estão intimamente ligadas a domínios de planejamento. Por esta razão, a UML em uma Abordagem de Planejamento (abreviada como UML.P) foi apresentada em (VAQUERO *et al.*, 2005), onde os conceitos de planejamento são especificados em UML.

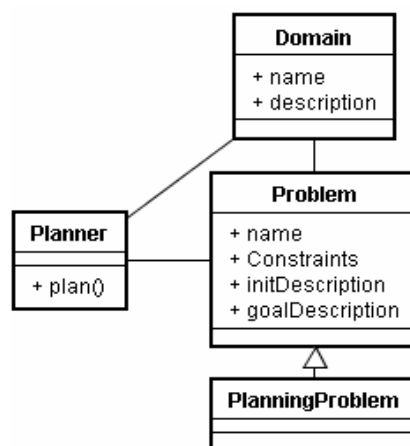


Figura 3.5: Esquema geral do ambiente de planejamento

Esta abordagem considera a relação entre planejadores, domínios e problemas de planejamento. Em um contexto de planejamento, o processo de modelagem segue os seguintes princípios: domínios têm sua própria descrição e especificação (incluindo estrutura estática, comportamento dinâmico, restrições etc.); problemas estão associados a domínios e eles têm suas próprias restrições (métricas, preferências etc.), descrição da condição inicial e condição final; planejadores atuam sobre a descrição do

domínio e dos problemas associados. Estes princípios tentam distinguir a descrição de um domínio da descrição de um problema e definir o papel do planejador. A figura 3.5 mostra um esquema onde estes conceitos são resumidos.

Ao utilizar a UML, alguns diagramas são mais apropriados para o contexto de planejamento, como o diagrama de Casos de Uso, Classe, Estados, Atividades e Objetos. As seguintes descrições desses diagramas mostram como *designers* podem especificar e entender melhor seus domínios de planejamento.

O diagrama de Casos de Uso modela o domínio no nível mais alto de abstração e facilita a unificação dos diferentes pontos de vista.

Na UML, especificações de *Casos de Uso* são muitas vezes descritas em linguagem natural no nível de abstração desejado, mas a UML.P faz isto de forma diferente. Como a especificação em linguagem natural pode criar ambigüidades e redundâncias, a utilização de uma especificação de estrutura dos Casos de Uso contribui para minimizar o problema. Esta estruturação foi apresentada no trabalho de Silva e Santos (2004).

Outro diagrama importante para planejamento é o diagrama de *Classes*. Os diagramas de *Classes* representam a estrutura e conceitos estáticos do domínio, mostrando as entidades existentes, sua relação, características, métodos (ações) e restrições.

Deve-se frisar que o diagrama de classes representa um grande conjunto de problemas e não um problema em particular.

Para especificar o comportamento dinâmico das ações do domínio é necessário utilizar o diagrama de *Estados*. Neste diagrama é possível definir as pré e pos-condições de cada método (ação) definido no diagrama de *Classes*.

Qualquer classe no *Diagrama de Classes* pode ter seu próprio diagrama de Estados, especialmente aquelas que executam ações. Um diagrama não especifica todas as mudanças causadas por uma ação, mas somente aquelas relativas à classe representada no diagrama de Estados.

As restrições no diagrama de Classe e todas as pré e pos-condições no diagrama de Estados são especificados utilizando a linguagem OCL.

Um problema em um domínio de planejamento é caracterizado por uma situação onde somente dois pontos são conhecidos: o estado inicial e o estado final. O diagrama utilizado para fazer esta representação é o diagrama de Objetos ou *Snapshots*.

Uma *snapshot* é uma fotografia em um determinado instante e uma instanciação da estrutura do domínio. Tal instanciação representa características como: quantos objetos estão no problema, quais suas classes, valores de seus atributos e as inter-relações entre os objetos. Um problema de planejamento possui dois diagramas de Objeto: um que descreve o estado inicial e outro o estado final.

4. MODELAGEM DO PORTO

Como foi dito anteriormente, o projeto visa buscar um modelo e algoritmos que permitam o planejamento das operações do terminal petrolífero de São Sebastião, tendo em vista a otimização de seu lucro, de forma eficiente e robusta.

Para a sua modelagem, este trabalho utiliza uma aplicação especial para UML – *Unified Modeling Language* (OMG, 2003) – em uma abordagem de planejamento, chamada de UML.P (VAQUERO, 2005). A utilização desta abordagem pode contribuir para a descoberta de melhores soluções, além de permitir a identificação de características e problemas do sistema que dificilmente são identificadas nos métodos utilizados até agora.

Para o desenvolvimento do modelo apresentado nesta seção, foi utilizada a ferramenta itSIMPLE (VAQUERO, 2005). Esta ferramenta consiste de um ambiente integrado de modelagem e análise de domínios de planejamento, que leva em consideração o ciclo de vida de projeto. O itSIMPLE provê um ambiente gráfico para modelagem com total suporte à UML.P e realiza a tradução desses modelos para PDDL de maneira automática.

Atualmente, o modelo não considera as restrições de tempo envolvidas no sistema. Esta restrição não permite a modelagem da operação das refinarias (já que elas se comportam com consumidoras contínuas de óleo cru) e de custos que envolvem tempo (como o custo de sobreestadia).

No entanto, mesmo que os custos relacionados ao tempo tenham um impacto importante no resultado econômico da operação, a maior parte da receita da operação esta relacionada a alocação correta de óleos em tanques de armazenamento. Como classes diferentes possuem valores de mercado distintos, a escolha dos tanques corretos onde alocar os óleos para maximizar seu valor e minimizar os custo não relacionados ao tempo (como o custo de interface) é uma parte essencial do planejamento do suprimento de óleo.

A seguir, será apresentado o modelo das operações do terminal de São Sebastião desenvolvido em UML.P.

4.1. Visão Geral

Diante do problema descrito no capítulo dois, a proposta é obter uma metodologia que forneça resultados de programação de descarregamento de navios, da utilização dos píeres e tanques dos terminais, utilização dos oleodutos e alocação, nas refinarias, das cargas de óleo cru que provêm dos navios petroleiros, respeitando as restrições impostas pelo sistema. A Figura 4.1, ilustra as entradas do modelo e as saídas esperadas.

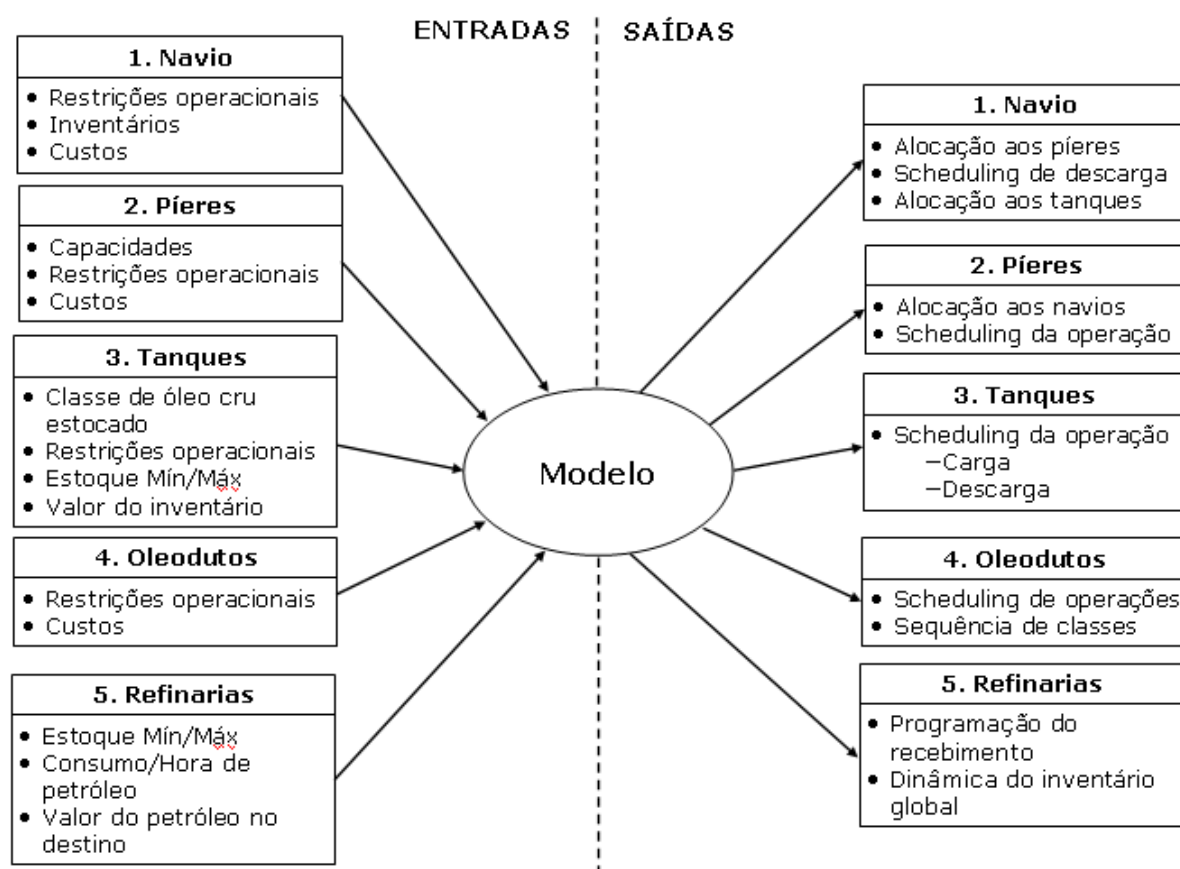


Figura 4.1 Entradas e saídas do modelo proposto

Considerando o problema a partir do porto até as refinarias, temos as seguintes decisões que devem ser determinadas pelo modelo:

- Alocação de navios aos píeres
- Descarregamento dos navios atracados aos potenciais tanques recebedores
- Descarregamento dos tanques aos oleodutos conectados
- Recebimento de óleo cru pelas refinarias

A saída do modelo, portanto, é a seqüência de ações (ou decisões) que levam sistema de um estado inicial a um estado final desejado, respeitando todas as restrições inerentes ao sistema. Dessa forma este problema pode ser definido como um problema de planejamento e programação, permitindo a utilização de técnicas de planejamento automático para sua resolução.

Como mencionado anteriormente, utilizaremos uma abordagem conhecida como UML.P que permite que domínios de planejamento sejam modelados utilizando UML e OCL que, posteriormente, são convertidos para a linguagem entendida pelos planejadores atuais: PDDL (*Planning Domain Definition Language*). A principal vantagem dessa abordagem esta na possibilidade de fazer uma modelagem *top down* permitindo que diversas visões (especialmente de indivíduos que não são *experts* em planejamento) sejam incluídas no modelo. Além disso, a utilização de diagramas da UML permite uma visualização muito mais clara do modelo, em todas as etapas da modelagem, do que um modelo em PDDL, facilitando a análise do domínio e a detecção de erros.

4.2. Hipóteses do Modelo de Porto

O modelo apresentado nesta seção se baseia em uma série de hipóteses, determinadas pelas restrições do sistema de distribuição de óleo cru, que foram baseadas no trabalho de (MÁS, 2001) e validadas junto à Diretoria de Operações do terminal de São Sebastião. Além disso, devido a limitações encontradas nos planejadores gerais disponíveis atualmente, foram adicionadas algumas hipóteses que permitissem que estes fossem capazes de lidar com o problema de forma satisfatória.

- **Cada item de inventário de óleo cru proveniente dos navios deve ser descarregado somente nos tanques cuja classe incluir este tipo de óleo.** Este procedimento é seguido na prática para evitar a contaminação de tanques que armazenam classes mais nobres de óleo cru.
- **Cada navio só poderá atracar nos píeres que possuem capacidade de fazê-lo.** Essa é uma restrição puramente operacional, pois navios só podem atracar nos píeres que reúnam condições de execução da operação.
- **O planejador pode decidir qual a ordem de chegada dos navios de forma a otimizar as operações.** Esta é uma hipótese simplificadora, pois, na prática, as datas de chegada dos navios no porto já estão previamente definidas. No entanto, os planejadores atuais se mostraram incapazes de lidar com eventos externos (chegada de um navio no porto).
- **O navio só começa a ser descarregado algumas horas (tempo de atracação) após o momento da alocação do navio ao píer.** Esta hipótese permite a inclusão do tempo de atracação de um navio.
- **O navio deve ser descarregado por completo antes de deixar o porto.** Esta hipótese garante que os navios só deixarão o porto após terem descarregado todo o óleo de seu inventário destinado ao terminal em questão.
- **Outro navio só pode ser enviado ao píer algumas horas (tempo de saída) após o final do descarregamento do petroleiro que se encontrava no píer anteriormente.** Assim como o tempo de atracação, o tempo de saída pode ser incluído no modelo, para tornar-lo mais condizente com a realidade.
- **Para cada navio, admite-se apenas uma conexão a um tanque em cada evento.** Esta premissa favorece a formulação das restrições de temporização dos descarregamentos dos navios e pode ser considerada válida, pois permite a operação de carregamento em regime de vazão constante.
- **Para cada tanque, deve-se armazenar uma e só uma classe de óleo cru.** Na prática, esta hipótese é legítima para um cronograma de curto prazo a ser determinado.
- **Cada tanque pode estar ocioso, ser carregado por um navio, ou ser descarregado para um oleoduto em cada instante.** Como já foi afirmado

anteriormente, as tarefas de carregamento e descarregamento não podem ser realizadas simultaneamente, pois impossibilita o controle do inventário do tanque, além de prejudicar a operação de decantação do mesmo.

- **O volume do tanque está restrito inferiormente por um volume mínimo e superiormente por um volume máximo.** O estoque mínimo é definido pela altura mínima de óleo cru (ou lastro) que deve ser mantida para evitar danos ao teto flutuante do tanque, enquanto o máximo é determinado pela capacidade de armazenamento do tanque.
- **Após o instante de carregamento, deve-se aguardar algumas horas (tempo de decantação) para o descarregamento do tanque.** Esta restrição deve ser imposta sempre, pois o óleo proveniente dos navios normalmente contém salmoura, aumentando a densidade, prejudicando um transporte via oleoduto e o processamento nas refinarias.
- **As capacidades de vazão, numa mesma classe, são constantes, independente dos tipos de óleo cru contidos nela.** Esta hipótese pode ser considerada razoável para valores adequados de densidades adotados para as classes de óleo cru em questão, já que a diferença percentual de densidade entre o tipo de óleo cru mais denso (oc-26) e o menos denso (oc-13) dentre os considerados neste trabalho é de apenas 31,1%. Esta aproximação é adotada para evitar perda de eficiência computacional necessária para a viabilidade do modelo.
- **A vazão no oleoduto corresponde à vazão da classe sendo bombeada.** Assim como no caso anterior, essa hipótese não reflete exatamente o que ocorre na realidade, pois a capacidade de vazão é determinada pela classe mais densa dentro do oleoduto. No entanto, como a vazão de uma classe é uma média da vazão dos óleos nela contida (um mesmo óleo pode pertencer a diversas classes) e a diferença de densidades entre os óleos considerados é muito pequena (31,1% como mencionado anteriormente), esta hipótese pode ser considerada razoável. Novamente, esta aproximação é adotada para evitar perda de eficiência computacional necessária para a viabilidade do modelo.

- **Cada oleoduto pode receber óleo cru de, no máximo, um tanque em cada instante.** Esta restrição respeita condições operacionais de sistemas reais, além de facilitar o controle das características dos tipos de petróleo transportados pelo oleoduto.
- **Cada refinaria é conectada ao porto através de um, e somente um, oleoduto.** Esta restrição condiz com a realidade dos cenários analisados neste trabalho e é estabelecida para facilitar o balanço material das refinarias e o consequente controle de seus inventários.

Como pode ser visto nas hipóteses acima, o custo de estadia e sobreestadia não foi incluído no modelo. Estes custos são diretamente proporcionais ao tempo de estadia e de sobreestadia, respectivamente. Apesar de serem itens de custo relevantes, verificou-se que, nos planejadores atuais, o tratamento do tempo ainda é incipiente, não permitindo a otimização de variáveis temporais. Por isso, estes dois parâmetros não foram considerados, diretamente, no modelo.

Apresentadas as hipóteses do modelo proposto, o *Modelo de Porto* é apresentado nas seções seguintes.

4.3. Casos de uso

O *Diagrama de Casos de Uso* modela o domínio no nível mais alto de abstração onde os requisitos do modelo são definidos pela primeira vez. Este diagrama facilita a unificação dos diferentes pontos de vistas envolvidos.

O *Diagrama de Casos de Uso* para as atividades de distribuição de óleo cru no terminal de São Sebastião pode ser visto na Figura 4.3. Cada caso de uso possui elementos que recebem uma descrição completa de suas pré e pos-condições, restrições, invariantes, fluxo de eventos e outras informações relevantes para a modelagem. A Figura 4.2 mostra o caso de uso *Atracar no píer*, onde os elementos mencionados acima podem ser vistos claramente.

Caso de Uso: Atracar no píer

Atores: Navio, Porto

Descrição: Os navios que chegam ao terminal petrolífero são alocados aos píeres para que seus tanques sejam descarregados.

Restrições:

Pré-condições

1. O navio deve estar desatracado.
2. Um navio só pode atracar em píeres que possuam capacidade de fazê-lo.
3. O píer deve estar livre para receber navios.
4. O instante de chegada do navio deve ser anterior ao das outras embarcações de desejam atracar no mesmo píer.

Pós-condições

1. O navio está atracado.
2. O píer está ocupado.
3. O inventário de óleo cru do navio pode ser descarregado.
4. O custo de estadia do navio passa a ser contabilizado.

Invariantes

O navio só pode atracar em um único píer.

Restrições Temporais

Após alocado a um píer, o navio demora um tempo determinado para finalizar esta operação, dependendo do porte do navio. Seu inventário só pode ser descarregado nos tanques do porto a partir desse instante.

Outros Detalhes

Informações Adicionais

1. A taxa de estadia que será cobrada de um navio depende unicamente do píer onde ele atracou.
2. O navio possui uma janela de tempo durante a qual pode permanecer atracado pagando as tarifas correspondentes ao seu porte. Caso o petroleiro ultrapasse o período contratado, será cobrada uma taxa de sobreestadia, que depende do seu porte e é diretamente proporcional ao tempo de sobreestadia.

Figura 4.2 - Caso de Uso *Atracar no píer*

Como grande parte da informação sobre um sistema é fornecida através da linguagem natural, os casos de uso constituem uma primeira tentativa de estruturar esses dados. Além disso, os casos de uso permitem uma primeira visualização do

funcionamento do modelo. Portanto, é possível realizar alguma análise de requisitos já nesta fase do projeto.

Como pode ser visto na Figura 4.1, este sistema de distribuição e recebimento de óleo está centrado no terminal e possui três agentes independentes (chamados de atores em UML): o navio petroleiro, o próprio terminal e as refinarias. Os atores do sistema interagem para realizar as ações necessárias para levar o óleo cru dos petroleiros até as refinarias.

Foram identificados onze casos de uso para o sistema: *Atracar no píer*, *Desatracar do píer*, *Receber determinado óleo cru de um navio em um tanque*, *Conectar um tanque a um navio*, *Bombear óleo do navio para o tanque*, *Desconectar um tanque de um navio*, *Distribuir óleo cru para um refinaria a partir de um tanque*, *Conectar um tanque e uma refinaria através de um oleoduto*, *Bombear óleo cru através do oleoduto*, *Desconectar um tanque e uma refinaria de um oleoduto* e *Consumir óleo cru*.

A descrição detalhada (pré-condições, pos-condições, invariantes, restrições temporais, etc.) pode ser encontrada no Anexo A deste relatório. A seguir serão apresentadas breves descrições de cada um desses casos de uso:

- **Atracar no píer.** Os navios que chegam ao terminal petrolífero são alocados aos píeres para que seus tanques sejam descarregados.
- **Desatracar do píer.** Após ter sido descarregado, o navio deixa o píer livre para que outra embarcação possa atracar.
- **Receber determinado óleo cru de um navio em um tanque.** Os inventários de óleo cru dos navios petroleiros que chegam ao terminal petrolífero são descarregados em píeres e armazenados em tanques.
- **Conectar um navio a um tanque.** A conexão permite que o óleo cru do navio possa ser bombeado para os tanques do porto. O bombeamento nunca pode ocorrer antes que os dois estejam conectados.
- **Bombear óleo cru do navio para o tanque.** Durante o bombeamento, um determinado tipo de óleo cru do navio é transferido para um tanque do porto. A vazão máxima da bomba depende do tipo de óleo cru sendo descarregado.

