Cristiane Salgado Pereira

**Otimização Multiobjetivo da Programação de Petróleo em
Refinaria por Programação Genética em Linguagem
Específica de Domínio**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para
obtenção do título de Mestre pelo Programa de Pós-
Graduação em Engenharia Elétrica da PUC-Rio.

Orientador: Prof. Marco Aurélio Cavalcanti Pacheco
Co-Orientador: Prof. Douglas Mota Dias

Rio de Janeiro
Novembro de 2012

Cristiane Salgado Pereira

Otimização Multiobjetivo da Programação de Petróleo em Refinaria por Programação Genética em Linguagem Específica de Domínio

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Marco Aurélio Cavalcanti Pacheco

Orientador

Departamento de Engenharia Elétrica – PUC-Rio

Prof. Douglas Mota Dias

Co-Orientador

Departamento de Engenharia Elétrica – PUC-Rio

Prof. André Vargas Abs da Cruz

Departamento de Engenharia Elétrica – PUC-Rio

Prof. Marcus Vinicius de Oliveira Magalhães

Petrobras

Prof. Paulo Cesar Ribas

Petrobras

Prof. José Eugenio Leal

Coordenador Setorial do Centro
Técnico Científico

Rio de Janeiro, 30 de novembro de 2012

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, da autora e do orientador.

Cristiane Salgado Pereira

Graduada em Engenharia Química pela Universidade Federal do Rio de Janeiro, em 2002.

Ficha Catalográfica

Pereira, Cristiane Salgado

Otimização multiobjetivo da programação de petróleo em refinaria por programação genética linear em linguagem específica de domínio / Cristiane Salgado Pereira ; orientador: Marco Aurélio C. Pacheco; co-orientador: Douglas Mota Dias; – 2012.

135 f. : il. ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2012.

Inclui bibliografia.

1. Engenharia Elétrica – Teses; 2. Programação Genética; 3. Computação Evolucionária; 4. Inspiração Quântica; 5. Linguagem Específica de Domínio; 6. Programação de Produção; 7. Refinaria; 8. Programação de Petróleo. I. Pacheco, Marco Aurélio Cavalcanti. II. Dias, Douglas Mota. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 621.3

Aos meus pais, Ilda e Valdemir

Agradecimentos

A Deus, por Seu exemplo e pelas pessoas que fazem parte da minha vida.

Aos meus pais, pelo amor, lealdade e por me transmitirem os valores de honestidade, respeito e educação.

Ao Wilson, pela enorme paciência, risadas, conversas, críticas e discussões. Enfim, por ser companheiro.

A tantos amigos queridos. Muitas fases, encontros e desencontros, semelhanças e diferenças. E a certeza de que tê-los em minha vida me faz sentir muito feliz. Agora, terão que me dedicar mais tempo de novo!

Aos meus orientadores Marco Aurélio Pacheco, pela oportunidade de desenvolver este trabalho, e Douglas Mota Dias, pelas ideias incríveis, paciência e disposição de fazê-lo dar certo.

Aos colegas e amigos da Petrobras que muito me ensinaram e pela constante troca de conhecimento e experiência. Em especial, a Alexandre Fuelber e Marcelo Kenji pelas muitas aulas e, a Estefane Horn e Marcelo Maia que ainda “seguraram as pontas” durante este trabalho.

À Petrobras, por investir na minha formação.

Aos colegas do ICA/PUC, em especial André Vargas, pela paciência nas explicações e pela inestimável ajuda.

Aos meus amores “de quatro patas”, pela companhia constante.

Resumo

Pereira, Cristiane Salgado; Pacheco, Marco Aurélio C.; Dias, Douglas Mota. **Otimização Multiobjetivo da Programação de Petróleo em Refinaria por Programação Genética em Linguagem Específica de Domínio.** Rio de Janeiro, 2012. 135p. Dissertação de Mestrado - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

A programação de produção em refinaria (*scheduling*) pode ser compreendida como uma sequência de decisões que buscam otimizar a alocação de recursos, o sequenciamento de atividades e a realização temporal dessas atividades, respeitando um conjunto de restrições de diferentes naturezas e visando o atendimento de múltiplos objetivos onde fatores como atendimento à demanda de produção e minimização de variações operacionais nos equipamentos coexistem na mesma função. Este trabalho propõe o uso da técnica de Programação Genética para automatizar a criação de programas que representem uma solução completa de programação de petróleo em uma refinaria dentro de um horizonte de tempo. Para a evolução destes programas foi desenvolvida uma linguagem específica para o domínio de problemas de *scheduling* de petróleo e aplicada de forma a representar as principais atividades do estudo de caso. Para tal, a primeira etapa consistiu da avaliação de alguns cenários de programação de produção de forma a selecionar as atividades que deveriam ser representadas e como fazê-lo. No modelo proposto, o cromossomo quântico guarda a superposição de estados de todas as soluções possíveis e, através do processo evolutivo e observação dos genes quânticos, o cromossomo clássico é criado como uma sequência linear de instruções a serem executadas. As instruções executadas representam o *scheduling*. A orientação deste processo é feita através de uma função de aptidão multiobjetivo que hierarquiza as avaliações sobre o tempo de operação das unidades de destilação, o prazo para descarregamento de navios, a utilização do duto que movimenta óleo entre terminal e refinaria, além de fatores como número de trocas de tanques e uso de tanques de injeção nas unidades de destilação. No desenvolvimento deste trabalho foi contemplado um estudo sobre o conjunto de parâmetros para o modelo desenvolvido com base em um dos cenários de programação selecionados. A partir desta definição, para avaliação do modelo proposto, foram executadas diversas rodadas para cinco cenários de

programação de petróleo. Os resultados obtidos foram comparados com estudo desenvolvido usando algoritmos genéticos cujas atividades, no cromossomo, possuem representação por ordem. A programação genética apresentou percentual de soluções aceitas variando entre 25% e 90% dependendo da complexidade do cenário, sendo estes valores superiores ao obtido usando Algoritmos Genéticos em todos os cenários, com esforço computacional menor.

Palavras-chave

Programação Genética; Computação Evolucionária; Inspiração Quântica; Linguagem Específica de Domínio; Programação de Produção; Refinaria; Programação de Petróleo; Programação Genética Linear Orientada à Gramática

Abstract

Pereira, Cristiane Salgado; Pacheco, Marco Aurélio C. (Advisor); Dias, Douglas Mota (Co-Advisor). **Petroleum Scheduling Multiobjective Optimization for Refinery by Genetic Programming using Domain Specific Language**. Rio de Janeiro, 2012. 135p. MSc. Dissertation – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Refinery scheduling can be understood as a sequence of decisions that targets the optimization of available resources, sequencing and execution of activities on proper timing; always respecting restrictions of different natures. The final result must achieve multiple objectives guaranteeing co-existence of different factors in the same function, such as production demand fulfillment and minimize operational variation. In this work it is proposed the use of the genetic programming technique to automate the building process of programs that represent a complete oil scheduling solution within a defined time horizon. For the evolution of those programs, it was developed a domain specific language to translate oil scheduling instructions that was applied to represent the most relevant activities for the proposed case studies. For that, purpose first step was to evaluate a few real scheduling scenarios to select which activities needed to be represented and how to do that. On the proposed model, each quantum chromosome represents the overlapping of all solutions and by the evolutionary process (and quantum gene measurement) the classic chromosome is created as a linear sequence of scheduling instructions to be executed. The orientation for this process is performed through a multi-object fitness function that prioritizes the evaluations according to: the operating time of the atmospheric distillation unities, the oil unloading time from the ships, the oil pipeline operation to transport oil to the refinery and other parameters like the number of charge tanks switchover and injection tank used for the distillation unities. The scope of this work also includes a study about tuning for the developed model based in one of the considered scenarios. From this set, an evaluation of other different scheduling scenarios was performed to test the model. The obtained results were then compared with a developed model that uses genetic algorithms with order representation for the activities. The proposed model showed between 25%-90% of good solutions depending on the scenario complexity. Those results exhibit higher percentage of good solutions requiring less computational effort than the

ones obtained with the genetic algorithms.