- **Desconectar um tanque de um navio.** A desconexão entre o navio e o tanque ocorre após o término da operação de bombeamento, liberando o navio e o tanque para uma nova operação.
- **Distribuir óleo cru para a refinaria a partir de um tanque.** O petróleo recebido no terminal petrolífero é distribuído para as refinarias através de oleodutos. Cada refinaria está conectada ao porto através de um único oleoduto.
- **Conectar um tanque e uma refinaria através de um oleoduto.** A conexão permite que o óleo cru do tanque possa ser bombeado, através do oleoduto, para a refinaria. O bombeamento nunca pode ocorrer antes que os dois estejam conectados.
- **Bombear óleo cru através do oleoduto.** Durante o bombeamento, óleo cru do tanque é transferido para a refinaria. A vazão máxima da bomba depende da classe do óleo cru sendo transportado.
- **Desconectar um tanque e uma refinaria de um oleoduto.** A desconexão entre o oleoduto e o tanque e entre o oleoduto e a refinaria ocorre após o término da operação de transporte, liberando o oleoduto, o tanque e a refinaria para uma nova operação.
- **Consumir óleo cru.** as refinarias serão modeladas como consumidoras de óleo cru do sistema. Será admitido que, no planejamento de curto prazo de suprimento de óleo cru, as refinarias já tenham preestabelecido o consumo médio de óleo cru no período.

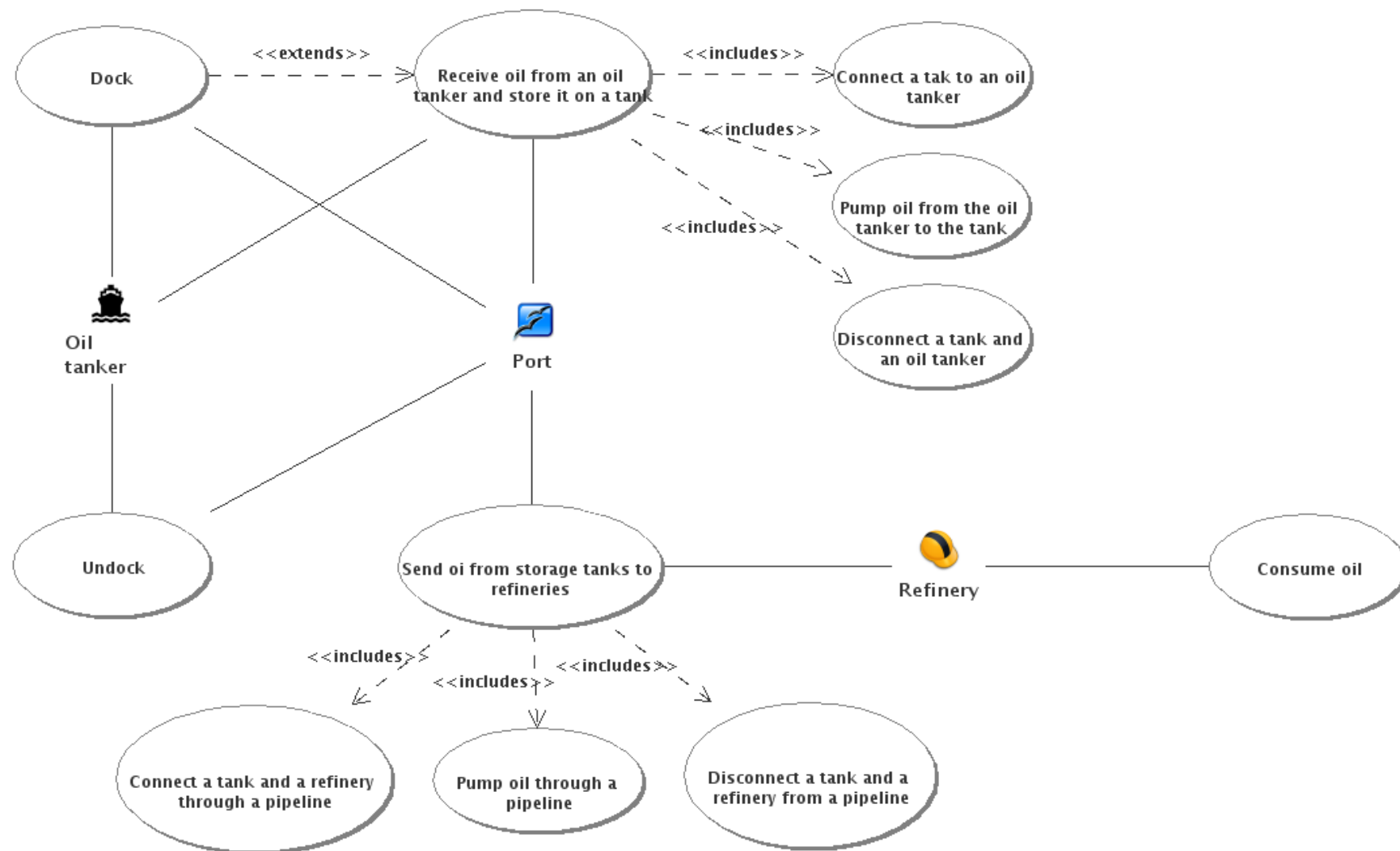


Figura 4.3 - Diagrama de casos de uso

4.4. Classes

O passo seguinte do processo de modelagem consiste na definição das classes do sistema e seu diagrama. O diagrama de Classes é uma representação da estrutura estática do domínio de planejamento que mostra as entidades existentes, suas relações, suas características, operadores (ações) e restrições. Um atributo e uma associação entre Classes fornecem uma noção visual da semântica do modelo. É importante frisar que o diagrama de Classes não representa um problema de planejamento em particular, mas um conjunto de problemas de planejamento.

A Figura 4.4 mostra o diagrama de classes desenvolvido para o problema de recebimento e distribuição de óleo no terminal de São Sebastião. O diagrama consiste de oito classes que modelam todas as entidades relevantes para o problema que está sendo modelado. A classe *Domain_Metrics* é uma classe do tipo *utility* que guarda informações relevantes para todas as outras classes no modelo (por exemplo, o custo de interface). Neste caso particular, ela corresponde à variáveis que são utilizadas como métrica para os planos gerados pelo modelo: perdas de receita no porto e na refinaria (devido a má alocação do óleo cru aos tanques), custo de interface e número de ações temporizadas utilizadas no plano.

As refinarias não foram modeladas individualmente, mas como uma classe do tipo *utility*. Apesar de o tempo contínuo poder ser descrito atualmente na PDDL, nenhum planejador é capaz de lidar com variáveis contínuas no tempo. Como a refinaria consome de forma contínua, sua modelagem ainda não é viável com os planejadores gerais. Assim, para tentar contornar este problema, a classe Refinaria controla o volume de óleo que deve ser enviado às refinarias durante o plano, que é definido pelo usuário.

Aqui, pode-se notar a importância dos casos de uso como etapa anterior à construção do diagrama de classes: muitas das classes surgem naturalmente durante a definição dos casos de uso.

Neste ponto, deve-se mencionar o caráter iterativo do processo de modelagem: a cada nova etapa, são realizadas alterações nas etapas anteriores. Por exemplo, durante a construção do diagrama de classes foram realizadas alterações no diagrama de casos

de uso. Isto ocorrerá em todo o processo, pois cada nova etapa mostra aspectos que não estavam evidenciados nas fases anteriores.

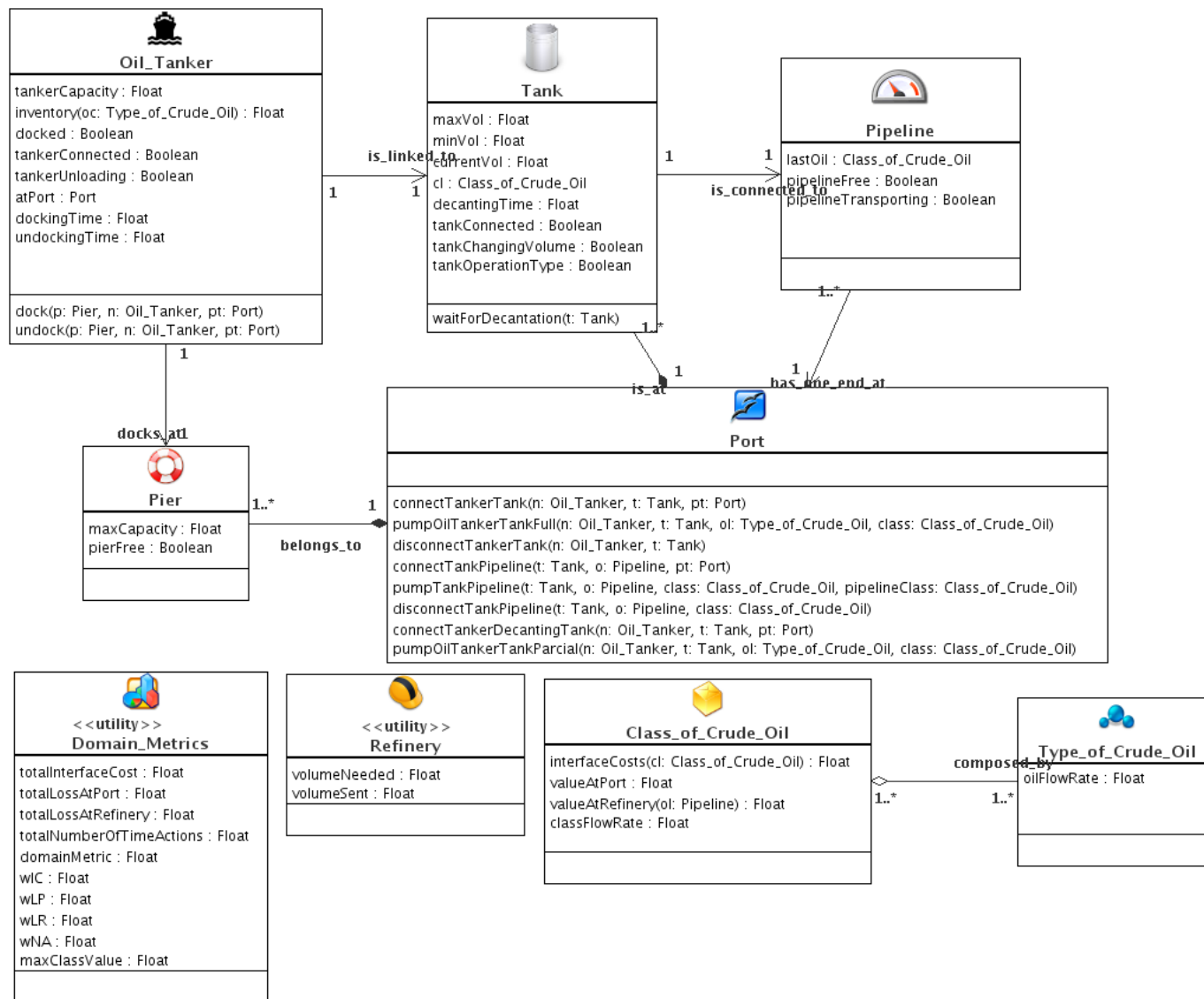


Figura 4.4 Diagrama de classes

4.5. Diagrama de estados

O comportamento dinâmico das ações é especificado no diagrama de Estados (também chamado de diagrama de Máquina de Estados), onde é possível definir as pré e pós condições para os operadores definidos no diagrama de classes. Este diagrama é muito útil quando deseja-se representar as entidades do sistema que possuem comportamento dinâmico. Na UML.P, todas as pré e pós condições são definidas através de uma linguagem formal chamada OCL (OMG 2003), que é uma linguagem predefinida da UML.

Normalmente cada classe em um diagrama de Classes (especialmente aqueles que mudam de estado durante a execução do plano) tem seu próprio diagrama de Estados. Um diagrama de Estados não tem como objetivo especificar todas as mudanças causadas por uma ação: ele detalha somente as mudanças que ela causa no objeto de uma classe específica.

No modelo do Terminal de São Sebastião foram definidos diagramas de estado para as classes *Navio*, *Tanque*, *Pier*, *Oleoduto*, *Porto*. A seguir, estes diagramas serão apresentados com mais detalhes.

4.5.1. Navio

O diagrama na Figura 4.5 modela as atividades de um navio, desde o momento em que ele chega ao terminal de São Sebastião até a sua saída, depois de descarregado. O navio possui cinco estados possíveis: *Fora do porto*, *Desatracado*, *Atracado e Desconectado*, *Atracado e Conectado*, *Descarregando*.

Uma vez no porto (estado *Desatracado*), o navio passa a esperar que um píer seja liberado para que possa atracar. Caso exista um píer livre que suporte o calado do navio, ele poderá atracar e iniciar o processo de descarregamento. Após esse período, o navio passa a estar atracado e esperar um tanque disponível para iniciar o descarregamento. O navio permanece atracado até que seja totalmente descarregado. Uma vez descarregado, o navio pode sair do porto, liberando o píer para o próximo petroleiro que será descarregado.

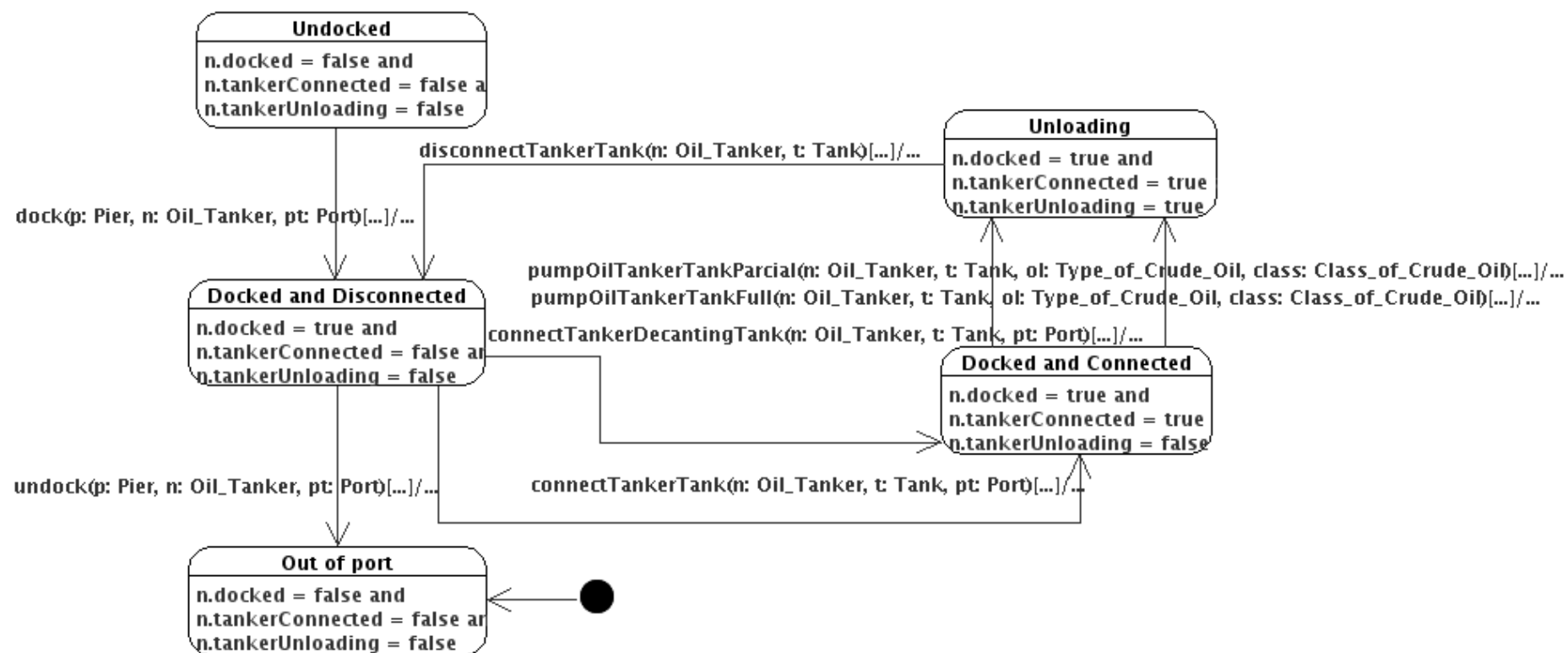


Figura 4.5 Diagrama de estados do Navio

4.5.2. Píer

O diagrama na Figura 4.6 modela as atividades de um píer do terminal de São Sebastião durante o processo de descarregamento de navios. O píer possui dois estados possíveis: *Livre*, *Ocupado*.

A dinâmica do Píer é muito simples: quando o navio atraca em algum píer, ele passa a estar ocupado e impedido de receber qualquer outra embarcação. A partir do momento que o navio desatraca, o píer esta livre para receber outro navio que necessite ser descarregado.

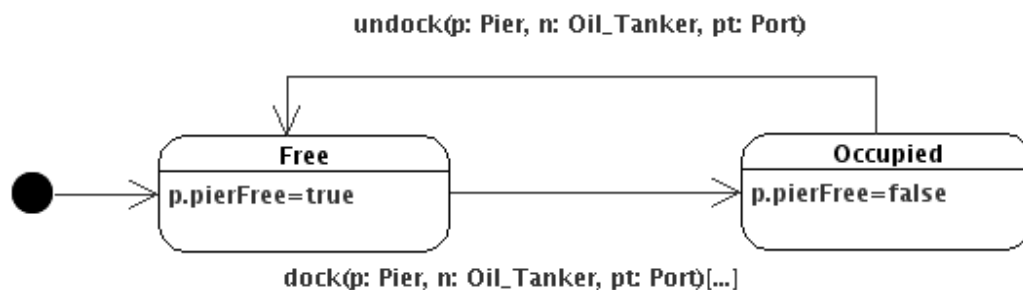


Figura 4.6 Diagrama de estados do *Píer*

4.5.3. Tanque

O diagrama na Figura 4.7 modela as atividades de descarregamento e carregamento de um tanque do terminal de São Sebastião. O tanque pode estar em seis estados diferentes: *Livre*, *Conectado a um navio*, *Carregando*, *Decantando*, *Conectado a um oleoduto*, *Descarregando*.

Um tanque nunca pode estar carregando e descarregando ao mesmo tempo. Portanto, uma vez alocado a um navio, este tanque deixa de estar livre para o descarregamento, e vice versa.

Após seu carregamento, o tanque entra em um período de decantação: período durante o qual não pode ser descarregado. No estágio atual este período é igual a zero: em outras palavras, apesar de passar por este estado, o tempo de decantação não é considerado. No entanto, um tanque que se encontra em decantação pode voltar a ser

carregado, caso seu volume atual seja menor que sua capacidade máxima. Da mesma forma, um tanque só pode ser descarregado se seu volume for maior que o lastro mínimo.

4.5.4. Oleoduto

O diagrama na Figura 4.8 modela as atividades de um oleoduto associado ao terminal de São Sebastião durante o processo de descarregamento de tanques. O oleoduto possui três estados possíveis: *Livre*, *Conectado a um tanque* e *a uma refinaria*, *Enviando óleo do tanque para a refinaria*.

A dinâmica do oleoduto, assim como a do píer, é muito simples: para que o óleo possa ser enviado para uma refinaria o oleoduto deve ser conectado a um tanque e à refinaria em questão. Uma vez iniciado o descarregamento, o oleoduto volta a estar livre para ser utilizado em outra operação uma vez que o volume desejado de óleo cru tenha sido transmitido para a refinaria.

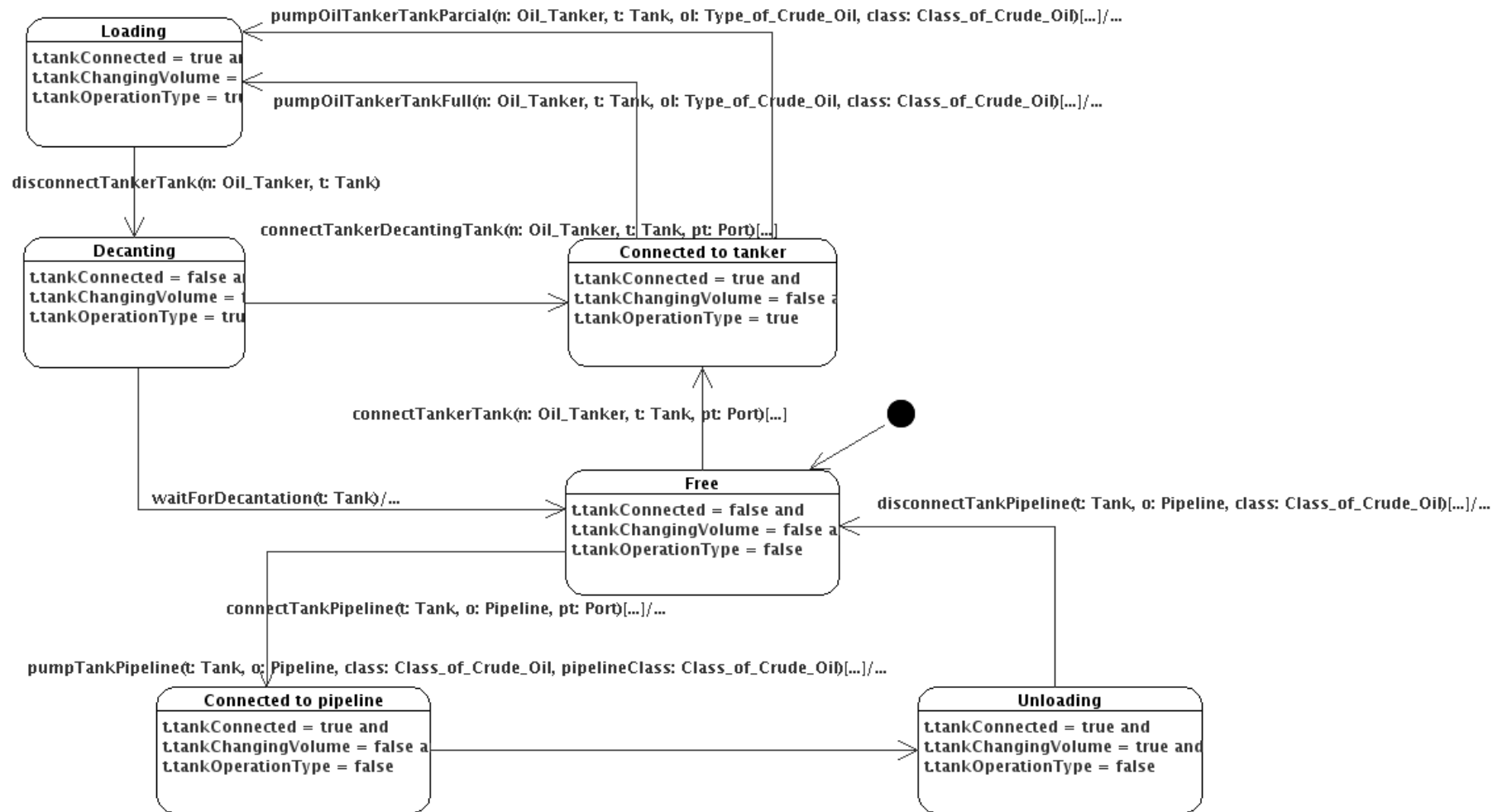


Figura 4.7 Diagrama de estados do *Tanque*

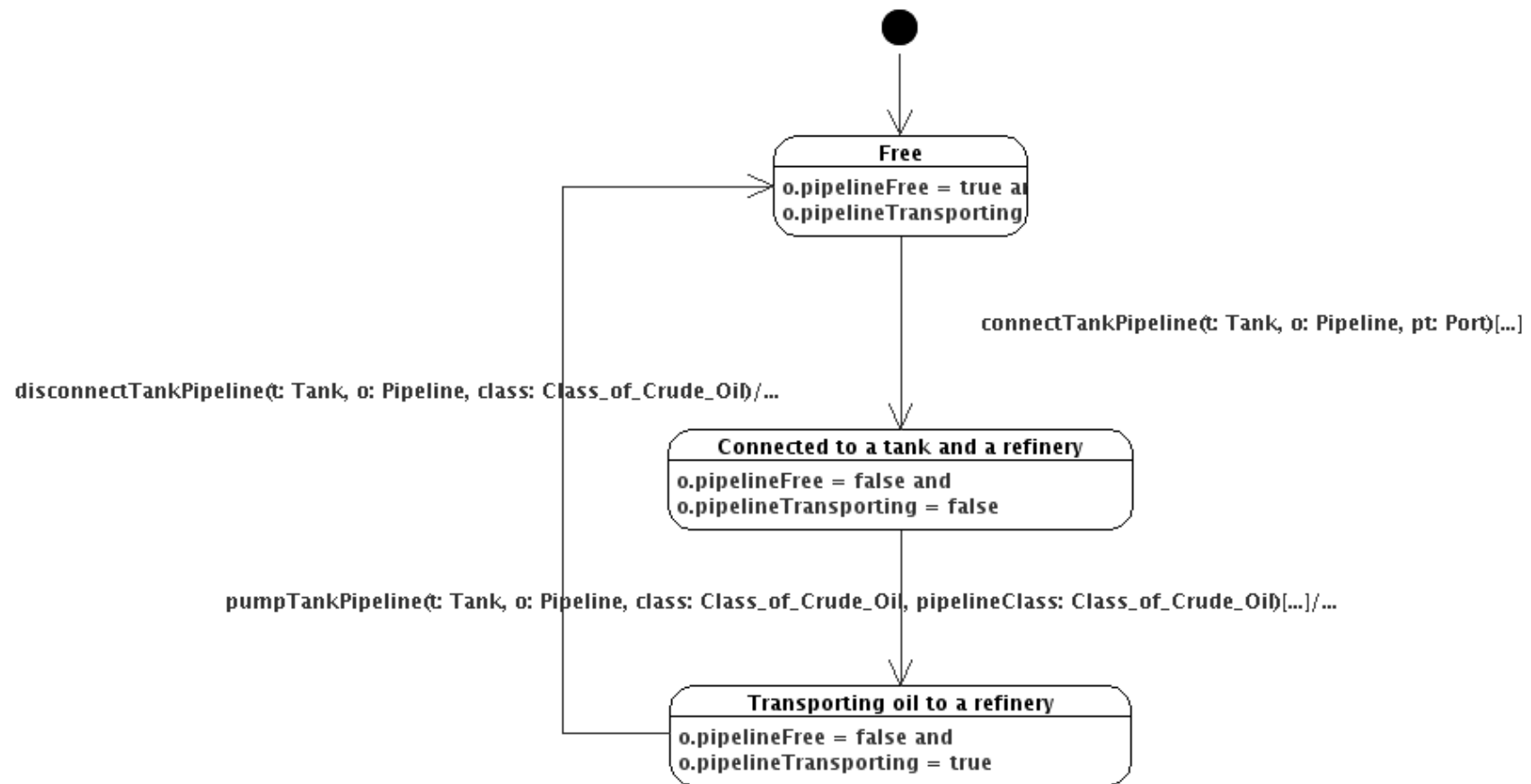


Figura 4.8 Diagrama de estados do *Oleoduto*

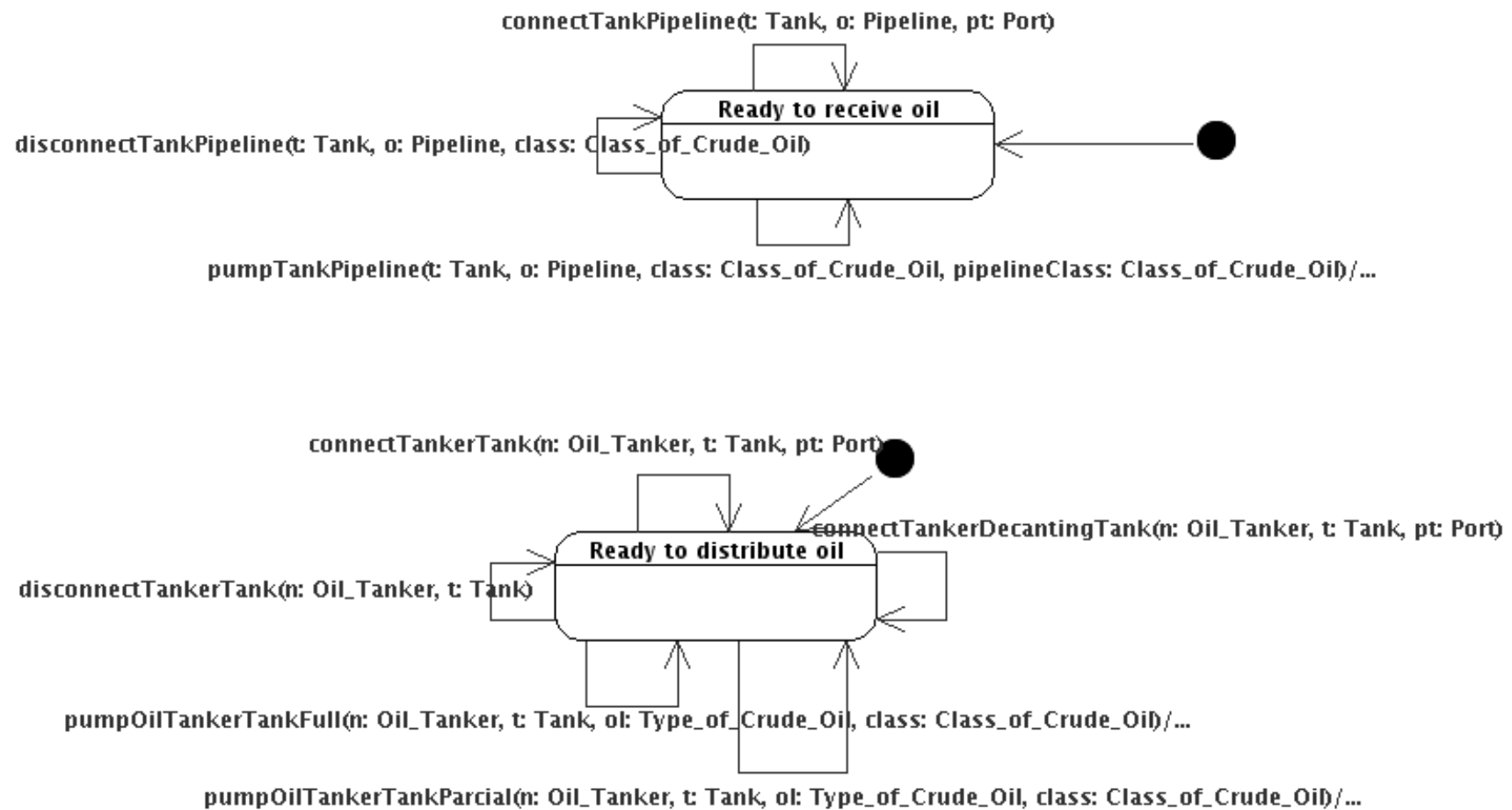


Figura 4.9 Diagrama de estados do *Porto*

5. RESULTADOS DO MODELO DE PORTO

Uma vez construído o modelo de um problema, passa-se para a próxima fase no ciclo de um projeto, que consiste em sua validação. Dessa forma, o modelo desenvolvido no capítulo 4 foi testado através da construção de problemas simplificados e seu teste posterior nos planejadores disponíveis no mundo acadêmico.

Outro objetivo desta fase do projeto é avaliar a capacidade dos planejadores atuais de resolverem o problema proposto. Devemos avaliar a capacidade das ferramentas disponíveis atualmente para verificar se o modelo deve ser simplificado e que tipo de melhoria deveria ser feito para incrementar o desempenho desses planejadores ao resolverem o problema proposto.

Neste capítulo, serão apresentados os resultados dos testes realizados no modelo do terminal Portuário de São Sebastião.

5.1. Modelo Seqüencial (sem tempo)

Apesar do problema real tratado neste trabalho requerer o planejamento com tempo para levar em consideração à concorrência e o paralelismo inerentes ao sistema, a inclusão desta variável tem um impacto importante no tempo computacional. Assim, como muitos parâmetros podem ser avaliados sem a presença do tempo no modelo, a primeira avaliação do modelo foi realizada sem levar em consideração o tempo de cada ação.

Nesta fase do trabalho, o planejador deveria achar uma seqüência de ações que levasse o sistema do estado final ao estado inicial sem se preocupar com o instante em que elas aconteceriam. O objetivo deste testes era avaliar se as regras básicas do sistema estavam sendo respeitadas. Todas as restrições operacionais envolvidas no sistema podem ser validadas sem levar em consideração o tempo. Além disso, os testes realizados nesta seção buscavam avaliar a sensibilidade do planejador ao número de objetos no problema e ao número de ações necessárias para atingir os objetivos do problema.

5.1.1. Escolha dos Planejadores

Como mencionado anteriormente, o objetivo deste trabalho é desenvolver um modelo para solução do problema do recebimento e distribuição de petróleo a partir de um terminal petrolífero. Por tanto, alterar ou desenvolver um planejador para esta aplicação, apesar de ser um passo importante, está fora do escopo deste projeto. Assim, torna-se necessário escolher planejadores disponíveis que possam ser utilizados para a resolução dos problemas propostos.

Os planejadores disponíveis na comunidade acadêmica são chamados de planejadores gerais, pois podem ser utilizados para resolução de qualquer problema, desde que os *requirements* do domínio sejam suportados pelo planejador. Assim, apesar de o tempo não ser um requisito nesta fase, o modelo é intensivo em variáveis numéricas, sendo necessário encontrar planejadores que as suportem.

A lista abaixo enumera os planejadores que participaram com sucesso da IPC (*International Planning Competition*) de 2002, 2004 e 2006 na seção *Temporal/Metric*:

- Metric-FF (HOFFMAN, 2006)
- MIPS-XXL (EDELKAMP; JABBAR; NAIZIH, 2006)
- SGPLAN 5 (HSU *et al.*, 2006)
- YochanPS (BENTON; KAMBHAMPATI; DO, 2006)
- LPG-TD (GEREVINI; SAETTI; SERINA, 2004)
- TP-4 (HASLUM, 2004)
- HSP*a (HASLUM, 2004)
- FF (HOFFMAN; NEBEL, 2001)
- SHOP2 (NAU *et al.*, 2001)
- SemSyn (PARKER, 1999)
- TLPlan (BACCHUS; KABANZA, 2000)

Cada um destes planejadores foi testado com problemas simples para verificar se, além de variáveis numéricas, seriam capazes de lidar com os outros aspectos do sistema e gerar soluções plausíveis. Após a realização destes testes, verificou-se que

apenas três planejadores tiveram um desempenho satisfatório nesta fase do projeto: *SGPLAN*, *Metric-FF* e *MIPS-XXL*.

5.1.2. Cenários

Dois cenários foram considerados nesta fase: um está relacionado a um exemplo de problema real e o outro consiste em problemas fictícios mais simples. Este último é utilizado para verificar como os planejadores respondem a determinadas variáveis (como o volume de óleo cru nos navios), enquanto que o primeiro testa o comportamento do planejador a medida que nos aproximamos de problemas reais.

5.1.2.1. Cenário 1

O problema real apresentado neste trabalho é um exemplo de uma situação no terminal de São Sebastião considerando um horizonte de planejamento de uma semana e foi retirado de (MÁS, 2001). O problema consiste de treze petroleiros, carregando quatorze tipos diferentes de óleos crus, que estão programados para chegar no porto para serem inteiramente descarregados.

Neste exemplo, o porto utiliza dezoito tanques (dos vinte e um disponíveis) para armazenar os óleos provenientes dos navios petroleiros. Estes óleos crus estão agrupados em sete classes diferentes e cada tanque é capaz de armazenar uma única classe de óleo cru (deve-se lembrar que um mesmo óleo pode fazer parte de mais de uma classe).

O descarregamento dos petroleiros é realizado nos quatro píeres presentes no terminal. O óleo é, então, enviado para as refinarias através de dois oleodutos que saem do terminal.

O propósito deste cenário é avaliar como os planejadores atuais respondem à medida que nos aproximamos de uma situação real. Para fazer isto, treze problemas foram criados onde o número de navios petroleiros aumenta de um até treze. Além disso, um objetivo de otimização também foi imposto ao planejador: a minimização do custo de interface durante a operação de distribuição.

5.1.2.2. Cenário 2

Os problemas pertencentes a estes cenários são mais simples e foram concebidos para medir como o planejador se comporta à medida que o volume de óleo nos tanques aumenta. À medida que o inventário de óleo cru nos navios cresce, o planejador é forçado a realizar mais ações para conseguir descarregar todos os navios no porto. Isto acontece pois, como os tanques possuem uma capacidade limitada, torna-se necessário realizar mais operações de descarregamento dos tanques, aumentando o número de ações necessárias para encontrar uma solução.

Os problemas neste cenário consistem de três tanques carregando volumes iguais de um determinado tipo de óleo cru. O terminal possui um píer e cinco tanques para realizar as operações de descarregamentos. Todos os tanques possuem a mesma capacidade e armazenam a mesma classe de óleo cru.

A variável que caracteriza cada problema neste cenário é o número de tanques necessários para realizar o descarregamento de todos os navios. Esta variável pode ser calculada pela razão entre o volume de óleo nos tanques e o volume de óleo nos navios. Esta razão varia de um até dez nos problemas presentes neste cenário.

5.1.3. Resultados do Modelo Seqüencial

Os planejadores foram testados em um computador com um processador Intel Pentium IV e 1.0 gigabytes de memória, cujo sistema operacional era Fedora Linux. As tabelas 5.1 e 5.2 mostram os resultados dos testes no Cenário 1 e a Tabela 5.3 apresenta os resultados do cenário 2. Todas as tabelas mostram o número de tanques e navios petroleiros no problema, o número de ações no plano e o tempo computacional utilizado pelo planejador para encontrar a solução. Todos os experimentos foram finalizados caso o planejador não encontrasse a solução em até dez minutos: o símbolo “TIME” nas tabelas indicam os casos onde isto ocorreu. Os casos onde o planejador finalizou a sua execução sem encontrar uma solução aparecem com um “X” nas tabelas.

Além disso, a Tabela 5.2 apresenta os resultados de problemas que incluíam a métrica relacionada à minimização do custo de interface total da operação, e a Tabela

5.3 os resultados de performance de cada planejador e o volume de óleo em cada navio do problema.

Como MIPS-XXL não mostra o tempo em seu arquivo de resultados, as tabelas não mostram nenhuma informação para este item em particular. Mesmo assim, deve-se mencionar que, durante os testes, este planejador pareceu necessitar de um tempo computacional maior que o SGPlan e o Metric-FF na maioria dos problemas.

A Tabela 5.1 mostra que o Metric-FF e o MIPS-XXL demonstraram dificuldades ao resolver a maioria dos problemas no cenário realista sem métrica, enquanto que o SGPlan conseguiu encontrar soluções para praticamente todos os problemas propostos neste experimento. No entanto, encontrar uma solução não é de grande utilidade para esta aplicação em particular, já que o problema é encontrar a solução que maximize os lucros.

Quando a métrica é adicionada ao problema, apesar do SGPlan continuar encontrando soluções para os problemas propostos, os resultados mostram claramente que ele não realiza nenhuma otimização durante a resolução do problema. O tempo computacional e o número de ações nas soluções continuam os mesmos. O Metric-FF gerou mensagens de erro mesmo antes de iniciar a busca por uma solução e o MILP-XXL não foi capaz de resolver nenhum dos problemas quando o critério de minimização foi utilizado. Neste último caso, o planejador possuía uma limitação interna de 300 segundos para encontrar uma solução. Mesmo assim, como o problema sem métrica foi resolvido em menos de 0,5 segundos tanto pelo Metric-FF como o SGPlan, 300 segundos representa um aumento notável no tempo computacional, o que mostra como o critério de otimização pode afetar a performance dos planejadores.

Estes resultados também mostram que o SGPlan teve dificuldades para resolver alguns problemas em particular. Isto pode ser explicado pelo fato deste planejador considerar que muitas aplicações de planejamento possuem uma estrutura de restrições em *clusters* e, portanto, o objetivo pode ser subdividido em sub-objetivos baseado nestes clusters. Cada sub-objetivo é resolvido por uma versão modificada do planejador FF e depois, uma análise global resolve inconsistências em restrições globais através da formulação de uma penalidade.