Keywords

Genetic Programming; Evolutionary Computing; Quantum-Inspired; Domain Specific Language; Production Scheduling; Refinery; Petroleum Scheduling; Grammar Oriented Linear Genetic Programming

Sumário

1 Introdução	17
1.1. Apresentação	17
1.1.1. Planejamento e Programação da Produção	18
1.2. Motivação	20
1.3. Objetivo	21
1.4. Descrição da Pesquisa	21
1.5. Estrutura do Texto	22
1.6. Termos Utilizados neste Trabalho	23
2 Programação de Produção em Refinarias de Petróleo	25
2.1. A Atividade “Programação de Produção”	25
2.1.1. Decisões de Programação de Produção	26
2.2. Abordagens para Solução de Problemas de Programação de Produção	28
3 Programação Genética	36
3.1. Introdução	36
3.2. Conceitos	37
3.2.1. Terminais e Funções	37
3.2.2. Avaliação e Seleção	38
3.2.3. Algoritmo Básico	39
3.3. Estruturas de Representação	40
3.3.1. Programação Genética em Árvore	41
3.3.2. Programação Genética Linear	42
3.3.3. Programação Genética Orientada à Gramática	42
3.4. Programação Genética Linear com Inspiração Quântica	44
3.4.1. Conjunto de Funções e Terminais	45
3.4.2. Indivíduo Quântico	45
3.4.3. Indivíduo Clássico	46
3.4.4. Operador Quântico	46
3.5. Motivação	46
4 Modelo Proposto	48
4.1. Configuração da Refinaria Modelada	48

4.1.1. Representação Física	48
4.1.2. Atividades de Programação	52
4.1.3. Informações de Entrada e Decisões do Modelo	56
4.2. Linguagem Específica de Domínio	56
4.3. Representação dos Indivíduos	60
4.3.1. Indivíduo Quântico	61
4.3.2. Indivíduo Clássico	64
4.3.3. Avaliação	66
4.3.4. Evolução	68
4.3.5. Funcionamento do Modelo	68
4.3.6. Simulador de Planta	72
5 Estudos de Caso	74
5.1. Descrição	74
5.2. Cenários	75
5.2.1. Restrições	77
5.2.2. Seleção dos Cenários	78
5.3. Definição dos parâmetros do modelo	83
5.3.1. Critério Usado para Definir as Soluções Aceitas	87
5.4. Avaliação do Conjunto de Terminais	89
5.5. Resultados dos Cenários	91
5.5.1. CEN01	92
5.5.2. CEN02	98
5.5.3. CEN03	101
5.5.4. CEN04	105
5.5.5. CEN05	108
5.5.6. Comentários Gerais	110
5.5.7. Comparação com Modelo em AG	112
6 Conclusões e Trabalhos Futuros	118
7 Referências Bibliográficas	123
APÊNDICE A Dados de Entrada e Outros Gráficos das Soluções Apresentadas como Exemplo em Cada Cenário	126

Lista de figuras

Figura 1 – Níveis hierárquicos de decisão em planejamento e programação	19
Figura 2 – Estrutura de desenvolvimento da pesquisa	22
Figura 3 - Algoritmo básico de execução da PG.	42
Figura 4 – Representação em árvore e operadores	43
Figura 5 - Exemplo de estrutura de cromossomo da PG Linear	42
Figura 6 - Exemplo de estrutura BNF para definir sintaxe de uma linguagem específica	43
Figura 7 – Exemplo de produção possível	43
Figura 8 – Representação do problema proposto	49
Figura 9 – Representação esquemática da movimentação do duto	55
Figura 10 – Estrutura BNF que representa a gramática da linguagem específica para o problema proposto.	60
Figura 11 – Representação de um <i>qudit</i> com probabilidades não uniformemente distribuídas.	62
Figura 12 – Função que define o valor do <i>token</i> associado à observação do <i>qudit</i>	62
Figura 13 – Exemplo de distribuição de probabilidades do símbolo <func> em um gene quântico	62
Figura 14 – Estrutura do gene quântico	63
Figura 15 – Todas as possibilidades de estados do <i>qudit</i> de um gene quântico	64
Figura 16 – Ciclo de uma geração da PGLOGIQ	70
Figura 17 – Ciclo simplificado do modelo proposto	71
Figura 18 – Fluxograma de funcionamento do modelo proposto	73
Figura 19 – Subconjunto dos rendimentos volumétricos e propriedades dos petróleos para CAMPANHA01 da UDA-01	76
Figura 20 – Curva evolutiva do melhor indivíduo de uma rodada com convergência prematura	84
Figura 21 – Curva evolutiva do melhor indivíduo de uma rodada sem convergência prematura	85
Figura 22 – Curva evolutiva da média da população na rodada sem convergência prematura	85
Figura 23 – Distribuição de soluções para o melhor conjunto de parâmetros, Config1	88

Figura 24 - Distribuição de soluções para o segundo melhor conjunto de parâmetros, Config2	89
Figura 25 – Distribuição de soluções aceitas para CEN01	92
Figura 26 – Gráfico de gantt de uma solução ideal para CEN01	94
Figura 27 – Gráfico de estoque de uma solução ideal para CEN01	94
Figura 28 - Distribuição de soluções aceitas para CEN02	98
Figura 29 - Gráfico de gantt de uma solução ideal para CEN02	99
Figura 30 - Gráfico de estoque de uma solução ideal para CEN02	99
Figura 31 - Distribuição de soluções aceitas para CEN03	101
Figura 32 - Gráfico de gantt de uma solução ideal para CEN03	103
Figura 33 - Gráfico de estoques de uma solução ideal para CEN03	103
Figura 34 - Distribuição de soluções aceitas para CEN04	105
Figura 35 - Gráfico de gantt de uma solução boa para CEN04	106
Figura 36 - Gráfico de estoque de uma solução boa para CEN04	106
Figura 37 - Distribuição de soluções aceitas para CEN05	108
Figura 38 - Gráfico de gantt de uma solução ideal para CEN05	109
Figura 39 - Gráfico de estoque de uma solução ideal para CEN05	109
Figura 40 – Ciclo de evolução do modelo em algoritmos genéticos	115
Figura 41 – Comparação do percentual de soluções aceitas em cada cenário usando modelagens em PGLOGIQ, AG 240 mil e AG 60 mil avaliações.	116
Figura 42 – Gráficos de Acidez das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN01.	127
Figura 43 - Gráficos de Enxofre das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN01.	127
Figura 44 – Perfil de Rendimento para todos os cortes de destilação das UDAs em CEN01	127
Figura 45 – Gráficos de Acidez das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN02.	129
Figura 46 - Gráficos de Enxofre das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN02	129
Figura 47 – Perfil de Rendimento para todos os cortes de destilação das UDAs em CEN02	129
Figura 48 – Gráficos de Acidez das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN03.	131
Figura 49 - Gráficos de Enxofre das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN03	131