Os problemas que causaram TIME na execução do SGPlan podem ter muitas causas. Uma delas é que estes problemas podem ter afetado o agrupamento das restrições e, conseqüentemente, os sub-problemas herdaram erros de partição que inviabilizaram o sistema para achar uma solução no tempo dado. Outra explicação possível é a utilização do Metric-FF para resolver os sub-problemas. O metric-FF não resolveu o mesmo problema em nossos experimentos e, portanto, possui uma grande probabilidade de não conseguir resolver um dos sub-problemas. Isto se torna ainda mais verdadeiro se considerarmos que a partição poderia conter um erro.

O Metric-FF segue os princípios do planejador FF. O planejador FF constrói uma rede inteira com todos os predicados (*fluents*) e todas as ações, assim como as ligações entre elas. Esta rede pode resultar em *overflows* de memória se tivermos muitos objetos. Esta pode ser uma causa para os erros de FLUENTS observados nos experimentos.

Tabela 5.1: Problema realista sem métrica

# of Tanks	# of Tankers	Metric FF		SGPLAN		MIPS	
		Actions	Time	Actions	Time	Action	Time
18	1	7	0,46	7	0,46	13	-
18	2	17	1,82	17	0,97	25	-
18	3	30	3,24	31	4,66	34	-
18	4	X	X	62	54,18	TIME	-
18	5	TIME	TIME	75	120,48	TIME	-
18	6	TIME	TIME	78	42,74	TIME	-
18	7	TIME	TIME	TIME	TIME	TIME	-
18	8	TIME	TIME	97	74,43	TIME	-
18	9	TIME	TIME	TIME	TIME	TIME	-
18	10	TIME	TIME	TIME	TIME	TIME	-
18	11	TIME	TIME	115	303,78	TIME	-
18	12	TIME	TIME	126	386,72	TIME	-
18	13	TIME	TIME	132	478,73	TIME	-

Tabela 5.2: Problema realista com métrica

# of Tankers	# of Tanks	Metric FF			SGPLAN			MIPS		
		Actions	Time	Metric	Actions	Time	Metric	Actions	Time	Metric
1	18	-	-	-	7	0,34	0,00	X	-	X
2	18	-	-	-	17	1,86	0,00	X	-	X
3	18	-	-	-	31	4,70	2,20	X	-	X
4	18	-	-	-	62	55,13	5,66	X	-	X
5	18	-	-	-	75	121,00	3,15	X	-	X
6	18	-	-	-	78	42,88	4,17	X	-	X
7	18	-	-	-	TIME	TIME	TIME	X	-	X
8	18	-	-	-	97	74,43	8,805	X	-	X
9	18	-	-	-	TIME	TIME	TIME	X	-	X
10	18	-	-	-	TIME	TIME	TIME	X	-	X
11	18	-	-	-	115	303,78	-	X	-	X
12	18	-	-	-	126	386,72	-	X	-	X
13	18	-	-	-	132	478,73	-	X	-	X

Tabela 5.3: Problema simples com variação no volume do inventário dos navios

# Tanks	# Tankers	Oil inventory in tanker (m ³)	Metric FF		SGPLAN		MIPS	
			Actions	Time	Actions	Time	Actions	Time
5	6	45000	41	0,48	38	0,34	38	-
5	6	90000	83	12,59	87	0,23	83	-
5	6	135000	TIME	TIME	131	0,85	TIME	-
5	6	180000	TIME	TIME	197	2,79	TIME	-
5	6	225000	TIME	TIME	248	9,75	TIME	-
5	6	270000	TIME	TIME	314	11,73	TIME	-
5	6	315000	TIME	TIME	332	7,10	TIME	-
5	6	360000	TIME	TIME	395	20,16	TIME	-
5	6	405000	TIME	TIME	413	14,53	TIME	-
5	6	450000	TIME	TIME	416	15,35	TIME	-

5.1. Modelo de Porto com Tempo

Neste capítulo, através da análise dos resultados de dois exemplos, procura-se avaliar a o *Modelo de Porto* proposto, sob o ponto de vista da eficiência computacional e capacidade de geração de soluções eficientes. Apesar de serem mais simples, os problemas sugeridos nesta seção se aproximam muito de uma situação real no Porto de

São Sebastião. Estes exemplos foram baseados em experimentos similares realizados por (MÁS, 2001).

Para cada um dos exemplos adota-se a seguinte sequência: inicialmente os cenários são apresentados; a seguir, os resultados obtidos durante a simulação são descritos. Finalmente, realiza-se uma análise dos planos em relação à eficiência computacional e viabilidade das respostas obtidas.

5.2.1. Escolha dos Planejadores

Na seção anterior, foi realizada uma análise de diversos planejadores capazes de lidar com variáveis numéricas para verificar quais seriam capazes de resolver os problemas propostos neste trabalho. Foi analisada uma lista de 11 planejadores gerais e apenas 3 planejadores e mostraram capazes de gerar soluções viáveis: *SGPlan*, *Metric-FF* e *MIPS-XXL*.

Ao introduzirmos o tempo (que trás problemas de planejamento paralelo e concorrência) verificamos que o *Metric-FF* não conseguia mais lidar com os desafios apresentados pelo domínio. Dessa forma, restaram dois planejadores que entendiam ações temporizadas na PDDL: *SGPlan* e *MIPS-XXL*.

Testes foram realizados nestes dois planejadores para verificar quais suas limitações ao lidar com domínios temporizados. Apesar de a PDDL2.1 possuir uma capacidade de representação muito ampla no que se refere ao tempo (ações com tempo discreto, ações com tempo discreto não definido, ações em tempo contínuo e métricas para minimização do tempo total do plano, por exemplo), poucos planejadores à implementaram de forma completa.

Verificou-se que nenhum planejador foi capaz de lidar com ações contínuas ou ações discretas de duração indefinida. Dessa forma, todas as ações temporizadas no modelo deveriam ter sua duração estabelecida a priori. Isto limita muito a qualidade das respostas para o problema proposto neste trabalho, pois o ideal seria que o planejador determinasse a duração das operações de carga e descarga de tanques. Por isso, foi necessária a adoção da hipótese de carga e descarga completa em todas as operações como mencionado anteriormente.

Além disso, apenas o MIPS-XXL se mostrou capaz de realizar otimizações relacionadas à duração total do plano. Na verdade, deve-se mencionar o segundo planejador considerado, *SGPlan*, não possui nenhum suporte a otimizações, seja ela relacionada ao tempo ou não. No entanto, na avaliação dos planos gerados pelo MIPS-XXL, verificou-se que o planejador muitas vezes não respeitava a duração das ações e, portanto, gerava seqüências de ações inviáveis.

Finalmente, apenas um planejador se mostrou como uma opção viável para resolver o problema proposto: *SGPlan*. No entanto, apesar de gerar soluções viáveis, este planejador não realiza nenhum tipo de otimização.

5.2.2. A Métrica Utilizada

Como mencionado na seção anterior, o *SGPlan* não realiza nenhum tipo de otimização, especialmente com relação ao tempo. No entanto, para variáveis não temporais, adotou-se uma estratégia que permitia obtenção de planos mais próximos ao que deveria ser o ótimo. A abordagem utilizada consistia de realizar iterações (resolvendo o mesmo problema diversas vezes) e, a cada passo, exigia-se do planejador que encontrasse uma métrica melhor que a métrica do plano anterior.

Um plano ótimo deve considerar seis variáveis: *Receita de óleo armazenado nas refinarias*, que mede o valor do óleo que foi enviado a refinaria; a *Receita de óleo no armazenado no porto*, que mede o valor do óleo armazenado nos tanques do terminal (o óleo tem mais valor nas refinarias que no porto); o *Custo de óleo cru dos navios*, que corresponde ao custo da matéria-prima, o petróleo, do sistema; o *Custo de operação dos píeres*, que depende do píer e do tempo que cada navio passou nos píeres; o *Custo de sobreestadia*, que mede o custo do tempo, além do permitido, que cada navio permaneceu atracado; e o *Custo de interface*, que mede as perdas na interface entre os óleos no oleoduto. O planejador deve maximizar as receitas e diminuir os custos.

O custo de sobreestadia e o custo de utilização dos píeres não foram considerados, pois, como o planejador não consegue otimizar variáveis temporais, não foi possível incluí-los no modelo e, por conseguinte, na métrica. Assim, apenas quatro componentes foram otimizados nos planos: *Receita de óleo armazenado nas refinarias*,

Receita de óleo armazenada no porto, Custo de interface. O *Custo de óleo cru dos navios* não foi considerado, pois é definido pelos inventários dos navios, que é um dado do problema e, por tanto, não pode ser minimizado. A métrica também considerou o número de ações temporizadas utilizadas no plano, em uma tentativa de minimizar o tempo total.

A utilização de métricas é sutil e pode ter um impacto dramático nos planos gerados. Talvez o caso mais simples se aquele onde todas as ações aumentam uma métrica que deve ser minimizada, ou reduzem uma que deve ser maximizada. Esta situação parece ser relativamente simples: o planejador deve tentar utilizar o menor número de ações possível para resolver o problema. Um caso mais complexo surge quando algumas ações contribuem com a métrica enquanto outros a pioram. Neste caso, o planejador pode tentar utilizar ações que aumentem a qualidade da métrica sem que essas ações contribuam para atingir o objetivo. Os testes realizados nos planejadores atuais mostraram que eles não conseguem lidar com o segundo caso e, por tanto, foi necessário desenvolver um métrica mais simples que se movesse em uma única direção.

A seguir serão apresentados os diferentes componentes da métrica:

- ***totalInterfaceCost.*** este componente simplesmente soma todos os custos de interface incorridos durante o plano.
- ***totalLossAtPort.*** A receita de óleo armazenado no porto depende da alocação do óleo dos navios nos tanques de classe mais nobre. Dessa forma, para evitar maximizar a receita, verifica-se quanta receita foi perdida por não ter armazenado o óleo em tanques com classes de maior valor.
- ***totalLossAtRefinery.*** A receita de óleo armazenado na refinaria depende do envio dos óleos mais nobres à refinaria. Dessa forma, para evitar maximizar a receita, verifica-se quanta receita foi perdida por não ter enviado óleo de tanques com classes de maior valor.
- ***totalNumberOfTimeActions.*** Esta componente mede o número de ações temporizadas utilizada no plano, em uma tentativa de minimizar o tempo total.

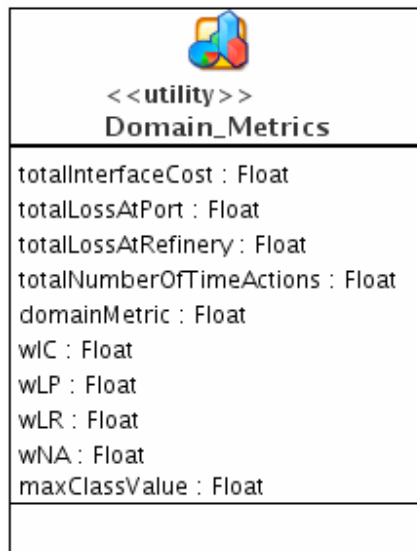


Figura 5.1 – Classe *Domain_Metrics*

Cada um dos componentes mencionados anteriormente possui pesos utilizados no cálculo da métrica total. As variáveis wIC , wLP , wLR e wNA correspondem aos pesos com componentes *totalInterfaceCost*, *totalLossAtPort*, *totalLossAtRefinery* e *totalNumberOfTimeActions* respectivamente.

5.2.2. Exemplo 1

Este problema, relativamente simples, envolve um porto que recebe três navios, os quais devem ser descarregados em cinco tanques e atracados em dois píeres disponíveis. Os tanques podem ser descarregados, desde que obedeçam o período de decantação, de forma a abastecer a refinaria conectada ao porto por meio de um oleoduto. O problema é ilustrado na Figura 5.2, onde o fluxo de óleo cru no sistema é indicado pelas setas.

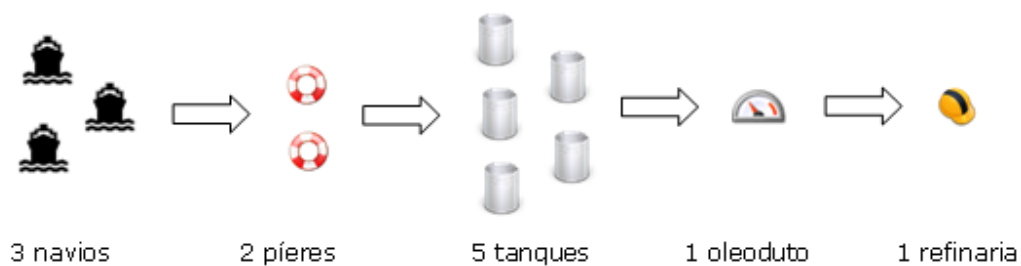


Figura 5.2 – Ilustração do exemplo 1

5.2.2.1. Estado inicial e Dados de Entrada

O estado inicial e, portanto, os dados de entrada, estão dispostos nas Figura 5.3 e 5.4. O problema configura-se, basicamente, da seguinte forma: três navios chegam carregados de óleo cru ao terminal. Dois deles devem descarregar um tipo de óleo cru cada, enquanto o terceiro possui dois tipos distintos de óleo cru em seu inventário. Estes navios podem atracar em dois píeres, P1 e P2. Cinco tanques estão disponíveis para receber óleo cru no sistema, sendo que cada um deles estoca uma classe distinta de petróleo, gerando diferentes níveis de receita por unidade de volume. Após ser carregado, cada tanque deve ficar inoperante por um período de 24 horas para executar o procedimento de decantação do inventário de óleo cru descarregado. Finalmente, através de um oleoduto, estes tanques devem abastecer uma refinaria, basicamente modelada como um grande armazém.

As unidades utilizadas, nos exemplos deste capítulo, para tempo, custos/receitas (\$) e volume são, respectivamente, horas, milhares de dólares e m^3 .

Vale destacar que os dados apresentados, assim como os utilizados em outros exemplos a serem analisados, são parâmetros relacionados a dados reais.

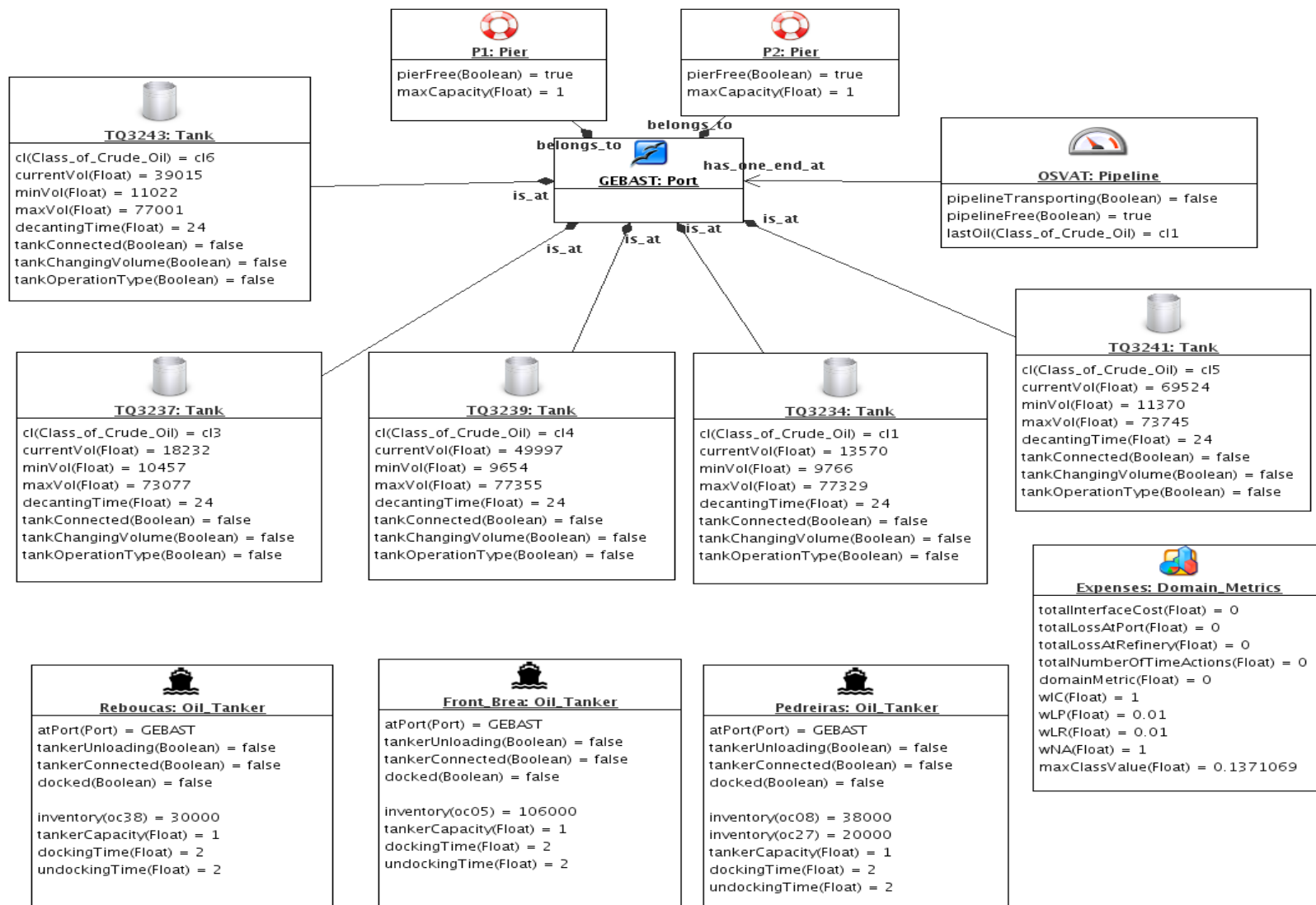


Figura 5.3 – Snapshot do estado inicial do exemplo 1

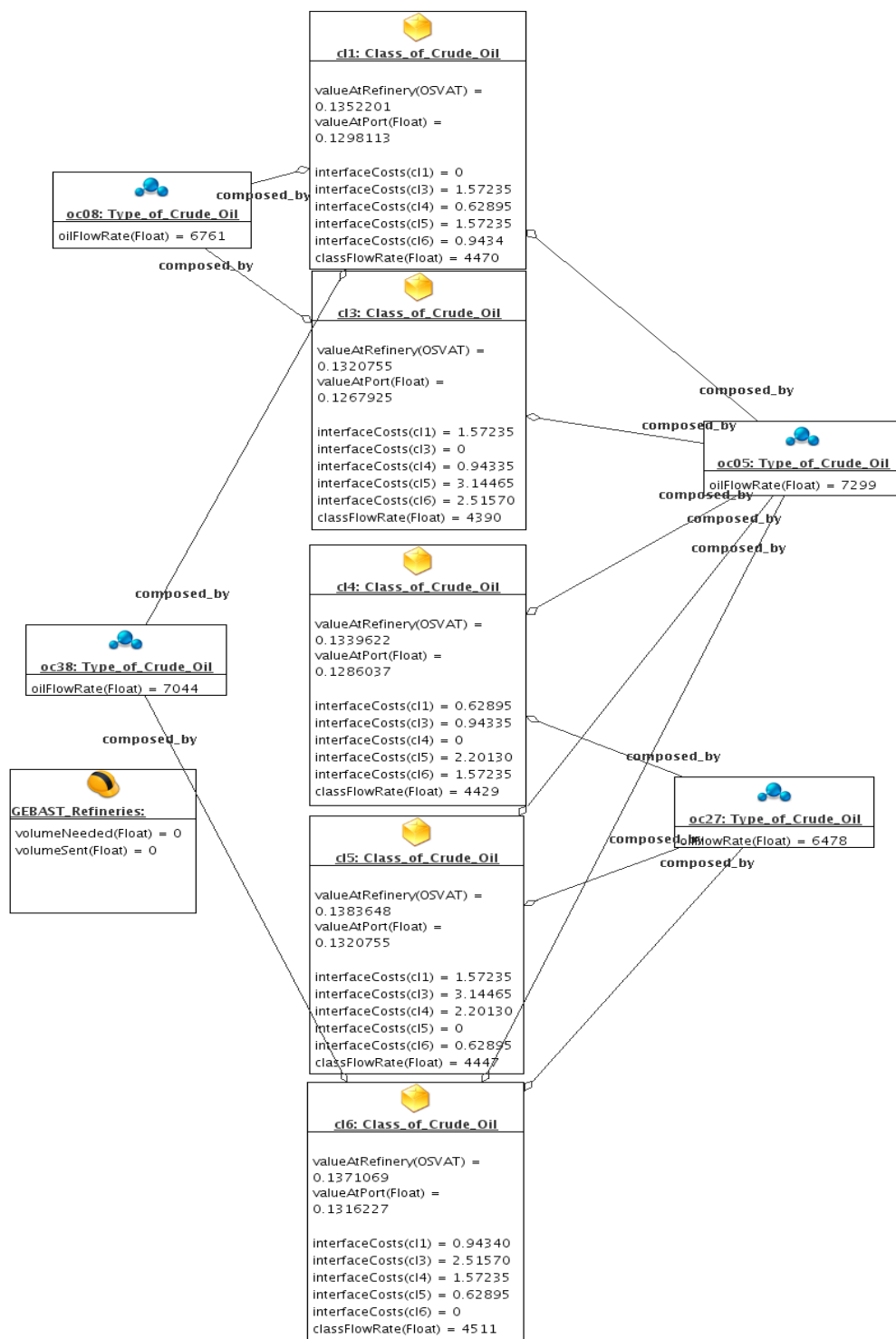


Figura 5.3 – *Snapshot* do estado inicial do exemplo 1 (cont.)

5.2.2.1 Resultados do Exemplo 1

O melhor plano, para o exemplo 1, foi obtido na segunda iteração. Os dados de desempenho computacional na obtenção desta solução, bem como o tamanho do plano, em termos de número de ações, encontram-se na Tabela 5.4. Os resultados foram coletados do arquivo de saída do planejador *SGPlan*.

Tabela 5.4 – Dados das soluções do Exemplo 1

	1ª Iteração	2ª Iteração
Tempo	0,17s	233,5s
Tempo de leitura	0,11s	0,11s
Número de ações	27	24
Valor da Métrica	30,636	26,427

Nota-se que a solução obtida na segunda iteração possui uma qualidade melhor que a primeira solução obtida pelo planejador. O tamanho do plano foi reduzido (de 27 para 24 ações) e a ,métrica final foi reduzida em 14%. Isto indica uma melhora tanto no tempo total utilizado no plano como na eficiência (redução de custo e aumento de receitas).

No entanto, observamos um aumento de mais de 100 vezes no tempo necessário para encontrar uma solução melhor. Isto se deve em grande parte ao processo iterativo utilizado, pois requer uma grande dose de replanejamento. Assim, caso o planejador possuísse um algoritmo de otimização interno, não haveria necessidade de replanejamento e o tempo requerido seria dramaticamente reduzido.

O plano gerado pelo planejador na segunda iteração esta ilustrado na Figura 5.5 abaixo.

```

0.001: (DOCK P2 FRONT_BREA GEBAST) [2.0000]
0.002: (CONNECTTANKPIPELINE TQ3241 OSVAT GEBAST) [0.0000]
0.003: (PUMPTANKPIPELINE TQ3241 OSVAT CL5 CL1) [13.0771]
0.004: (DOCK P1 PEDREIRAS GEBAST) [2.0000]
2.005: (CONNECTTANKERTANK PEDREIRAS TQ3239 GEBAST) [0.0000]
2.006: (PUMPOILTANKERTANKFULL PEDREIRAS TQ3239 OC27 CL4) [3.0874]
5.094: (DISCONNECTTANKERTANK PEDREIRAS TQ3239) [0.0000]
5.095: (CONNECTTANKERTANK PEDREIRAS TQ3237 GEBAST) [0.0000]
5.096: (PUMPOILTANKERTANKFULL PEDREIRAS TQ3237 OC08 CL3) [5.6205]
10.718: (DISCONNECTTANKERTANK PEDREIRAS TQ3237) [0.0000]
10.719: (UNDOCK P1 PEDREIRAS GEBAST) [2.0000]
12.720: (DOCK P1 REBOUCAS GEBAST) [2.0000]
13.090: (DISCONNECTTANKPIPELINE TQ3241 OSVAT CL5) [0.0000]
13.091: (CONNECTTANKERTANK FRONT_BREA TQ3241 GEBAST) [0.0000]
13.092: (PUMPOILTANKERTANKPARCIAL FRONT_BREA TQ3241 OC05 CL5) [8.5457]
14.724: (CONNECTTANKERTANK REBOUCAS TQ3243 GEBAST) [0.0000]
14.725: (PUMPOILTANKERTANKFULL REBOUCAS TQ3243 OC38 CL6) [4.2589]
18.985: (DISCONNECTTANKERTANK REBOUCAS TQ3243) [0.0000]
18.986: (UNDOCK P1 REBOUCAS GEBAST) [2.0000]
21.643: (DISCONNECTTANKERTANK FRONT_BREA TQ3241) [0.0000]
21.644: (CONNECTTANKERTANK FRONT_BREA TQ3234 GEBAST) [0.0000]
21.645: (PUMPOILTANKERTANKFULL FRONT_BREA TQ3234 OC05 CL1) [5.9768]
27.623: (DISCONNECTTANKERTANK FRONT_BREA TQ3234) [0.0000]
27.624: (UNDOCK P2 FRONT_BREA GEBAST) [2.0000]

```

Figura 5.5 – Plano gerado na segunda iteração para o exemplo 1

Neste plano, estão ilustradas as ações necessárias para esvaziar o inventário de todos os navios na ordem em que elas acontecem. Os números que aparecem à extrema esquerda na figura correspondem ao instante onde a ação acontece. A seguir, entre parênteses é relatado o nome da ação e, em seguida todos os seus parâmetros, separados por espaços. Por último, os números que aparecem à extrema direita, entre colchetes, correspondem à duração da ação.

A partir do plano disposto na Figura 5.5, atesta-se que a solução é viável e implementável, pois respeitam todas as restrições operacionais do processo de carga e descarga de petróleo especificadas no capítulo dois deste trabalho. O sequenciamento temporal das tarefas também é respeitado, comprovando a viabilidade temporal das respostas obtidas. A partir destes resultados, conclui-se que a formulação é capaz de gerar respostas viáveis.

O sistema procurou, primeiramente, alimentar o oleoduto com tanques que possuísem maior inventário inicial, para tornar-los disponíveis para tarefas de carregamento, realizadas no píer P-1. Os gráficos nas Figura 5.6 e 5.7 mostram a

evolução dos inventários dos tanques e dos navios respectivamente. Observa-se que os últimos foram completamente esvaziados durante o plano, cumprindo a missão de transferir rapidamente os estoques de óleo cru recebidos.

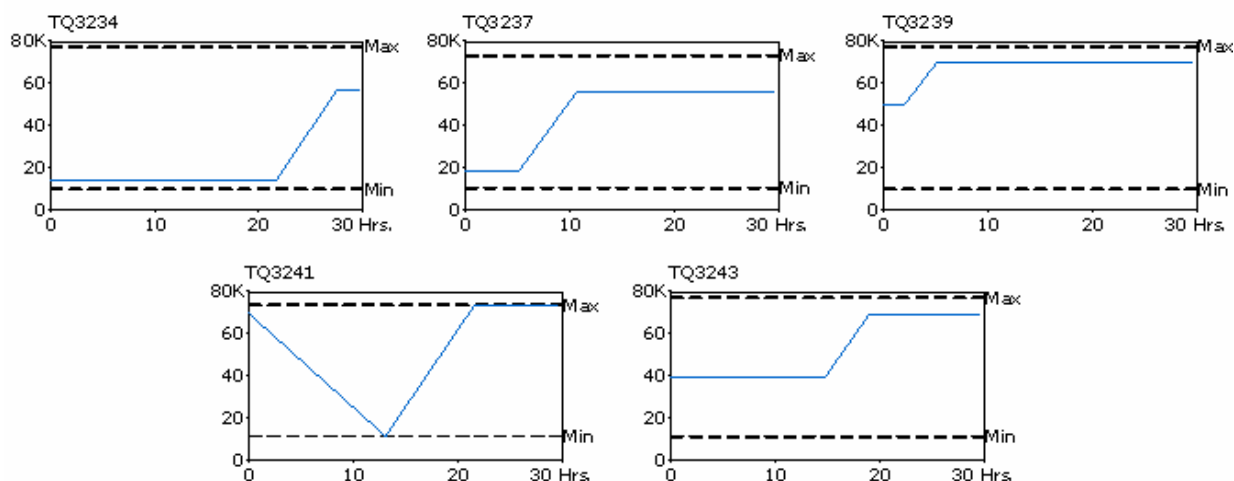


Figura 5.6 – Inventario dos tanques, em m^3 , para a solução do exemplo 1

Pelos gráficos da figura 5.6, pode-se observar que o sistema utiliza praticamente a total capacidade do tanque TQ3241, enquanto não aproveita os outros tanques em sua totalidade. Este comportamento é esperado coerente com o esperado, pois esses tanque estoca uma classe de óleo cru mais nobres, que traz maior ganho na métrica utilizada.

Os gráficos da figura 5.7 ilustram a dinâmica do inventário dos navios descarregados no exemplo 1. Nos casos onde o navio possui mais de um óleo cru, o tipo de óleo é identificado junto à curva correspondente. Como mencionado anteriormente, o plano também indica quando o navio deve chegar ao porto. Isto pode ser visto no gráfico no momento em que o inventário do navio deixa de ser igual a zero.

Apesar de o descarregamento ter sido realizado de maneira eficiente em geral, podemos observar que o navio *Front Brea* foi atracado muito antes de seu descarregamento ser iniciado. Esta é uma ineficiência que adiciona custos desnecessários de utilização dos píeres, decorrente da impossibilidade de otimização de variáveis relacionadas ao tempo verificada no planejador utilizado.

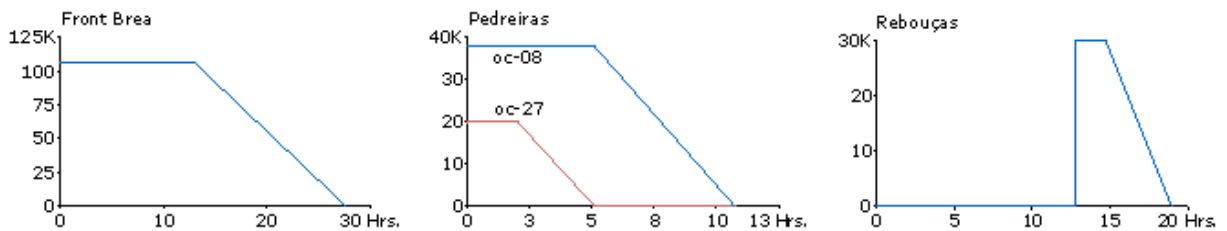


Figura 5.7 – Inventário dos navios, em m^3 , para a solução do exemplo 1

Um aspecto importante a ser analisado no modelo é a sua capacidade de gerar respostas que abasteçam as refinarias eficientemente. Para o Exemplo 1, este objetivo foi atingido, como pode ser observado na Figura 5.8. No gráfico 5.8(a), a evolução do inventário da refinaria abastecida pelo oleoduto OSVAT mostra que as variações não comprometem o estoque de petróleo da refinaria, pois a quantidade de óleo cru estocada permanece em um patamar praticamente constante. Já o gráfico 5.8(b) mostra a dinâmica do inventário em uma escala ampliada, mostrando a tendência de aumento de estoque quando a refinaria é alimentada pelo oleoduto e de diminuição quando não há reposição de estoque.

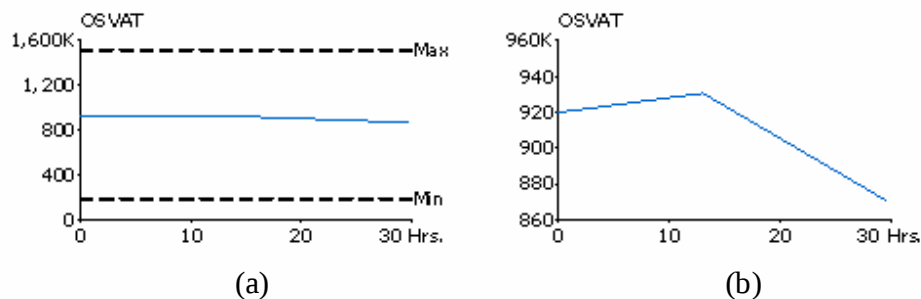


Figura 5.7 – Inventário das refinarias, em m^3 , para a solução do exemplo 1

Como mencionado anteriormente, o modelo não levou em consideração os tempos de utilização do píeres e de sobreestadia. No entanto, a figura 5.8 mostra que o descarregamento dos navios foi bastante eficiente e que, em nenhum caso, um navio ficou mais tempo do que o permitido nos píeres do porto. Ou seja, não houve custos de sobreestadia na solução proposta. Isto mostra que a métrica utilizada para redução do

tempo total (*totalNumberOfTimeActions*) é bastante eficiente em gerir o tempo total utilizado e que o planejador automaticamente busca soluções que façam uso eficiente do tempo, apesar disto não ter sido colocado no modelo. Logo, mesmo com as limitações de software, o modelo gerou, para o exemplo 1, soluções eficientes para a operação do porto.

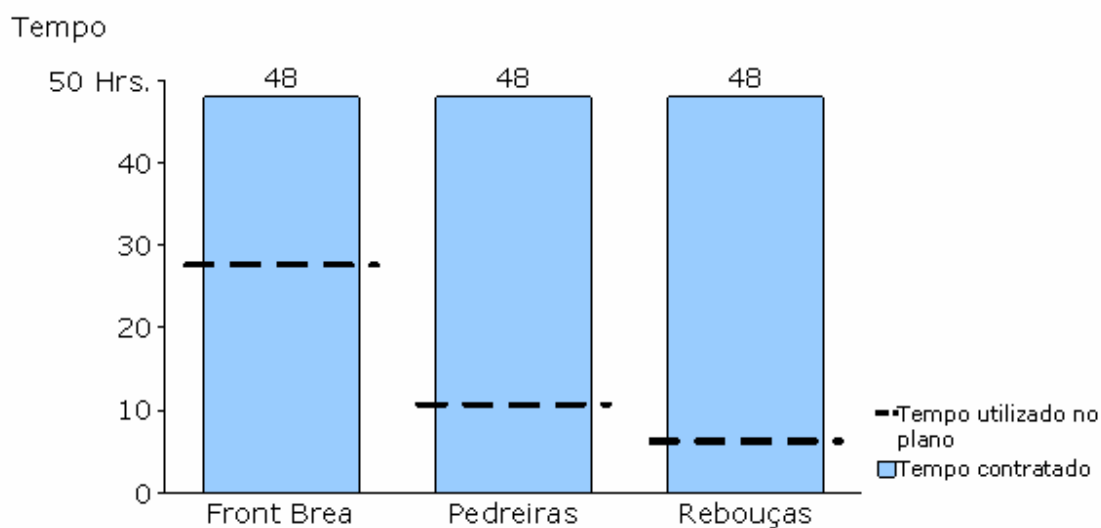


Figura 5.8 –Tempo que cada navio permaneceu atracado nos píeres na solução do exemplo 1 vs. Tempo permitido

5.2.3. Exemplo 2

Este exemplo possui um grau maior de complexidade em relação ao exemplo 1, pois envolve um porto que recebe sete navios, os quais devem ser descarregados em dez tanques e atracados em quatro píeres disponíveis. Os tanques também devem respeitar os procedimentos de decantação, de forma a abastecer a refinaria conectada ao porto por meio de um oleoduto. O exemplo 2 é ilustrado na figura 5.9.

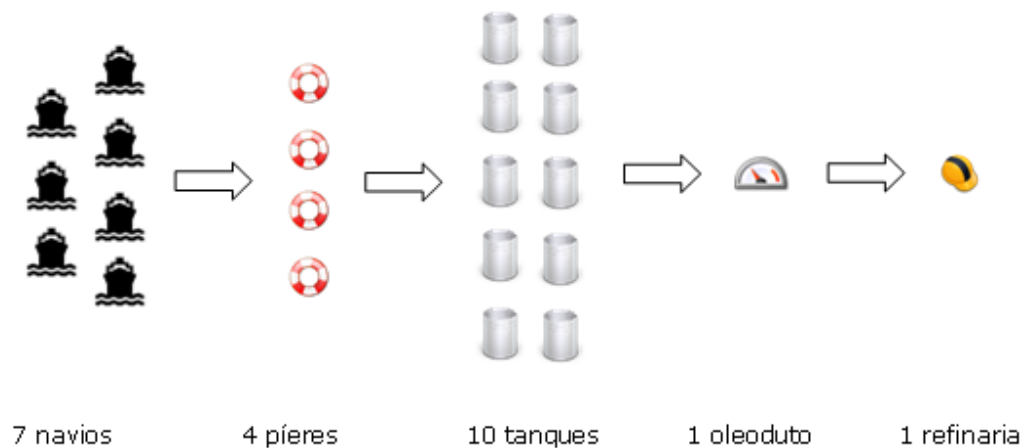


Figura 5.9 – Ilustração do exemplo 2

5.2.3.1. Estado inicial e Dados de Entrada

O estado inicial e, por tanto os dados de entrada, estão dispostos nas Figura 5.10 e 5.11. O processo de carga e descarga dos tanques é idêntico ao do exemplo 1, com a ressalva de que há, neste exemplo, 10 tanques disponíveis para o processo. Da mesma forma, há um maior número de píeres disponíveis e, portanto, maior flexibilidade e capacidade do sistema em descarregar os navios petroleiros considerados. Neste exemplo, cada uma das cinco classes de óleo cru consideradas pode ser armazenada em dois tanques no sistema. Todos os tanques devem respeitar um período de 24 horas para executar o procedimento de decantação após serem abastecidos. O terminal abastece uma refinaria através de um oleoduto, que possui total conectividade com os tanques do sistema, podendo, contudo, receber petróleo de apenas um tanque em cada instante de tempo.