Figura 50 – Perfil de Rendimento para todos os cortes de destilação das UDAs em CEN03	131
Figura 51 – Gráficos de Acidez das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN04.	133
Figura 52 – Gráficos de Enxofre das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN04.	133
Figura 53 – Perfil de Rendimento para todos os cortes de destilação das UDAs em CEN04	133
Figura 54 – Gráficos de Acidez das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN05.	135
Figura 55 – Gráficos de Enxofre das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN05.	135
Figura 56 – Perfil de Rendimento para todos os cortes de destilação das UDAs em CEN01.	135

Lista de tabelas

Tabela 1 – Exemplos de diferentes configurações de navios	49
Tabela 2 - Lastro e capacidade total dos tanques do terminal	50
Tabela 3 - Lastro e capacidade total dos tanques da refinaria	51
Tabela 4 - Estrutura dos tipos de dados utilizados no sistema	58
Tabela 5 – Funções e argumentos desenvolvidos para a linguagem	59
Tabela 6 – Exemplo de distribuição de probabilidades para dois genes quânticos	71
Tabela 7 – Exemplo de distribuição de probabilidades para os mesmos elementos terminais em genes quânticos diferentes	72
Tabela 8 – Propriedades dos petróleos utilizados	76
Tabela 9 – Restrições de qualidade aplicadas aos cenários	77
Tabela 10 – Duração, inventário e configuração de navios dos cenários estudados.	78
Tabela 11 – Pre-Def de CEN01	79
Tabela 12 – Vazões de navio e UDAs de CEN01	79
Tabela 13 - Pre-Def de CEN02	80
Tabela 14 - Vazões de navio e UDAs de CEN02	80
Tabela 15 – Pre-Def de CEN03	81
Tabela 16 - Vazões de navio e UDAs de CEN03	81
Tabela 17 - Pre-Def de CEN04	82
Tabela 18 - Vazões de navio e UDAs de CEN04	82
Tabela 19 - Pre-Def de CEN05	83
Tabela 20 - Vazões de navio e UDAs de CEN05	83
Tabela 21 – Parâmetros avaliados para o modelo evolutivo	86
Tabela 22 – Configurações pré-selecionadas dos parâmetros	86
Tabela 23 – Critério de aceitação de uma solução de <i>scheduling</i>	88
Tabela 24 – Tabela comparativa dos objetivos entre programação real e modelo proposto para CEN01	95
Tabela 25 – Programa gerado pelo modelo evolutivo para CEN01	97
Tabela 26 - Tabela comparativa dos objetivos entre programação real e modelo proposto para CEN02	100
Tabela 27 - Tabela comparativa dos objetivos entre programação real e modelo proposto para CEN03	104

Tabela 28 - Tabela comparativa dos objetivos entre programação real e modelo proposto para CEN04	107
Tabela 29 - Tabela comparativa dos objetivos entre programação real e modelo proposto para CEN05	110
Tabela 30 – Taxa de aplicação dos operadores do modelo evolutivo em AG	113
Tabela 31 – Dados iniciais de composição e volume para CEN01	126
Tabela 32 - Dados iniciais de composição e volume para CEN02	128
Tabela 33 - Dados iniciais de composição e volume para CEN03	130
Tabela 34 - Dados iniciais de composição e volume para CEN04	132
Tabela 35 - Dados iniciais de composição e volume para CEN05	134

1

Introdução

1.1. Apresentação

A Cadeia de Suprimento pode ser entendida como a integração dos processos de planejamento, fornecimento, produção e comercialização de produtos que abrange desde os fornecedores de seus fornecedores até os clientes de seus clientes, alinhando estratégia operacional com fluxo material, produtivo e de informação [1]. O conceito foi introduzido na metade da década de 80 e atualmente é usado como base sobre o qual são aplicadas poderosas técnicas de gerenciamento de operações interorganizacionais, integração de sistemas e compartilhamento de informações [2].

A indústria do petróleo possui as mesmas características descritas acima. Envolve desde a obtenção de diversos tipos de óleo, seja através de compra no mercado internacional ou exploração e produção de campos petrolíferos, transporte, processamento e consequente conversão em produtos combustíveis e/ou petroquímicos, até a entrega destes produtos aos clientes finais ou distribuidores. No entanto, o segmento possui particularidades, tais como: poucos fornecedores de matéria-prima, instabilidade dos preços dos óleos, necessidade de continuidade operacional das unidades de processo, mesmo em cenários de baixa demanda, dificuldade do planejamento e a extensão da própria cadeia [2].

O aprimoramento da cadeia de suprimentos – considerado fator essencial para a manutenção da vantagem competitiva – tornou-se uma preocupação para as empresas. Esta necessidade existe também na indústria do refino de petróleo, onde as margens são reduzidas e há dificuldade em obter diferenciação de produtos [3].

Nesse contexto de aprimoramento, dois elos muito importantes da cadeia de suprimento de derivados de petróleo são as atividades de planejamento (*planning*) e programação (*scheduling*) da produção.

1.1.1. Planejamento e Programação da Produção

O planejamento visa definir as condições que aumentem a lucratividade da organização, buscando a melhor relação de uso dos recursos para produção, distribuição e armazenamento de produtos de forma a atender às demandas previstas [4].

Dependendo do tamanho e da estrutura da organização existem diferenças específicas nos diversos níveis de planejamento, mas de maneira geral, o processo de tomada de decisões avança, hierarquicamente, entre os níveis estratégico, tático e operacional. Este último, compreendendo a programação de produção.

O planejamento estratégico tem por objetivo a maximização da margem de lucro da corporação, obtida dentro de uma visão global e agregada, ainda que isso signifique que partes dessa estrutura não operem localmente em suas condições ótimas. O horizonte de planejamento estratégico é da ordem de um ou mais anos, sendo feito com periodicidade máxima de 3 meses. O escopo de análise é o maior da cadeia, incluindo variáveis como: obtenção de matéria-prima (importação/exportação, produção), previsões de demanda, variações de preço e oferta no mercado internacional, capacidade de refino, alterações no parque de produção, malhas logísticas e etc. O planejamento estratégico baliza as decisões sobre as importações/exportações, alocação de matéria-prima, metas de produção e de movimentação para atender aos mercados consumidores. Normalmente uma ferramenta com modelagem baseada na técnica de programação linear auxilia neste processo de tomada de decisão [5,6]. Seguindo a hierarquia de decisão, as informações do planejamento estratégico são utilizadas no nível seguinte.