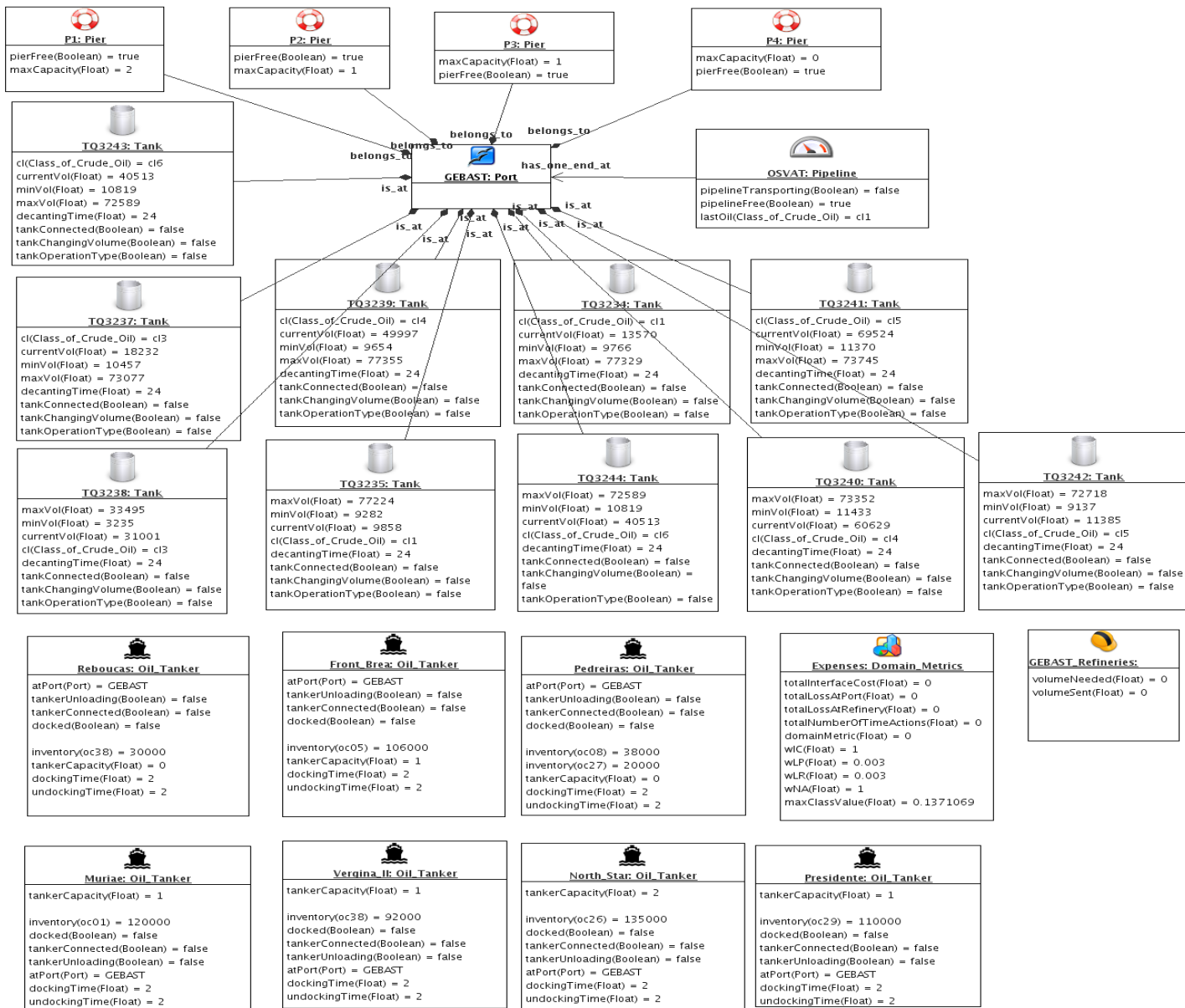


Figura 5.10 – Snapshot do estado inicial do exemplo 1

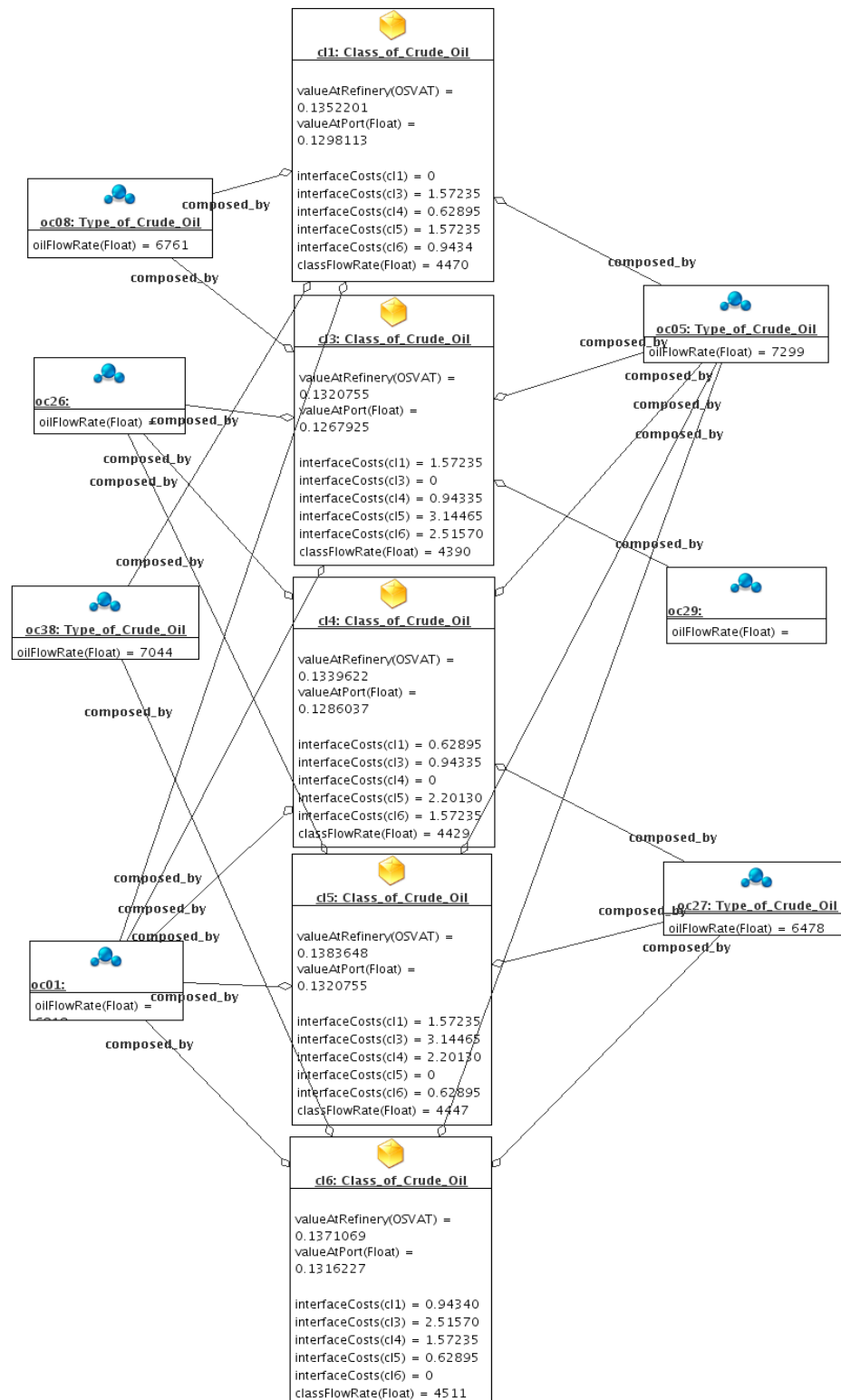


Figura 5.11 – Snapshot do estado inicial do exemplo 1 (cont.)

5.2.3.1. Resultados do exemplo 2

O melhor plano, para o exemplo 2, foi obtido na primeira iteração. Os dados de desempenho computacional na obtenção desta solução encontram-se na Tabela 5.5. Os resultados foram coletados do arquivo de saída do planejador *SGPlan*.

Tabela 5.5 – Dados das soluções do Exemplo 2

	1ª iteração
Tempo	210,09
Tempo de leitura	3,57
Número de ações	88
Valor da Métrica	73,789

Apesar do tempo reduzido para encontrar o primeiro plano, a segunda iteração (que não encontrou solução e, por isso, confirmou a validade da primeira como melhor plano) demorou diversas horas para ser completada. O plano gerado pelo planejador na primeira iteração está ilustrado na Figura 5.12 e 5.13 abaixo.

0.001: (CONNECTTANKPIPELINE TQ3240 OSVAT GEBAST) [0.0000]
 0.002: (DOCK P1 NORTH_STAR GEBAST) [2.0000]
 0.003: (DOCK P2 PRESIDENTE GEBAST) [2.0000]
 0.004: (PUMPTANKPIPELINE TQ3240 OSVAT CL4 CL1) [11.1077]
 2.005: (CONNECTTANKERTANK NORTH_STAR TQ3242 GEBAST) [0.0000]
 2.006: (CONNECTTANKERTANK PRESIDENTE TQ3237 GEBAST) [0.0000]
 2.007: (PUMPOILTANKERTANKPARCIAL NORTH_STAR TQ3242 OC26 CL5) [9.5833]
 2.008: (PUMPOILTANKERTANKPARCIAL PRESIDENTE TQ3237 OC29 CL3) [8.4663]
 10.475: (DISCONNECTTANKERTANK PRESIDENTE TQ3237) [0.0000]
 10.476: (WAITFORDECANTATION TQ3237) [24.0000]
 11.119: (DISCONNECTTANKPIPELINE TQ3240 OSVAT CL4) [0.0000]
 11.120: (CONNECTTANKPIPELINE TQ3238 OSVAT GEBAST) [0.0000]
 11.121: (PUMPTANKPIPELINE TQ3238 OSVAT CL3 CL4) [6.3248]
 11.597: (DISCONNECTTANKERTANK NORTH_STAR TQ3242) [0.0000]
 11.598: (CONNECTTANKERTANK NORTH_STAR TQ3239 GEBAST) [0.0000]
 11.599: (PUMPOILTANKERTANKPARCIAL NORTH_STAR TQ3239 OC26 CL4) [4.2747]
 15.875: (DISCONNECTTANKERTANK NORTH_STAR TQ3239) [0.0000]
 15.876: (CONNECTTANKERTANK NORTH_STAR TQ3240 GEBAST) [0.0000]
 15.877: (PUMPOILTANKERTANKFULL NORTH_STAR TQ3240 OC26 CL4) [7.2358]
 17.453: (DISCONNECTTANKPIPELINE TQ3238 OSVAT CL3) [0.0000]
 17.454: (CONNECTTANKERTANK PRESIDENTE TQ3238 GEBAST) [0.0000]
 17.455: (PUMPOILTANKERTANKPARCIAL PRESIDENTE TQ3238 OC29 CL3) [4.6712]
 22.127: (DISCONNECTTANKERTANK PRESIDENTE TQ3238) [0.0000]
 23.118: (DISCONNECTTANKERTANK NORTH_STAR TQ3240) [0.0000]
 23.119: (UNDOCK P1 NORTH_STAR GEBAST) [2.0000]
 25.120: (DOCK P1 MURIAE GEBAST) [2.0000]
 27.121: (CONNECTTANKERTANK MURIAE TQ3235 GEBAST) [0.0000]
 27.122: (PUMPOILTANKERTANKPARCIAL MURIAE TQ3235 OC01 CL1) [9.8806]
 34.495: (CONNECTTANKPIPELINE TQ3237 OSVAT GEBAST) [0.0000]
 34.496: (PUMPTANKPIPELINE TQ3237 OSVAT CL3 CL3) [14.2642]
 37.005: (DISCONNECTTANKERTANK MURIAE TQ3235) [0.0000]
 48.763: (DISCONNECTTANKPIPELINE TQ3237 OSVAT CL3) [0.0000]
 48.764: (CONNECTTANKERTANK PRESIDENTE TQ3237 GEBAST) [0.0000]
 48.765: (PUMPOILTANKERTANKFULL PRESIDENTE TQ3237 OC29 CL3) [3.8430]
 48.766: (CONNECTTANKPIPELINE TQ3241 OSVAT GEBAST) [0.0000]
 48.767: (PUMPTANKPIPELINE TQ3241 OSVAT CL5 CL3) [13.0771]
 52.611: (DISCONNECTTANKERTANK PRESIDENTE TQ3237) [0.0000]
 52.612: (UNDOCK P2 PRESIDENTE GEBAST) [2.0000]
 52.613: (CONNECTTANKERDECANTINGTANK MURIAE TQ3237 GEBAST) [0.0000]
 52.614: (PUMPOILTANKERTANKPARCIAL MURIAE TQ3237 OC01 CL3) [5.5331]
 54.615: (DOCK P2 VERGINA_II GEBAST) [2.0000]
 56.616: (CONNECTTANKERTANK VERGINA_II TQ3234 GEBAST) [0.0000]
 56.617: (PUMPOILTANKERTANKPARCIAL VERGINA_II TQ3234 OC38 CL1) [9.0515]
 58.151: (DISCONNECTTANKERTANK MURIAE TQ3237) [0.0000]
 58.152: (CONNECTTANKERDECANTINGTANK MURIAE TQ3240 GEBAST) [0.0000]
 58.153: (PUMPOILTANKERTANKFULL MURIAE TQ3240 OC01 CL4) [2.1867]
 60.340: (DISCONNECTTANKERTANK MURIAE TQ3240) [0.0000]
 60.341: (UNDOCK P1 MURIAE GEBAST) [2.0000]

Figura 5.12 – Plano gerado na segunda iteração para o exemplo 2

61.857: (DISCONNECTTANKPIPELINE TQ3241 OSVAT CL5) [0.0000]
 62.343: (DOCK P1 PEDREIRAS GEBAST) [2.0000]
 64.344: (CONNECTTANKERTANK PEDREIRAS TQ3241 GEBAST) [0.0000]
 64.345: (PUMPOILTANKERTANKFULL PEDREIRAS TQ3241 OC27 CL5) [3.0874]
 65.678: (DISCONNECTTANKERTANK VERGINA_II TQ3234) [0.0000]
 65.679: (CONNECTTANKERTANK VERGINA_II TQ3243 GEBAST) [0.0000]
 65.680: (PUMPOILTANKERTANKFULL VERGINA_II TQ3243 OC38 CL6) [4.0092]
 65.681: (WAITFORDECANTATION TQ3234) [24.0000]
 67.438: (DISCONNECTTANKERTANK PEDREIRAS TQ3241) [0.0000]
 69.692: (DISCONNECTTANKERTANK VERGINA_II TQ3243) [0.0000]
 69.693: (UNDOCK P2 VERGINA_II GEBAST) [2.0000]
 71.694: (DOCK P2 REBOUCAS GEBAST) [2.0000]
 89.686: (CONNECTTANKPIPELINE TQ3234 OSVAT GEBAST) [0.0000]
 89.687: (PUMPTANKPIPELINE TQ3234 OSVAT CL1 CL5) [15.1148]
 104.803: (DISCONNECTTANKPIPELINE TQ3234 OSVAT CL1) [0.0000]
 104.804: (CONNECTTANKERTANK PEDREIRAS TQ3234 GEBAST) [0.0000]
 104.805: (PUMPOILTANKERTANKFULL PEDREIRAS TQ3234 OC08 CL1) [5.6205]
 104.806: (CONNECTTANKPIPELINE TQ3244 OSVAT GEBAST) [0.0000]
 104.807: (PUMPTANKPIPELINE TQ3244 OSVAT CL6 CL1) [6.5826]
 110.428: (DISCONNECTTANKERTANK PEDREIRAS TQ3234) [0.0000]
 110.429: (UNDOCK P1 PEDREIRAS GEBAST) [2.0000]
 110.430: (CONNECTTANKERDECANTINGTANK REBOUCAS TQ3234 GEBAST) [0.0000]
 110.431: (PUMPOILTANKERTANKPARCIAL REBOUCAS TQ3234 OC38 CL1) [4.1969]
 111.394: (DISCONNECTTANKPIPELINE TQ3244 OSVAT CL6) [0.0000]
 112.433: (DOCK P1 FRONT_BREA GEBAST) [2.0000]
 114.434: (CONNECTTANKERDECANTINGTANK FRONT_BREA TQ3241 GEBAST) [0.0000]
 114.435: (PUMPOILTANKERTANKPARCIAL FRONT_BREA TQ3241 OC05 CL5) [5.8056]
 114.633: (DISCONNECTTANKERTANK REBOUCAS TQ3234) [0.0000]
 114.634: (CONNECTTANKERDECANTINGTANK REBOUCAS TQ3243 GEBAST) [0.0000]
 114.635: (PUMPOILTANKERTANKFULL REBOUCAS TQ3243 OC38 CL6) [0.0620]
 114.698: (DISCONNECTTANKERTANK REBOUCAS TQ3243) [0.0000]
 114.699: (UNDOCK P2 REBOUCAS GEBAST) [2.0000]
 120.247: (DISCONNECTTANKERTANK FRONT_BREA TQ3241) [0.0000]
 120.248: (CONNECTTANKERTANK FRONT_BREA TQ3244 GEBAST) [0.0000]
 120.249: (PUMPOILTANKERTANKPARCIAL FRONT_BREA TQ3244 OC05 CL6) [8.4628]
 128.713: (DISCONNECTTANKERTANK FRONT_BREA TQ3244) [0.0000]
 128.714: (CONNECTTANKERDECANTINGTANK FRONT_BREA TQ3243 GEBAST) [0.0000]
 128.715: (PUMPOILTANKERTANKFULL FRONT_BREA TQ3243 OC05 CL6) [0.2541]
 128.970: (DISCONNECTTANKERTANK FRONT_BREA TQ3243) [0.0000]
 128.971: (UNDOCK P1 FRONT_BREA GEBAST) [2.0000]

Figura 5.13 – Plano gerado na segunda iteração para o exemplo 2 (cont.)

Os gráficos de evolução dos inventários dos tanques do sistema estão ilustrados na figura 5.14. Nesta figura, os eixos das abscissas baseiam-se em unidades de hora, e das ordenadas em m³.

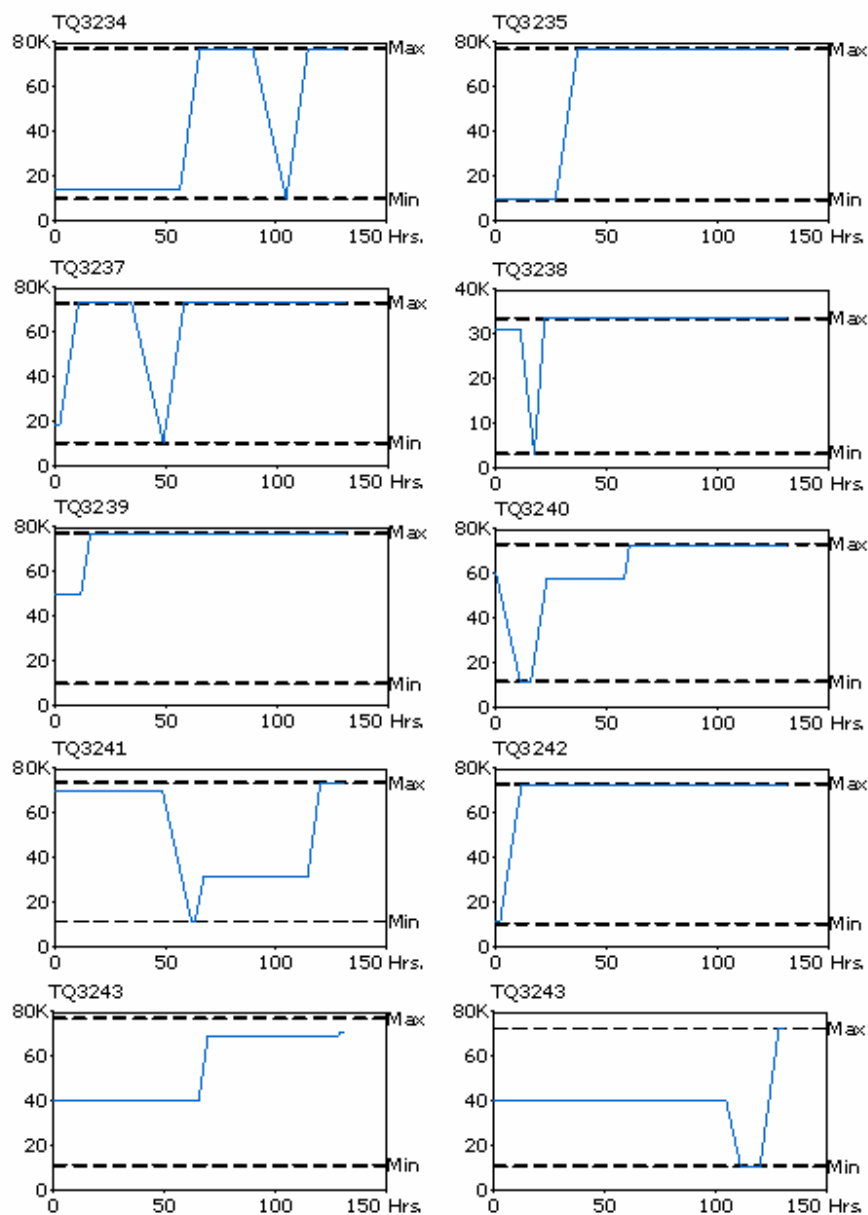


Figura 5.14 – Inventário dos tanques, em m^3 , para a solução do exemplo 2

Pode-se notar pelas figuras 5.27 e 5.28 que os tanques do sistema respeitam o tempo mínimo de decantação antes de qualquer tarefa de descarga. Neste exemplo, percebe-se que todos os tanques são utilizados em sua capacidade máxima. Isto mostra uma limitação no modelo, pois, como as operações de carga e descarga de navios e tanques devem ser totais, existe pouco espaço para maximizar o valor do óleo quando temos muitos navios e devemos minimizar o número de ações que consomem tempo.

Mais uma vez, fica clara a necessidade de um planejador capaz de lidar com as complexidades deste problema.

A evolução do inventário dos navios do sistema está ilustrada nos gráficos da figura 5.29.