O planejamento tático tem por objetivo a maximização da margem bruta da unidade produtiva, neste caso a refinaria, respeitando as condições e diretrizes definidas pelo planejamento estratégico. O horizonte de planejamento é da ordem poucos meses, feito com maior frequência. O escopo de análise é mais restrito, localizado nas variáveis diretamente relacionadas à refinaria, porém a representatividade da unidade produtiva é mais detalhada. Apesar da característica de não linearidade do problema, uma vez que é importante considerar volumes, vazões e valores de propriedades das correntes, é comum que se façam simplificações na modelagem de maneira que, assim como no planejamento estratégico, a programação linear é a técnica mais utilizada nas

ferramentas de apoio a essa etapa. Alguns exemplos de orientações do planejamento tático para o operacional são: média de utilização de capacidade das unidades de processo, modo de operação (campanha) das unidades, média de produção de cada produto e algumas diretrizes operacionais, como o valor médio da propriedade RCR (resíduo de carbono *Ramsbottom*) para carga da unidade de processo FCC (*Fluid Catalytic Cracking*) [6].

A “programação de produção”, também conhecida como *scheduling*, faz a ligação entre o planejamento e as operações da planta, através da distribuição temporal das ações que foram planejadas nas etapas anteriores. A programação é refeita diariamente e seu horizonte de visualização está em torno de sete dias. Em outras palavras, se por um lado, a atividade de planejamento busca respostas às questões como, por exemplo, “Qual a melhor mistura de crus a ser processada?” e “Qual o melhor perfil de derivados a ser produzido?”, a atividade de programação de produção, por sua vez, busca respostas às questões como “Quando iniciar a produção de um perfil de derivados e quanto tempo mantê-la?”, “Qual o sequenciamento de tanques que deve ser feito para compor a carga da unidade de processo?”, “Qual o sequenciamento para receber produtos?”, “Que volume transferir de um tanque a outro?”, “Que correntes e em que quantidades utilizar para compor o produto final?”, “Que vazões serão definidas dentro do período?”, “Como administrar o estoque?” etc. Todas essas questões devem ser respondidas de modo a gerar pelo menos um modo viável de operar a refinaria, compatibilizando a programação de chegada de crus e a de entrega de produtos. A relação entre distribuição temporal e abrangência entre os níveis hierárquicos de decisão é apresentada na Figura 1.

Figura 1 – Níveis hierárquicos de decisão em planejamento e programação

1.2. Motivação

Dadas as limitações de recursos computacionais e a natureza complexa dos problemas de programação de produção, caracterizada pelo grande número de variáveis, não-linearidades e restrições, durante muitos anos o uso de técnicas de otimização para problemas de *scheduling* esteve restrito a problemas menores e simplificados, não correspondendo a problemas reais de uma refinaria [3,7]. Com a evolução dos computadores e das técnicas de otimização, principalmente associadas à solução de problemas combinatórios, os problemas de *scheduling* passaram a ter maior importância no setor industrial [3].

Considerando que a indústria de petróleo movimenta bilhões de dólares anualmente em um ambiente onde, muitas vezes, as margens são reduzidas, é razoável admitir que soluções que melhorem os processos de tomada de decisão devem se traduzir em aumento da rentabilidade da indústria e, portanto, representem potencial interesse para as organizações [6,7].

A motivação de desenvolvimentos utilizando técnicas de inteligência computacional está na sua capacidade de gerar soluções otimizadas sem a necessidade de modelagem usando representações matemáticas em forma fechada. Através da função de aptidão do modelo as soluções são avaliadas de acordo com as características mais importantes do problema.

Estudos foram e continuam sendo realizados utilizando a técnica de algoritmos genéticos (AG) para otimização da atividade de programação da produção [3,8,9,10]. Entretanto, um fator representa um desafio sobre essa modelagem: a complexidade do problema, que implica em indivíduos (cromossomos) de tamanho muito grande cuja representação compromete o desempenho do sistema.

Este trabalho se propõe a definir uma linguagem de programação simples, capaz de gerar programas que representem uma solução completa de programação de produção de petróleo em uma refinaria, e a utilizar uma técnica de computação evolutiva – Programação Genética (PG) – para automatizar a criação destes programas e otimizá-los com relação aos objetivos da produção.

1.3. Objetivo

O objetivo principal deste trabalho é avaliar a aplicabilidade de um modelo de otimização baseado em programação genética com inspiração quântica [11] e utilizando uma linguagem específica de domínio para a solução de problemas de *scheduling* de petróleo. Considera-se *scheduling* de petróleo de uma refinaria todas as atividades associadas à movimentação e processamento de óleo que aconteçam dentro do escopo de decisão da refinaria.

Nesta dissertação pretende-se analisar a capacidade da modelagem proposta em gerar soluções de *schedule* operacionalmente viáveis e analisar a qualidade dessas soluções.

Ao final deste trabalho os resultados obtidos são comparados com os alcançados utilizando AG em um desenvolvimento continuado após a pesquisa apresentada em [3].

1.4. Descrição da Pesquisa

O trabalho que resultou nesta dissertação foi desenvolvido seguindo as etapas:

- Seleção de um modelo de refinaria cuja programação de petróleo envolvesse um maior conjunto de decisões. Neste caso, programação de navios, terminal e a área de petróleo da refinaria.
- Estudo das atividades, restrições, heurísticas e decisões envolvidas no dia a dia da programação. Desenvolvimento de uma linguagem para representação deste domínio e uso de uma notação formal para especificar a sintaxe dessa linguagem.
- Modelagem e implementação de algoritmo evolutivo baseado em programação genética utilizando a linguagem especificada e do operador evolutivo baseado em inspiração quântica.
- Definição dos parâmetros do algoritmo evolutivo e do operador quântico com base em um cenário de programação e em vários experimentos preliminares.
- Geração dos programas cuja execução representa uma solução completa de *scheduling*. Análise dessas soluções geradas para cinco cenários baseados em casos reais de uma refinaria. Nesta avaliação foram consideradas algumas das principais

preocupações do programador de produção: continuidade operacional, administração de estoques e qualidade da carga e dos produtos.

A Figura 2 apresenta um esquemático da metodologia usada no desenvolvimento deste estudo.

Figura 2 – Estrutura de desenvolvimento da pesquisa

1.5. Estrutura do Texto

Esta dissertação está estruturada da forma apresentada a seguir.

No capítulo 2 são discutidas algumas características das principais atividades que compõem a programação de produção e é realizado um levantamento bibliográfico sobre os desenvolvimentos acadêmicos para resolver problemas de *scheduling* em refinarias de petróleo, seja através de métodos matemáticos convencionais ou algoritmos evolutivos.

O capítulo 3 apresenta conceitos e parâmetros utilizados na computação evolutiva, mais especificamente na programação genética. Por fim, faz-se um pequeno levantamento bibliográfico sobre os tipos de problemas solucionados utilizando esta técnica.

O capítulo 4 detalha o estudo deste trabalho e, para tal, descreve a configuração da refinaria estudada e a linguagem específica de domínio desenvolvida, além de explicar o algoritmo evolutivo modelado.

No capítulo 5, primeiramente é feita uma avaliação de alguns parâmetros para o modelo. Uma vez definidos estes parâmetros, são apresentados os resultados para 5 cenários utilizados como estudos de caso e algumas considerações sobre as soluções obtidas.

O capítulo 6, finalmente, apresenta as conclusões deste trabalho e propostas de trabalhos futuros.

1.6. Termos Utilizados neste Trabalho

Algumas expressões e sinônimos utilizados ao longo de todo o trabalho são definidos aqui:

- Programação de produção: é considerada como sinônimo dos termos em inglês *scheduling*, utilizado no ambiente industrial para representar a atividade e *schedule*, utilizado para representar o resultado da atividade.
- TanqueCN: termo usado nesta dissertação para indicar cada um dos compartimentos de um navio que armazena uma mistura de petróleo. É uma representação abstrata de quantas misturas diferentes de petróleo um navio está transportando. O volume contido em um compartimento não se mistura ao de outros compartimentos.
- Mistura de petróleo: mistura uniforme de petróleo. É a representação de um volume de óleo com composição conhecida, podendo ser constituída de um (neste caso, composição deste único igual a 100%) ou mais petróleos (participação de todos totalizando 100%).
- Parcela: volume equivalente a uma mistura de petróleo que está dentro de um duto que movimenta óleo. O máximo deste volume é a própria capacidade do duto.
- *Campanha* ou *Modo de Operação*: significa um dado perfil de rendimentos e propriedades dos cortes de uma unidade de destilação atmosférica.
- *Item* ou *Item de bombeio*: refere-se a um volume produto que é recebido ou enviado pela refinaria. Seu volume total, composição e previsão de data de recebimento (ou envio) são conhecidos.

- Modelagem: refere-se a toda a estrutura desenvolvida nesta dissertação. Isto é, gramática, representação dos indivíduos clássico e quântico, operador quântico, algoritmo evolutivo e função de avaliação.
- Modelo: refere-se ao modelo evolutivo implementado, ou seja, é um subconjunto de Modelagem, que exclui referência à linguagem específica de domínio ou à gramática usada para estruturá-la.
- Simulador da planta: refere-se à parte de simulação do sistema que garante toda a consistência das informações da planta de processo, ou seja, volume dos equipamentos, propriedades de correntes ou misturas em equipamentos, horário das atividades, etc.
- Rodada ou execução do modelo: ciclo completo da otimização que resulta em um indivíduo apresentado como o melhor, depois de atendido o critério de terminação.
- Sistema: refere-se a toda estrutura necessária para execução da otimização, ou seja, modelo evolutivo, dados de entrada e informações de configuração.

2

Programação de Produção em Refinarias de Petróleo

A programação de produção pode ser entendida como a definição do que cada etapa da produção deve executar, desdobrada em diversas atividades de pequenos períodos de tempo que, totalizados, podem chegar a vários turnos de operação ou mesmo algumas semanas. Cada atividade de operação deve ter suas entradas e saídas definidas tanto do ponto de vista material (volumes, vazões, composições, propriedades) quanto do temporal, ou seja, com eventos ou datas de início e fim associados [3 *apud* 12].

Considera-se o *scheduling* como a parte central da estrutura de apoio à decisão operacional [13], uma vez que é uma atividade complexa que engloba a obtenção, coordenação e interpretação de um grande volume de dados, uma estratégia financeira e a solução de problemas combinatórios difíceis [14].

Embora o horizonte de programação possa ser de poucos dias a algumas semanas, as atividades definidas para o horizonte curto tem maior importância na avaliação da programação [6]. Este comportamento é reflexo da natureza instável das informações manipuladas na atividade, o que se traduz no fato de que boa parte das atividades previstas num horizonte médio de tempo tem alta probabilidade de não se realizar, seja por variações nas condições de recebimento de matéria-prima, no processamento, na necessidade de atendimento à demanda de produtos, em questões logísticas ou operacionais. A constatação natural destas características é que a programação de produção precisa ser refeita diariamente, tendo como ponto de partida as informações atualizadas sobre o que está acontecendo ou previsto para acontecer.

2.1.

A Atividade “Programação de Produção”

A programação de produção é caracterizada pela necessidade de tomar uma grande quantidade de decisões diariamente, lidando com operações contínuas e em batelada [6 *apud* 12]. Esta característica aliada à escassez de soluções de otimização que resolvam problemas reais de maneira eficiente, isto é, considerando boa parte das variáveis de decisão necessárias ao

programador, fazem com que a principal abordagem da atividade ainda seja gerar uma solução operacionalmente viável.

O programador de produção precisa reagir tanto às variabilidades do processo como às do negócio e deve ser capaz de prever as consequências de desvios em relação às atividades programadas. Um dos objetivos da programação é conseguir identificar possíveis situações de crise (por exemplo, baixo estoque) antes que elas aconteçam e tentar evitá-las.

Os programadores integram informações oriundas de diferentes origens. Precisam conhecer os estoques e propriedades atuais de produtos, níveis de produção, disponibilidade de equipamentos, expectativa de recebimento de matéria-prima, movimentações logísticas e previsões de entrega de produtos. Precisam conhecer como as diferentes unidades de processo interagem e estarem familiarizados com limitações de vazão de carga, rendimentos, tempos de processamento, tamanho de bateladas e quaisquer outros parâmetros que representem eficiência, economia ou restrições físicas do processo produtivo com o qual trabalham. O esforço de manipular e conciliar todas essas informações demanda muito tempo, restando pouco para análises de otimização e o processo decisório em si [6].

Frequentemente, uma métrica importante em um problema de programação é o desvio do planejamento, que deve ser minimizado. É de se esperar que existam graus de liberdade inerentes ao nível decisório da programação devido à natureza dinâmica das operações, como sobreestadia de navios, sequenciamento dos bombeios, dos modos de operação, composição de cargas, uso de tancagem e de utilidades, entre outras, que devem ser consideradas e, de alguma forma, otimizadas. Estas questões não são tratadas no nível do planejamento.

2.1.1. Decisões de Programação de Produção

Podem-se identificar os seguintes tipos de atividades, ou tarefas, a serem programadas:

- Definição de Campanha e Vazão de Carga das Unidades:

A campanha de uma unidade define como esta unidade irá operar para favorecimento de um perfil de rendimentos e propriedades dos derivados. A vazão de carga define a quantidade recebida e processada por uma unidade de processo. As decisões sobre campanha e vazão das unidades são válidas e

determinadas para todos os períodos de tempo, não precisando estar inter-relacionadas. Isto é, as mudanças de campanha e vazão não precisam ocorrer no mesmo instante para uma unidade. Esta relação também não existe entre unidades de processo. Isto é, quando é alterada a campanha ou vazão de uma unidade não precisam ocorrer mudanças nas demais.

Para que as metas do planejamento sejam viabilizadas na programação, esta deve levar em conta as campanhas e vazões das unidades que foram utilizadas para montar o plano de produção.