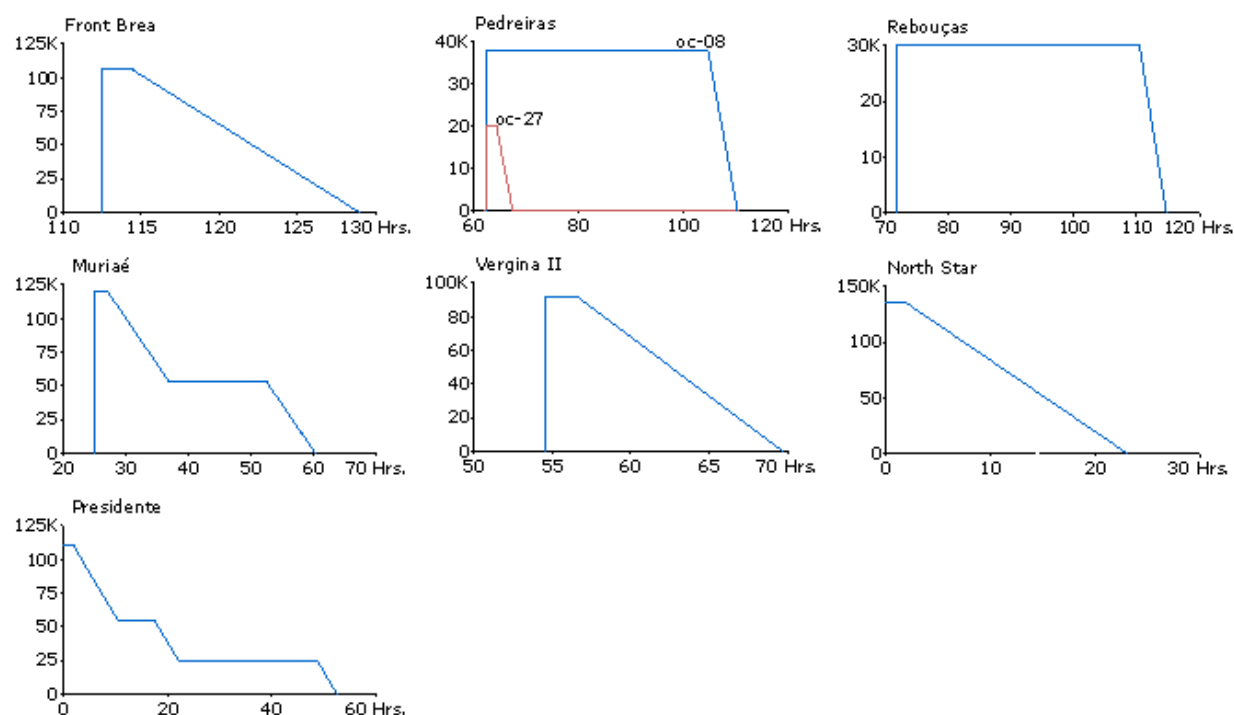


Figura 5.15 – Inventário dos navios, em m^3 , para a solução do exemplo 2

Através da figura 5.15, é possível observar que todos os navios, à exceção do *Rebouças*, são descarregados de forma ótima: no tempo mínimo necessário. Nenhum outro navio foi atracado sem que seu descarregamento fosse iniciado logo em seguida. Além disso, a grande maioria dos navios foi descarregada de forma contínua, tornando o planejamento muito eficiente.

A figura 5.16(a) mostra que, para o problema mais complexo, o nível de estoque nas refinarias não foi tão bem administrado como no exemplo 1. Nota-se que as variações do volume são muito mais claras neste caso, o que é esperado devido ao horizonte maior de planejamento inerente a este problema. A Figura 5.16(b) expõe com mais clareza como a queda nos estoques é muito mais agressiva neste exemplo, apesar

de um volume muito maior de óleo cru ter sido transferido para as refinarias. No entanto, conclui-se que a queda de estoques não compromete a qualidade do cronograma de atividades obtido, já que o estoque se mantém sempre muito acima dos níveis mínimos.

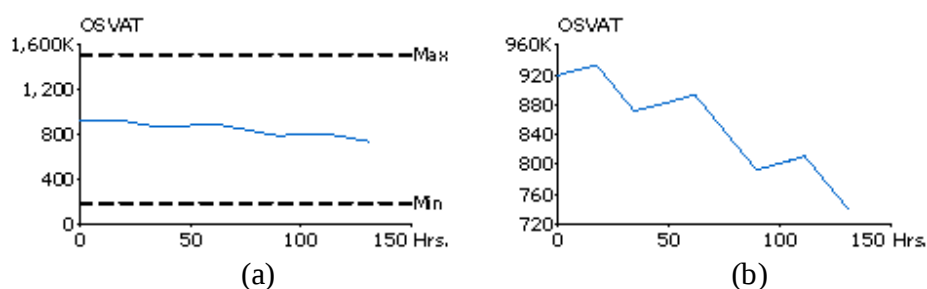


Figura 5.16 – Inventário das refinarias, em m^3 , para a solução do exemplo 2

Assim como no exemplo 1, Figura 5.17 mostra que o descarregamento dos navios foi bastante eficiente e que apenas um navio permaneceu atracado mais tempo que o permitido, reafirmando que a métrica utilizada e o planejador são capazes de gerar planos eficientes em relação ao tempo.

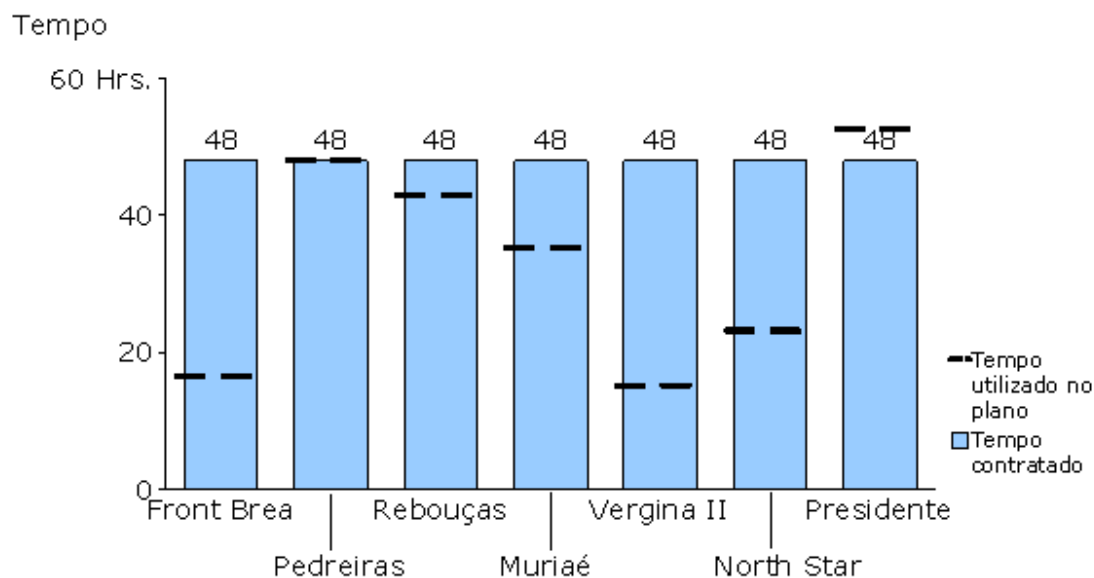


Figura 5.17 – Tempo que cada navio permaneceu atracado nos píeres na solução do exemplo 2 vs. Tempo permitido

6. CONCLUSÃO E PROXIMOS PASSOS

Neste trabalho, um problema de planejamento real e complexo, como o planejamento da programação das atividades de um terminal de recebimento e distribuição de petróleo foi apresentado e modelado utilizando técnicas de planejamento automático e a linguagem UML.P como ferramenta de modelagem. A tradução para PDDL (linguagem entendida pelos planejadores) foi realizada automaticamente pela plataforma itSIMPLE, que implementa o ciclo de design em planejamento utilizando a UML.P.

A utilização da UML.P e técnicas de Planejamento Automático para a modelagem deste problema implicam em vantagens consideráveis em relação aos métodos de programação matemática utilizados até hoje. O modelo desenvolvido possui um alto grau de flexibilidade e pode ser alterado com facilidade para se adequar a eventuais mudanças no sistema. Além disso, a modelagem em si não considera simplificações matemáticas (como linearizações) que comprometem o modelo. As hipóteses simplificadoras utilizadas neste trabalho estão relacionadas apenas às limitações encontradas nos planejadores disponíveis. Outro ponto importante é que a utilização da UML.P permite um processo de modelagem mais simples e visual do que os modelos matemáticos baseados em equações utilizados até hoje. Além disso, a utilização de uma metodologia de design *top down*, que começa nos *Diagramas de Casos de Uso*, permite e facilita a inclusão de pontos de vistas de diversas pessoas e não apenas de *experts* em planejamento.

Além disso, foram apresentados os resultados dos testes do modelo seqüencial (sem tempo) realizados com planejadores gerais. Estes experimentos buscavam validar o modelo proposto e investigar a habilidade dos planejadores modernos para resolver estes problemas. Por este ser um domínio complexo e grande, os processos de requisitos e especificação foram realizados utilizando a ferramenta de KE (Knowledge Engineering), itSIMPLE. Também foram apresentados e analisados dois exemplos de problemas considerando o modelo completo (com tempo). Apesar de mais simples estes cenários se baseavam em dados reais e se aproximavam muito bem de situações enfrentadas diariamente no terminal.

Os resultados obtidos com estes exemplos demonstraram a aplicabilidade de técnicas de Planejamento Automático para o problema de suprimento de petróleo em complexos de distribuição. As soluções obtidas mostraram um grau satisfatório de eficiência, mesmo quando a escala dos problemas proposto aumentava, aproximando-se de uma situação real.

Devido às propriedades do domínio, poucos planejadores foram capazes de lidar com todas as características envolvidas. Por esta razão somente três planejadores foram estudados: *Metric-FF*, *SGPlan* e *MILP-XXL*. Em geral, estes planejadores não mostraram um bom desempenho quando tiveram que resolver problemas realistas no domínio estudado. Os resultados não foram satisfatórios em relação ao tempo de resposta e/ou à qualidade do plano. As respostas só foram boas quando os problemas eram relativamente simples, pequenos e não realistas. Aparentemente, domínios que utilizam muitas variáveis numéricas podem causar um grande impacto no processo de planejamento se comparados a domínios puramente lógicos. Além disso, quando o tempo foi incluído ao modelo, apenas o *SGPlan* foi capaz de gerar planos viáveis para os problemas propostos.

Os testes realizados com estes planejadores identificaram limitações que, se eliminadas, podem aumentar a qualidade tanto os resultados como a eficiência computacional. Verificou-se que os planejadores ainda tratam o tempo de uma maneira muito simples, aceitando apenas ações cuja duração é pré-determinada. No entanto, para que o modelo encontre as melhores soluções torna-se necessário que o tempo de uma ação seja determinado pelo planejador. Deve-se mencionar que a linguagem PDDL já permite um tratamento completo do tempo, incluindo tempo contínuo. Por tanto, a tecnologia de planejamento disponível ainda está muito defasada da capacidade de expressão dessa linguagem.

O único planejador capaz de lidar com o problema, não realizava qualquer tipo de otimização. Assim, tornou-se impossível incluir variáveis dependentes do tempo (como custo de estadia e sobreestadia) na métrica utilizada para avaliar a qualidade do plano. Outras variáveis puderam ser otimizadas através de iterações que, apesar de conseguirem encontrar bons planos, aumentaram significativamente o tempo

computacional devido à necessidade de replanejamento. Um planejador que realizasse otimização automaticamente apresentaria resultados muito mais eficientes.

Assim, devido às hipóteses simplificadoras necessárias para que os planejadores gerais fossem capazes de lidar com o problema e à sua baixa eficiência, a solução de um problema real ainda não é prática. Logo, a partir deste trabalho, surge a necessidade de alterar os planejadores gerais disponíveis atualmente (ou desenvolver novos planejadores) para que possam resolver este problema com mais eficiência. Em outras palavras, para solução deste problema (como a maioria dos problemas reais) planejadores específicos, e não gerais são necessários.

Finalmente, este trabalho deixa clara a aplicabilidade das técnicas de Planejamento Automático na resolução de problemas de grande escala. Todas as soluções apresentadas se mostraram viáveis e obtiveram um grau satisfatório de eficiência. No entanto, observam-se algumas limitações relacionadas aos planejadores gerais disponíveis atualmente que impedem a obtenção de soluções realmente ótimas. Portanto, para ir um passo além na resolução deste problema é necessário desenvolver um planejador dedicado e com heurísticas específicas para este domínio.

7. BIBLIOGRAFIA

BACCHUS F. and KABANZA F. *Using Temporal Logics to Express Search Control Knowledge for Planning*. Artificial Intelligence, vol 116, 2000.

BENTON, J., SUBBARAO, K. and DO M. *YochanPS: PDDL3 Simple Preferences as Partial Satisfaction Planning*. In The Fifth International Planning Competition Booklet, ICAPS 2006.

D'SOUZA, F.D. and WILLS, A.C. *Object, Components and Frameworks with UML – The Catalysis Approach*. Addison-Wesley. United States of America and Canada, 1999.

DAHAL, K., GALLOWAY, S., BURT, G., MCDONALD, J., and HOPKINGS, I. 2003. *Port system simulation facility with an optimization capability*, International Journal of Computational Intelligence and Applications, vol. 3, no. 4, pp. 395-410.

EDELKAMP, S., JABBAR, S., and NAIZIH, M. *Large-scale optimal PDDL3 planning with MIPS-XXL*. In 5th International Planning Competition Booklet (IPC-2006), pp. 28– 30, Lake District, England, July 2006.

FOX, M., and LONG, D. *PDDL2.1: An extension of pddl for expressing temporal planning domains*. Journal of Artificial Intelligence Research (JAIR) 20:61–124, 2003.

GHALLAB M., NAU D. and TRAVERSO P. *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.

GEREVINI, A., SAETTI, A. and SERINA, I. *LPG-TD: a Fully Automated Planner for PDDL2.2 Domains*. International Planning Competition, 14th Int. Conference on Automated Planning and Scheduling (ICAPS-04) , Whistler, Canada, 2004.

HASLUM, P. *TP4 '04 and HSP*a*. International Planning Competition, 14th Int. Conference on Automated Planning and Scheduling (ICAPS-04), booklet of the system demo section, Whistler, Canada, 2004.

HOFFMANN, J. *The Metric-FF planning system: Translating "ignoring delete lists" to numeric state variables*. Journal of Artificial Intelligence Research. Accepted for special issue on the 3rd International Planning Competition, 2003.

HOFFMANN, J., Edelkamp, S., Englert, R., Liporace, F. S. Thiébaux, and S. Trüg. *Towards Realistic Benchmarks for Planning: the Domains Used in the Classical Part of IPC-4*. Booklet of the Fourth International Planning Competition, pages 7-14, June 2004.

HOFFMAN, J, NEBEL, B. *The FF Planning System: Fast Plan Generation Through Heuristic Search*. Journal of Artificial Intelligence Research, Volume 14, 2001, pages 253 – 302

HSU, C.W., WAH, B.W., HUANG, R., and CHEN, Y. X. 2006. *New Features in SGPlan for Handling Soft Constraints and Goal Preferences in PDDL3.0*. Proc. Fifth International Planning Competition, International Conf. on Automated Planning and Scheduling ICAPS'06, June 2006.

JIA, Z. and IERAPETRITOU, M. *Refinery Short-Term Scheduling Using Continuous Time Formulation: Crude-Oil Operations*. Ind. Eng. Chem. Res., vol. 42, pp. 3085-3097, 2003.

LI, J., WENKAI L., KARIMI, I.A., and SRINIVASAN, R. 2005. *Robust and Efficient Algorithm for Optimizing Crude Oil Operations*. American Institute of Chemical Engineers Annual Meeting.

MÁS, R. and PINTO, J. *A Mixed-Integer Optimization Strategy for Oil Supply in Distribution Complexes*. Optimization and Engineering, vol. 4, no. 1-2, pp. 23-64, Jun. 2003.

MCDERMOTT, D. *The PDDL Planning Domain Definition Language*. The AIPS-98 Planning Competition Committee, 1998.

MORO L. and PINTO J. *Mixed-Integer Programming Approach for Short-Term Crude Oil Scheduling*. Ind. Eng. Chem. Res., vol. 43, pp. 85-94, Dez. 2003.

NAU, D., MUNOZ-AVILA H., CAO Y., LOTEM A., and MITCHELL, S. *Total-Order Planning with Partially Ordered Subtasks*. IJCAI-2001. Seattle, August, 2001.

NEIRO S., PINTO J. *A general modeling framework for the operational planning of petroleum supply chains*. Computers and Chemical engineering, vol. 28, pp. 871-896, 2004.

NEIRO, S., PINTO J. *Multiperiod Optimization for Production Planning of Petroleum Refinery*. Chemical Engineering Communications, Estados Unidos, vol. 192, pp. 62-88, 2005.

OMG. 2001. OMG Unified Modeling Language Specification, m Version 1.4.

OMG. 2003. OMG Unified Modeling Language Specification - Object Constraint Language m Version 2.0.

PARKER, E. *Making graphplan goal-directed*. In Proceedings of ECP-99, 333-346, 1999.

ZHENYA J. *Large-scale scheduling optimization considering uncertainty: Refinery operations*. RUTGERS THE STATE UNIVERSITY OF NEW JERSEY - NEW BRUNSWICK, 2005.

ANEXO A

A.1 Atracar no píer

Use Case: *Atracar no píer*

Actors: Navio, Porto

Description: Os navios que chegam ao terminal petrolífero são alocados aos píeres para que seus tanques sejam descarregados.

Constraints:

Pre-conditions

1. O navio deve ter chegado no porto.
2. O navio deve estar desatracado.
3. Um navio só pode atracar em píeres que possuam capacidade de fazê-lo.
4. O píer deve estar livre para receber navios.
5. O instante de chegada do navio deve ser anterior ao das outras embarcações de desejam atracar no mesmo píer.

Post-conditions

1. O navio está atracado.
2. O pier está ocupado.
3. O inventário de óleo cru do navio pode ser descarregado.
4. O custo de estadia do navio passa a ser contabilizado.

Invariants

O navio só pode atracar em um único pier.

Temporal Constraints

Após alocado a um píer, o navio demora um tempo determinado para finalizar esta operação, dependendo do porte do navio. Seu inventário só pode ser descarregado nos tanques do porto a partir desse instante.

Further Details

Additional Information

1. A taxa de estadia que será cobrada de um navio depende unicamente do pier onde ele atracou.
2. O navio possui uma janela de tempo durante a qual pode permanecer atracado pagando as tarifas correspondentes ao seu porte. Caso o petroleiro ultrapasse o período contratado, será cobrada uma taxa de sobreestadia, que depende do seu porte e é diretamente proporcional ao tempo de sobreestadia.

A.2 Desatracar no píer

Use Case: *Desatracar do píer*

Actors: Navio, Porto

Description: Após ter sido descarregado, o navio deixa o píer livre para que outra embarcação possa atracar.

Constraints:

Pre-conditions

1. Todo o inventário de óleo cru do navio destinado ao porto foi descarregado por completo.
2. O navio deve estar atracado.
3. O navio deve estar desconectado.

Post-conditions

1. Navio desatracado.
2. Píer livre.
3. Finaliza a contabilização do custo de estadia.

Invariants

O navio demora um tempo determinado para deixar o porto, dependendo do seu porte.
O píer só pode receber outro navio depois desse instante.

A.3 Receber determinado óleo cru de um navio em um tanque

Use Case: *Receber determinado óleo cru de um navio em um tanque*

Actors: Porto, Navio

Description: Os inventários de óleo cru dos navios petroleiros que chegam ao terminal petrolífero são descarregados em píeres e armazenados em tanques.

Constraints:

Pre-conditions

1. O óleo do navio só pode ser descarregado em tanques cuja classe o contenha.
2. O volume de óleo no tanque deve ser menor que a sua capacidade máxima de armazenamento.

Post-conditions

1. Um determinado volume de óleo cru foi transferido do navio para o tanque do porto.
2. O custo do petroleo retirado dos navios e contabilizado.

Invariants

1. Um navio só pode estar conectado a um tanque em cada operação.
2. Um tanque só pode estar conectado a um navio em cada operação.
3. Um tanque nunca pode ser carregado e descarregado ao mesmo tempo.
4. Cada tanque só pode armazenar uma única classe de óleo cru.

Temporal Constraints

O tempo de duracao do descarregamento de um oleo do navio em um tanque depende do volume a ser descarregado e da vazao maxima de bombeamento.