- Recebimento e Envio de Itens:

Esta etapa possui características diferentes dependendo do escopo de decisão da refinaria. Em caso de haver terminal dedicado e autonomia de decisão sobre a descarga de navios, o óleo bombeado do navio deve ser alocado no terminal, ou seja, como será distribuído o recebimento do volume total do navio entre os tanques do terminal. Também deve ser definido como será o bombeio de tanques do terminal para o oleoduto que movimentará o petróleo até a refinaria. Quando não há essa autonomia, a refinaria é informada sobre volume, composição e data prevista de chegada do óleo diretamente à refinaria. Independente da origem, quando o petróleo chega à refinaria deve ser alocado em tanques que mais tarde serão utilizados para dar carga em unidades de destilação (UDA).

Da mesma forma, quando os produtos finais estão especificados (ou certificados), devem ser entregues aos clientes. É chamada de “item” a programação para o recebimento ou envio de uma quantidade qualquer de um determinado produto em um dado instante, que tenha a refinaria como origem ou destino, respectivamente. Os itens representam a interface externa da refinaria (compras e vendas).

- Transferências entre os equipamentos:

Esta atividade é a mais frequente numa refinaria. A todo instante, durante todo o horizonte de programação, existe algum produto sendo transferido de um equipamento para outro. Podemos classificar as transferências como: entre dutos e tanques (envio e recebimento de itens); entre tanques/esferas e unidades (de processo ou equipamentos divisores/misturadores de corrente); entre as próprias unidades (de processo ou equipamentos divisores/misturadores de corrente); ou entre os próprios tanques/esferas;

Todas as origens, destinos, início e fim dessas atividades são decisões do programador de produção. No caso de divisores de corrente, também faz parte da atividade definir em que proporções a corrente será dividida entre seus diferentes destinos.

- Preparação e Circulação de Produtos em tanques:

A preparação e circulação são atividades que ocorrem em tanques e que, durante o período em que ocorre, o equipamento não pode enviar ou receber produto. A circulação é utilizada para tornar mais homogêneo o produto em tanque, e não é representada nesta dissertação. A preparação é o tempo durante o produto descansa em tanque para que haja decantação da salmoura recebida na mistura. Após a decantação, a salmoura é drenada do tanque para que o petróleo esteja apto a ser processado sem implicar em danos à operação estável das unidades de processo.

- Manutenção de equipamentos:

Esta atividade serve como restrição ao uso de determinados equipamentos durante um dado período. Sua ocorrência deve ser conhecida *a priori*, ou seja, antes de se programar a produção, pois interfere diretamente na alocação dos equipamentos.

2.2.

Abordagens para Solução de Problemas de Programação de Produção

A técnica mais utilizada nas ferramentas comerciais para abordar o problema de programação local é a simulação [3,6,14]. Através desta é possível apenas retornar as consequências das atividades propostas pelo programador para a planta modelada. Fica a critério do usuário a validação da solução após verificação das possíveis inviabilidades, como por exemplo, sobrealocação de recursos ou o não atendimento a restrições em geral. Esta abordagem é limitada na busca de soluções automáticas, isto é, não propõe uma solução para o problema e, portanto, a busca de uma solução fica restrita ao ciclo da tentativa-e-erro [6].

As áreas de planejamento da produção e controle de processo já possuem técnicas de resolução de seus problemas em estágio avançado e, para que a indústria se aproprie economicamente dos seus benefícios, é crucial que se

investida em técnicas de solução efetiva para os problemas de programação de produção [15].

Podem ser encontrados na literatura diversos trabalhos que apresentam técnicas de solução para aspectos específicos do problema de programação de produção ou fornecem soluções para problemas cuidadosamente definidos [6,7]. Vários destes utilizaram técnicas de programação matemática, normalmente das classes de Programação Matemática Mista-Inteira Linear (*Mixed-Integer Linear Programming* - MILP) ou de Programação Matemática Mista-Inteira Não Linear (*Mixed-Integer Nonlinear Programming* - MINLP) [6].

Em [7] é apresentado um trabalho no qual foi utilizado o problema de programação de petróleo em uma refinaria que recebe petróleo através de um oleoduto, alguns tanques de petróleo e uma unidade de destilação para avaliar as abordagens discreta e contínua de modelagem do tempo. Primeiros testes foram realizados somente sobre a questão quantitativa e assumindo diversas simplificações de forma que a modelagem MILP fosse compatível. Dentre essas simplificações, o tamanho do lote de tempo, definido em 8 horas, torna a modelagem restritiva, pois impacta a dinâmica de decisões da atividade. Uma segunda avaliação foi realizada com representação contínua do tempo onde os intervalos de tempo são definidos tendo como referencia os eventos de recebimento de óleo, conhecidos *a priori*. Nestes testes o impacto da representação contínua do tempo já foi notado através da diminuição significativa da quantidade de variáveis binárias, o que permitiu a discretização das relações não lineares e manteve o problema linear. Posteriormente foi realizada uma avaliação utilizando a mesma representação contínua do tempo, mas desta vez, o tratamento dado às propriedades das correntes manteve-o como um problema de otimização MINLP, resolvido utilizando um *solver* adequado para este tipo de problema. Neste trabalho não foram consideradas qualidade dos petróleos ou restrições para carga da unidade, tampouco a produção de derivados na unidade.

Em [6] é apresentado um trabalho no qual o autor desenvolveu uma formulação com representação contínua do tempo para propor o *schedule* cuja abrangência é um terminal, um duto de petróleo e algumas unidades de processo de uma refinaria. O problema foi decomposto em dois subsistemas: programação de petróleos e programação de produtos, sendo o segundo caso tratado com Programação Linear Mista-Inteira (MILP). Para o problema de programação de petróleo que, quando tratado com todas as suas restrições, é uma formulação MINLP, foi usada técnica de decomposição com o objetivo de

se obter bons resultados em tempo computacional razoável. Foram avaliadas as abordagens contínua e discreta para tratar o tempo. As propriedades de uma mistura de petróleo são abordadas indiretamente, através de limites de composições dos óleos. Esta abordagem apresentou resultados interessantes nos casos aplicados.

Em [16] é apresentado um trabalho no qual os autores propuseram um modelo com representação contínua do tempo para resolver o *scheduling* envolvendo recebimento de óleo, tanques de armazenamento para essas correntes, transferência destes tanques para tanques de carga, e carga de unidades de destilação. A modelagem considerou restrições de lastro e capacidade de tancagem, não operação simultânea de entrada e saída de vazão em um mesmo tanque, relação de exclusividade entre o tanque de carga e a unidade de destilação, além de garantir balanço volumétrico e de componentes. A solução proposta foi uma variação do algoritmo de aproximação externa (*Outer-Approximation*) [17]. Foram gerados limites inferiores e superiores para um espaço de busca do problema original, que então foram usados na convergência do modelo. Para obter o limite inferior, o problema original foi decomposto em áreas nas quais são usadas relaxação MILP. O limite superior foi obtido através da resolução do problema de programação não-linear (*Nonlinear Programming* - NLP) gerado pela fixação das variáveis binárias do problema original no valor calculado no problema relaxado. Neste modelo não foram consideradas misturas e propriedades para os tanques de armazenamento.

Em [15] é apresentado um trabalho onde os autores propuseram uma abordagem com base na teoria de controle de processos para o *scheduling* de curto-prazo. A metodologia usada, modelada a partir da rede de Petri, parte da consideração de que o *scheduling* é um conjunto de decisões de operação que levam a planta de um estado a outro, onde cada decisão de operação atua como um comando do controlador. Se o novo estado infringe alguma restrição tornando o *scheduling* inviável, ele é chamado de “estado inseguro” e a decisão de operação que o originou não pode ser realizada. A solução é constituída de várias decisões de operação que mantem a planta entre “estados seguros”. Estudo realizado validou a metodologia proposta, que trabalha com modelagem de tempo contínuo, contempla a representação de tempo de preparação em tanques, mas assume a existência de, no mínimo, 3 tanques de carga para cada unidade de destilação e penaliza fortemente a mistura de petróleo nos tanques. O trabalho demonstrou que a quantidade e a capacidade de tanques de carga, o

volume de diferentes óleos presente neles, a vazão do duto e a carga de processamento afetam a definição do “estado seguro”.

Em [18] é apresentado um trabalho no qual os autores modelaram o recebimento de óleo nos tanques da refinaria através de oleoduto conectado diretamente a navios. Foram modelados balanços volumétricos nos dutos e tancagem, bem como balanço por componente em cada tanque. Foi utilizada representação contínua do tempo e planos cortantes para diminuir o tempo de processamento. No entanto, a unidade de processo define a receita de óleos que ela demanda, o que nem sempre é possível na realidade operacional.

É consenso entre diversos autores que a representação do tempo é um dos grandes desafios para a representação matemática precisa dos problemas de *scheduling* [6,7,13]. Modelos com representação discreta são relativamente fáceis de implementar e garantir a integridade do balanço volumétrico. No entanto, para que tenham acurácia na representação das atividades é necessária a definição de um intervalo de tempo pequeno, o que leva à geração de muitos intervalos no período de *scheduling* e, conseqüentemente, a modelos muito grandes e computacionalmente demorados para resolver. Já os modelos com representação contínua do tempo são menores porque tem liberdade de definir a duração de cada intervalo de atividade. No entanto, seu desafio está em pré-determinar o número de intervalos que o horizonte de programação terá.

Outro grande desafio é a capacidade de tratar variações de composição e propriedades das misturas de petróleo. A modelagem destas características implica em uma sobrecarga do equacionamento, pela introdução de não-linearidades, que compromete o desempenho computacional dos modelos matemáticos [6,16,18].

Em [19] é apresentado um trabalho no qual os autores fizeram uma avaliação do estado da arte em métodos de otimização para problemas de programação de processos em batelada considerando as questões de representação de tempo e eventos, balanço material e função objetivo. Uma avaliação sobre o uso de métodos matemáticos em problemas reais da indústria também é feita e, a partir da qual, se afirma que essas técnicas podem ser usadas em conjunto com métodos heurísticos e técnicas de decomposição, de forma a manter o número de variáveis de decisão do modelo dentro de um nível aceitável. Este trabalho faz menção a uma linha de estudos que envolvem modelos híbridos para resolução de problemas de *scheduling*.

Em [20] é apresentado um trabalho no qual os autores propuseram um sistema híbrido para resolução do *scheduling* combinando MILP e um “sistema

especialista”, uma das abordagens de inteligência artificial, baseado na construção de regras extraídas de uma base de dados operacional. Para uma dada especificação de óleo a ser processado, definida no planejamento, a formulação de otimização matemática determinou a quantidade de óleo que deveria ser movimentada entre os grupos: tanque de armazenamento–tanque de carga e tanque de carga-unidade de destilação. As condições da movimentação entre esses grupos foram determinadas pelas regras especialistas de forma que o número de transições fosse minimizado. Em outras palavras, as decisões quantitativas foram tomadas pelo modelo MILP, enquanto as qualitativas ficaram a cargo do sistema especialista. Neste estudo a programação de petróleo foi representada como vários problemas modulares: descarga de navio, transferência entre os tanques de armazenamento e de carga e, por último, carga das unidades de processo. Cada módulo foi tratado com autonomia, com seus objetivos específicos e algoritmo mais eficiente para o problema. Por exemplo, no módulo de descarga de navio, cujo objetivo era descarregar o petróleo o quanto antes, o sistema especialista decidiu qual seria o tanque de armazenamento usado com base no inventário (volume e composição) da tancagem disponível. Desta forma, a otimização é de cada módulo e não, necessariamente, da solução completa.

Em [21] é apresentado um trabalho no qual os autores desenvolveram um modelo híbrido usando AG e MILP. O primeiro nível de decisão ficava a cargo do AG e consistia em determinar a sequência de atividades a serem executadas. O segundo nível consistia em determinar todas as demais variáveis de decisão através da solução de um modelo determinístico. A proposta por trás dessa divisão era retirar do modelo determinístico algumas decisões representadas por variáveis inteiras que oneram significativamente este tipo de representação. A representação foi validada para o *scheduling* incluindo descarga de navio, tanques de armazenamento e tanques de carga, além de 2 unidades de destilação. Não foram mencionadas questões de qualidade ou composição. Os resultados dessa abordagem foram confrontados com a solução do mesmo cenário obtida usando modelagem exclusivamente determinística com tempo discreto, onde se demonstrou que as duas modelagens chegaram à mesma função objetivo, com o modelo híbrido tendo melhor desempenho computacional.

Há alguns anos observa-se em todo o mundo o aumento do número de aplicações de sistemas dotados de inteligência artificial e, na indústria de petróleo não foi diferente, podendo-se encontrar alguns estudos nessa linha.

Em [8] é apresentado um trabalho no qual o autor desenvolveu um modelo baseado em algoritmos genéticos usando coevolução cooperativa. Nesta proposta foi adotada a decomposição do problema em duas partes atribuindo a responsabilidade a duas populações. Uma população indicou quando a atividade deveria ser realizada e a outra indicou quais os recursos utilizados para sua execução. A população responsável pela definição das tarefas possuía duas estruturas: uma lista de prioridades, que definia o cromossomo sobre o qual a evolução acontece, e um grafo de precedências que define que tarefas devem ser realizadas antes de outras. A planta utilizada no estudo foi uma versão simplificada do esquema real de uma refinaria, o recebimento de petróleo era feito por oleoduto, excluído do escopo de decisão da refinaria. Tendo como referência casos reais de programação de produção, não foram representadas atividades que já estivessem acontecendo no início do horizonte de programação e que, portanto, não poderiam ser interrompidas ou alteradas.