Further Details

Additional Information

- 1.Cada classe de oleo cru possui um valor quando esta armazenada nos tanques do porto.
2. Cada tipo de oleo cru possui um determinado custo quando e retirado dos navio (custo de materia-prima).
- 3.Ao final do horizonte de planejamento contabiliza-se a "Receita de Oleo Armazenado no Porto" que é dada pela diferença de receita de petróleo entre os instantes final e inicial do horizonte de planejamento considerado.

A.4 Conectar um tanque a um navio

Use Case: *Conectar um tanque a um navio*

Description: A conexão permite que o óleo cru do navio possa ser bombeado para os tanques do porto. O bombeamento nunca pode ocorrer antes que os dois estejam conectados.

Constraints:

Pre-conditions

1. O navio deve estar atracado no pier.
2. O navio não pode estar conectado a um outro tanque.
3. O tanque não pode estar conectado a um outro navio ou a um oleoduto.

Post-conditions

Navio e tanques estão conectados.

A.5 Bombear óleo do navio para o tanque

Use Case: *Bombear óleo do navio para o tanque*

Description: Durante o bombeamento, um determinado tipo de óleo cru do navio é transferido para um tanque do porto. A vazão máxima da bomba depende do tipo de óleo cru sendo descarregado.

Constraints:

Pre-conditions

1. O navio e o tanque devem estar conectados.
2. A vazão de descarga é igual a vazão máxima para o bombeamento daquele tipo de óleo cru.

Post-conditions

1. O volume daquele tipo de óleo cru no navio diminuiu.
2. O volume de petróleo no tanque aumentou.

Invariants

1. O navio e tanque estão conectados.
2. O volume do tanque é sempre menor ou igual à sua capacidade máxima de armazenamento de óleo cru.

A.6 Desconectar um tanque de um navio

Use Case: *Desconectar um tanque de um navio*

Description: A desconexão entre o navio e o tanque ocorre após o término da operação de bombeamento, liberando o navio e o tanque para uma nova operação.

Constraints:

Pre-conditions

1. O navio e o tanque devem estar conectados.
2. A operação de bombeamento foi finalizada.

Post-conditions

1. O navio está desconectado.
2. O tanque está desconectado.
3. O tanque começa a decantar.

A.7 Distribuir óleo cru para a refinaria a partir de um tanque

Use Case: *Distribuir óleo cru para a refinaria a partir de um tanque*

Actors: Porto, Refinaria

Description: O petróleo recebido no terminal petrolífero é distribuído para as refinarias através de oleodutos. Cada refinaria está conectada ao porto através de um único oleoduto.

Constraints:

Pre-conditions

1. O tanque não pode estar decantando.
2. O volume do tanque deve ser maior que seu volume mínimo (lastro).

Post-conditions

1. Um determinado volume de óleo cru foi transportado do tanque para a refinaria.
2. A receita correspondente ao óleo cru enviado para a refinaria é contabilizada.
3. O custo de interface é contabilizado.

Invariants

Cada oleoduto só pode estar conectado a um único tanque e a uma única refinaria em cada operação de transporte.

Temporal Constraints

1. O tempo de duração da distribuição de óleo de um tanque para uma refinaria depende do volume a ser transportado e da vazão máxima de bombeamento.
2. O de óleo cru na refinaria deve ser sempre menor ou igual à sua capacidade máxima de estocagem.

Further Details

Additional Information

1. Os oleodutos podem conter mais de um tipo de óleo cru e, portanto, há sempre um item sendo bombeado e outro saindo durante a operação. Quando duas classes diferentes são bombeadas ao mesmo tempo, o óleo da classe mais nobre sofre uma perda de valor (custo de interface).
2. Os inventários de óleo cru do porto podem não ter o destino totalmente livre. Há tanques que devem armazenar óleo destinados a algum oleoduto. Esta restrição não será considerada nesta fase do modelo, pois não sabemos se isso é uma restrição física do problema.
3. Cada classe de óleo cru possui um valor quando esta armazenada nos tanques da refinaria.

A.8 Conectar um tanque e uma refinaria através de um oleoduto

Use Case: *Conectar um tanque e uma refinaria através de um oleoduto*

Description: A conexão permite que o óleo cru do tanque possa ser bombeado, através do oleoduto, para a refinaria. O bombeamento nunca pode ocorrer antes que os dois estejam conectados.

Constraints:

Pre-conditions

1. Deve existir um oleoduto capaz de levar óleo para a refinaria.
2. O oleoduto não pode estar conectado a outro tanque ou a outra refinaria.
3. O tanque não pode estar conectado a outro oleoduto ou a um navio.

Post-conditions

1. O tanque está conectado ao oleoduto.
2. A refinaria está conectada ao oleoduto.

A.9 Bombear óleo cru através do oleoduto

Use Case: *Bombear óleo cru através do oleoduto*

Description: Durante o bombeamento, óleo cru do tanque é transferido para a refinaria. A vazão máxima da bomba depende da classe do óleo cru sendo transportado.

Constraints:

Pre-conditions

1. O oleoduto deve estar conectado à refinaria.
2. O oleoduto deve estar conectado ao tanque.
3. Vazão de bombeamento no oleoduto igual à vazão máxima da classe de óleo cru mais densa presente no oleoduto.

Post-conditions

1. Volume do tanque diminuiu.
2. Volume de óleo na refinaria aumentou.

Invariants

1. O volume do tanque é sempre maior ou igual ao seu lastro.
2. O oleoduto está conectado ao tanque e a refinaria.

A.10 Desconectar um tanque e uma refinaria de um oleoduto

Use Case: *Desconectar um tanque e uma refinaria de um oleoduto*

Description: A desconexão entre o oleoduto e o tanque e entre o oleoduto e a refinaria ocorre após o término da operação de transporte, liberando o oleoduto, o tanque e a refinaria para uma nova operação.

Constraints:

Pre-conditions

1. O tanque está conectado ao oleoduto.
2. A refinaria está conectada ao oleoduto.
3. A operação de bombeamento foi finalizada.

Post-conditions

1. O tanque está desconectado.
2. A refinaria está desconectada.
3. O oleoduto está livre.

A.11 Consumir óleo cru

Use Case: *Consumir óleo cru*

Actors: Refinaria

Description: As refinarias serão modeladas como consumidoras de óleo cru do sistema, pois o modelo de operação da refinaria não será considerado. Será admitido que, no planejamento de curto prazo de suprimento de óleo cru, as refinarias já tenham preestabelecido o consumo médio de óleo cru no período.

Constraints:

Pre-conditions

O volume de óleo cru na refinaria deve ser maior que o volume mínimo dos tanques de armazenamento.

Post-conditions

Volume de óleo armazenado na refinaria diminui.

Invariants

1. O volume deve ser sempre maior que o volume mínimo permitido no estoque da refinaria.
2. A refinaria consome óleo cru em todos os instantes da operação.

Further Details

Additional Information

Durante toda a operação do sistema de distribuição, a refinaria deve ter óleo suficiente para manter suas operações. Ou seja, o volume de óleo cru nas refinarias deve ser sempre maior que a sua capacidade mínima.

ANEXO B

B.1 Modelo de Porto em PDDL

```
(define (domain Oil_Supply_Domain)
  (:requirements :typing :fluents :quantified-preconditions :durative-
actions)
  (:types
    Oil_Tanker - object
    Port - object
    Tank - object
    Pipeline - object
    Pier - object
    Class_of_Crude_Oil - object
    Type_of_Crude_Oil - object
  )
  (:predicates
    (is_at ?tan - Tank ?por - Port)
    (belongs_to ?pie - Pier ?por - Port)
    (has_one_end_at ?pip - Pipeline ?por - Port)
    (composed_by ?cla - Class_of_Crude_Oil ?typ - Type_of_Crude_Oil)
    (docks_at ?oil - Oil_Tanker ?pie - Pier)
    (is_connected_to ?tan - Tank ?pip - Pipeline)
    (is_linked_to ?oil - Oil_Tanker ?tan - Tank)
    (docked ?oil - Oil_Tanker)
    (tankerConnected ?oil - Oil_Tanker)
    (tankerUnloading ?oil - Oil_Tanker)
    (atPort ?oil - Oil_Tanker ?por - Port)
    (cl ?tan - Tank ?cla - Class_of_Crude_Oil)
    (tankConnected ?tan - Tank)
    (tankChangingVolume ?tan - Tank)
    (tankOperationType ?tan - Tank)
    (lastOil ?pip - Pipeline ?cla - Class_of_Crude_Oil)
    (pipelineFree ?pip - Pipeline)
    (pipelineTransporting ?pip - Pipeline)
    (pierFree ?pie - Pier)
  )
  (:functions
    (tankerCapacity ?oil - Oil_Tanker)
    (inventory ?oil - Oil_Tanker ?oc - Type_of_Crude_Oil)
    (dockingTime ?oil - Oil_Tanker)
    (undockingTime ?oil - Oil_Tanker)
    (maxVol ?tan - Tank)
    (minVol ?tan - Tank)
    (currentVol ?tan - Tank)
    (decantingTime ?tan - Tank)
    (maxCapacity ?pie - Pier)
    (interfaceCosts ?cla - Class_of_Crude_Oil ?cl - Class_of_Crude_Oil)
    (valueAtPort ?cla - Class_of_Crude_Oil)
    (valueAtRefinery ?cla - Class_of_Crude_Oil ?ol - Pipeline)
    (classFlowRate ?cla - Class_of_Crude_Oil)
    (oilFlowRate ?typ - Type_of_Crude_Oil)
    (totalInterfaceCost)
```

```

(totalLossAtPort)
(totalLossAtRefinery)
(totalNumberOfTimeActions)
(domainMetric)
(wIC)
(wLP)
(wLR)
(wNA)
(maxClassValue)
(volumeNeeded)
(volumeSent)
)
(:durative-action dock
:parameters (?p - Pier ?n - Oil_Tanker ?pt - Port)
:duration (= ?duration (dockingTime ?n))
:condition
  (at start
    (and
      (not (docked ?n))
      (not (tankerConnected ?n))
      (not (tankerUnloading ?n))
      (atPort ?n ?pt)
      (pierFree ?p)
      (>= (maxCapacity ?p) (tankerCapacity ?n))
      (belongs_to ?p ?pt)
    )
  )
)
:effect
  (at end
    (and
      (docked ?n)
      (not (tankerConnected ?n))
      (not (tankerUnloading ?n))
      (docks_at ?n ?p)
      (increase (totalNumberOfTimeActions) 1)
      (increase (domainMetric) (* (wNA) 1))
      (not (pierFree ?p))
    )
  )
)
)
(:durative-action undock
:parameters (?p - Pier ?n - Oil_Tanker ?pt - Port)
:duration (= ?duration (undockingTime ?n))
:condition
  (at start
    (and
      (docked ?n)
      (not (tankerConnected ?n))
      (not (tankerUnloading ?n))
      (docks_at ?n ?p)
      (atPort ?n ?pt)
      (not (pierFree ?p))
    )
  )
)
)

```

```

:effect
  (at end
    (and
      (not (docked ?n))
      (not (tankerConnected ?n))
      (not (tankerUnloading ?n))
      (increase (totalNumberOfTimeActions) 1)
      (increase (domainMetric) (* (wNA) 1))
      (pierFree ?p)
      (not (docks_at ?n ?p))
      (not (atPort ?n ?pt))
    )
  )
)

(:durative-action connectTankerTank
:parameters (?n - Oil_Tanker ?t - Tank ?pt - Port)
:duration (= ?duration 0)
:condition
  (at start
    (and
      (docked ?n)
      (not (tankerConnected ?n))
      (not (tankerUnloading ?n))
      (atPort ?n ?pt)
      (not (tankConnected ?t))
      (not (tankChangingVolume ?t))
      (not (tankOperationType ?t))
      (is_at ?t ?pt)
    )
  )
)

:effect
  (at end
    (and
      (docked ?n)
      (tankerConnected ?n)
      (not (tankerUnloading ?n))
      (is_linked_to ?n ?t)
      (tankConnected ?t)
      (not (tankChangingVolume ?t))
      (tankOperationType ?t)
    )
  )
)

(:durative-action pumpOilTankerTankFull
:parameters (?n - Oil_Tanker ?t - Tank ?ol - Type_of_Crude_Oil ?class -
Class_of_Crude_Oil)
:duration (= ?duration (/ (inventory ?n ?ol) (oilFlowRate ?ol)))
:condition
  (at start
    (and
      (docked ?n)

```

```

        (tankerConnected ?n)
        (not (tankerUnloading ?n))
        (is_linked_to ?n ?t)
        (<= (inventory ?n ?ol) (- (maxVol ?t) (currentVol ?t)))
        (> (inventory ?n ?ol) 0)
        (tankConnected ?t)
        (not (tankChangingVolume ?t))
        (tankOperationType ?t)
        (< (currentVol ?t) (maxVol ?t))
        (cl ?t ?class)
        (composed_by ?class ?ol)
    )
)

:effect
  (at end
    (and
      (docked ?n)
      (tankerConnected ?n)
      (tankerUnloading ?n)
      (assign (inventory ?n ?ol) 0)
      (tankConnected ?t)
      (tankChangingVolume ?t)
      (tankOperationType ?t)
      (increase (currentVol ?t) (inventory ?n ?ol))
      (increase (totalLossAtPort) (* (inventory ?n ?ol) (-
(maxClassValue) (valueAtPort ?class)))))
      (increase (domainMetric) (* (wLP) (* (inventory ?n ?ol) (-
(maxClassValue) (valueAtPort ?class)))))
      (increase (totalNumberOfTimeActions) 1)
      (increase (domainMetric) (* (wNA) 1))
    )
  )
)

(:durative-action disconnectTankerTank
:parameters (?n - Oil_Tanker ?t - Tank)
:duration (= ?duration 0)
:condition
  (at start
    (and
      (docked ?n)
      (tankerConnected ?n)
      (tankerUnloading ?n)
      (is_linked_to ?n ?t)
      (tankConnected ?t)
      (tankChangingVolume ?t)
      (tankOperationType ?t)
    )
  )
)

:effect
  (at end
    (and
      (docked ?n)
      (not (tankerConnected ?n))

```

```

        (not (tankerUnloading ?n))
        (not (tankConnected ?t))
        (not (tankChangingVolume ?t))
        (tankOperationType ?t)
        (not (is_linked_to ?n ?t))
    )
)

(:durative-action connectTankPipeline
:parameters (?t - Tank ?o - Pipeline ?pt - Port)
:duration (= ?duration 0)
:condition
    (at start
        (and
            (not (tankConnected ?t))
            (not (tankChangingVolume ?t))
            (not (tankOperationType ?t))
            (is_at ?t ?pt)
            (pipelineFree ?o)
            (not (pipelineTransporting ?o))
            (has_one_end_at ?o ?pt)
        )
    )
:effect
    (at end
        (and
            (tankConnected ?t)
            (not (tankChangingVolume ?t))
            (not (tankOperationType ?t))
            (is_connected_to ?t ?o)
            (not (pipelineFree ?o))
            (not (pipelineTransporting ?o))
        )
    )
)

(:durative-action pumpTankPipeline
:parameters (?t - Tank ?o - Pipeline ?class - Class_of_Crude_Oil
?pipelineClass - Class_of_Crude_Oil)
:duration (= ?duration (/ (- (currentVol ?t) (minVol ?t)) (classFlowRate
?class)))
:condition
    (at start
        (and
            (tankConnected ?t)
            (not (tankChangingVolume ?t))
            (not (tankOperationType ?t))
            (is_connected_to ?t ?o)
            (cl ?t ?class)
            (not (pipelineFree ?o))
            (not (pipelineTransporting ?o))
            (lastOil ?o ?pipelineClass)
        )
    )
)

```

```

    )

:effect
  (at end
    (and
      (tankConnected ?t)
      (tankChangingVolume ?t)
      (not (tankOperationType ?t))
      (assign (currentVol ?t) (minVol ?t))
      (increase (totalLossAtRefinery) (* (- (currentVol ?t) (minVol ?t))
      (- (maxClassValue) (valueAtRefinery ?class ?o))))
      (increase (volumeSent) (- (currentVol ?t) (minVol ?t)))
      (increase (domainMetric) (* (* (wLR) (- (currentVol ?t) (minVol
?t))) (- (maxClassValue) (valueAtRefinery ?class ?o))))
      (not (pipelineFree ?o))
      (pipelineTransporting ?o)
      (increase (totalInterfaceCost) (interfaceCosts ?pipelineClass
?class))
      (increase (domainMetric) (* (wIC) (interfaceCosts ?pipelineClass
?class)))
      (increase (totalNumberOfTimeActions) 1)
      (increase (domainMetric) (* (wNA) 1))
      (not (lastOil ?o ?pipelineClass))
    )
  )
)

(:durative-action disconnectTankPipeline
:parameters (?t - Tank ?o - Pipeline ?class - Class_of_Crude_Oil)
:duration (= ?duration 0)
:condition
  (at start
    (and
      (tankConnected ?t)
      (tankChangingVolume ?t)
      (not (tankOperationType ?t))
      (is_connected_to ?t ?o)
      (cl ?t ?class)
      (not (pipelineFree ?o))
      (pipelineTransporting ?o)
    )
  )
)

:effect
  (at end
    (and
      (not (tankConnected ?t))
      (not (tankChangingVolume ?t))
      (not (tankOperationType ?t))
      (pipelineFree ?o)
      (not (pipelineTransporting ?o))
      (lastOil ?o ?class)
      (not (is_connected_to ?t ?o))
    )
  )
)

```

```

)

(:durative-action connectTankerDecantingTank
:parameters (?n - Oil_Tanker ?t - Tank ?pt - Port)
:duration (= ?duration 0)
:condition
  (at start
    (and
      (docked ?n)
      (not (tankerConnected ?n))
      (not (tankerUnloading ?n))
      (atPort ?n ?pt)
      (not (tankConnected ?t))
      (not (tankChangingVolume ?t))
      (tankOperationType ?t)
      (is_at ?t ?pt)
    )
  )
)

:effect
  (at end
    (and
      (docked ?n)
      (tankerConnected ?n)
      (not (tankerUnloading ?n))
      (is_linked_to ?n ?t)
      (tankConnected ?t)
      (not (tankChangingVolume ?t))
      (tankOperationType ?t)
    )
  )
)

)

(:durative-action pumpOilTankerTankParcial
:parameters (?n - Oil_Tanker ?t - Tank ?ol - Type_of_Crude_Oil ?class -
Class_of_Crude_Oil)
:duration (= ?duration (/ (- (maxVol ?t) (currentVol ?t)) (oilFlowRate
?ol)))
:condition
  (at start
    (and
      (docked ?n)
      (tankerConnected ?n)
      (not (tankerUnloading ?n))
      (is_linked_to ?n ?t)
      (> (inventory ?n ?ol) (- (maxVol ?t) (currentVol ?t)))
      (tankConnected ?t)
      (not (tankChangingVolume ?t))
      (tankOperationType ?t)
      (< (currentVol ?t) (maxVol ?t))
      (cl ?t ?class)
      (composed_by ?class ?ol)
    )
  )
)

:effect

```

```

    (at end
      (and
        (docked ?n)
        (tankerConnected ?n)
        (tankerUnloading ?n)
        (decrease (inventory ?n ?ol) (- (maxVol ?t) (currentVol ?t)))
        (tankConnected ?t)
        (tankChangingVolume ?t)
        (tankOperationType ?t)
        (assign (currentVol ?t) (maxVol ?t))
        (increase (totalLossAtPort) (* (- (maxVol ?t) (currentVol ?t)) (-
(maxClassValue) (valueAtPort ?class))))
        (increase (domainMetric) (* (* (wLP) (- (maxVol ?t) (currentVol
?t)))) (- (maxClassValue) (valueAtPort ?class))))
        (increase (totalNumberOfTimeActions) 1)
        (increase (domainMetric) (* (wNA) 1))
      )
    )
  )

(:durative-action waitForDecantation
:parameters (?t - Tank)
:duration (= ?duration (decantingTime ?t))
:condition
  (at start
    (and
      (not (tankConnected ?t))
      (not (tankChangingVolume ?t))
      (tankOperationType ?t)
    )
  )
)

:effect
  (at end
    (and
      (not (tankConnected ?t))
      (not (tankChangingVolume ?t))
      (not (tankOperationType ?t))
      (increase (totalNumberOfTimeActions) 1)
      (increase (domainMetric) (* (wNA) 1))
    )
  )
)
)
)

```