Em [3] é apresentado um trabalho no qual o autor propôs um modelo baseado em algoritmos genéticos onde o cromossomo tinha três segmentos: o primeiro, baseado em ordem, determinava a sequência de tarefas a ser realizada, os demais estabeleciam os recursos (equipamentos) de origem e destino a serem utilizados, dependendo da tarefa. O escopo tratou o *scheduling* da descarga de navios, movimentação terminal-refinaria através de oleoduto e carga de duas unidades de destilação atmosférica. A função de avaliação multiobjetivo considerou critérios de atendimento de processamento da destilação, movimentação do duto, não atraso em descarga de navios, minimização de trocas de tanque entre atividades, entre outros. A representação indireta do cromossomo deixou a carga da etapa de decodificação a submissão das tarefas aos testes de atendimento de restrições. A representação do tempo foi tratada indiretamente a partir da definição de um lote mínimo para realização de cada atividade. Para aumentar a probabilidade de fazer todas as tarefas necessárias ao horizonte de programação, o tamanho do cromossomo foi definido a partir do número de tarefas obtidas dividindo-se o volume total que poderia ser transferido entre todas as áreas do escopo pelo referido lote mínimo. O tamanho do cromossomo foi uma das dificuldades enfrentadas na evolução. Restrições de qualidade de carga foram tratadas como penalizações de não atendimento ao objetivo de manutenção da carga da unidade. Também não foram contempladas atividades que já estivessem acontecendo no início do horizonte de programação.

Em [9] é apresentado um trabalho no qual os autores propuseram um modelo baseado em AG. O escopo tratou o *scheduling* das áreas de óleo combustível e produção de asfalto. Foram avaliadas as representações direta e indireta do cromossomo. Na representação direta, o tempo foi discretizado e cada gene representava a atividade executada no intervalo de uma hora. Consequentemente, para um dado cenário, todos os cromossomos da população tinham o mesmo tamanho. O cromossomo foi construído com base na demanda de cada produto e maximizava a produção destes em cada intervalo de tempo. Na representação indireta cada gene estava associado à demanda de um produto. O objetivo era encontrar uma sequência de atividades que fosse viável para essa produção. A decodificação lidava com o tempo necessário para atender essa demanda, considerando quando deveria estar pronta e o tempo necessário para produção, desconsiderando inventário em tanque. A inviabilidade de alocar os recursos necessários para essa produção no tempo correto foi um dos grandes desafios dessas abordagens. Foram usados 4 cenários para avaliação dos modelos propostos, no entanto, nem todos atenderam aos objetivos de produção.

Em [10] é apresentado um trabalho no qual os autores desenvolveram uma integração entre a representação por grafo e algoritmos genéticos (*structure adapted genetic algorithm* - SAGA) para resolver o problema de *scheduling* de uma refinaria contemplando a descarga de navios diretamente nos tanques da refinaria e a carga das unidades de processo, modelando também a linha submarina pela qual o óleo transportado por navios de grande capacidade é movimentado até a refinaria. O cromossomo foi modelado de maneira a contemplar duas partes, chamados de subcromossomo estrutural e subcromossomo de pesos. O subcromossomo estrutural representava as estruturas operacionalmente viáveis definidas por subgrafos e contemplavam restrições do tipo: um tanque não pode receber e enviar ao mesmo tempo, ou duas parcelas de bombeio não podem ser descarregadas simultaneamente. O subcromossomo de pesos representava os pesos de cada aresta presente no subcromossomo estrutural. Estes pesos guardavam relação com a vazão da atividade que cada aresta representava em um dado período.

A interpretação dos subcromossomos foi feita para cada período onde, no subcromossomo estrutural estava o subgrafo que representava as atividades daquele período e no subcromossomo de pesos estavam as vazões dessas atividades. Restrições de vazão e capacidade são tratadas no balanço de massa do subcromossomo de pesos. Não foram tratadas restrições ou metas de

qualidade ou produção. O horizonte de programação foi tratado em períodos de tempo nos quais se assumiu que as vazões de todos os equipamentos não mudam dentro de cada um destes períodos.

A modelagem foi avaliada em 24 cenários com diferentes graus de dificuldade e comparado com uma modelagem usando AG+MILP, apresentando menor faixa de variação para o tempo de execução e função objetivo semelhante na maior parte dos casos.

3

Programação Genética

3.1.

Introdução

Desde a década de 1950 pesquisadores tentaram ensinar computadores a desenvolver automaticamente programas que fossem capazes de resolver problemas simples. Isto significa que a ideia de evoluir programas de computador é quase tão antiga quanto o computador em si. No entanto, este campo não foi explorado sistematicamente por mais de 40 anos [22].

A programação genética (PG) foi desenvolvida a partir de diversos trabalhos pioneiros das décadas de 1970 e 1980. O termo “programação genética” foi cunhado independentemente por John R. Koza e Hugo DeGaris em trabalhos publicados no começo da década de 1990 [11].

Em princípio, qualquer problema em inteligência artificial, aprendizado de máquina, sistemas adaptativos e aprendizado automatizado pode ser formulado e representado como um programa de computador [23]. A programação genética fornece um meio de sucesso de conduzir a busca deste desejado programa dentro do espaço de soluções [24].

Quando a definição de Koza para este termo começou a predominar, a maioria dos pesquisadores passou a representar a evolução de estruturas de programas por árvores ou, mais especificamente, na forma de programas em linguagem LISP (*List Processing*) [11]. Entretanto, atualmente entende-se por programação genética toda forma evolutiva de indução de programas [25].

A PG faz parte do paradigma da computação evolutiva (CE) que, por sua vez, é composta por algoritmos probabilísticos inspirados no princípio de evolução das espécies, sendo os algoritmos genéticos (AGs) sua forma mais popular. Ao contrário de técnicas determinísticas, em que somente é possível ter uma solução em cada instante de tempo, nas técnicas de CE uma população é iniciada com um conjunto de soluções aleatórias. Cada indivíduo (ou cromossomo) representa uma solução completa para o problema. A pressão seletiva faz com que, a cada geração, sejam gerados indivíduos iguais ou melhores aos melhores da geração anterior [22].

A característica de métodos baseados em CE é que estes permitem sua aplicação a uma variedade de problemas mesmo quando não se sabe como a solução deve parecer. De fato, este paradigma foi bem sucedido em várias aplicações, produzindo resultados que são considerados *human-competitives*, pois atendem critérios estabelecidos para aceitação de aprendizado de máquina, resultando em diversas patentes como novas invenções [24]. Alguns exemplos de aplicações são: projeto de filtros de circuitos digitais, projetos de circuitos de computadores quânticos, redução de ruídos de imagens, projetos de antena LAN para comunicação sem fio e mineração de dados [24,25]. No entanto, deve estar claro para quem desenvolve um estudo com técnicas de CE, que o sucesso nunca é garantido [22,26].

A Programação Genética é uma técnica de computação evolutiva para evoluir processos computacionais, como programas de computador. Nesta técnica os cromossomos são tratados como programas escritos em um subconjunto ou variação de uma linguagem de programação convencional [27].

O desenvolvimento de qualquer sistema de PG para resolver um problema implica na definição de algumas informações importantes: conjunto de terminais, conjunto de funções, seleção e avaliação.

O algoritmo básico de uma PG, de forma geral, é equivalente ao dos AG: a busca é feita avaliando-se iterativamente a aptidão dos indivíduos em uma população e aplicando-se operadores genéticos aos indivíduos, de forma a explorar diferentes áreas promissoras do espaço de busca. Em PG, um indivíduo tem sua aptidão avaliada pela execução do programa gerado e pelo acesso à qualidade de sua saída resultante.

3.2. Conceitos

3.2.1. Terminais e Funções

Os terminais e as funções são as primitivas com as quais um programa em PG é construído. Em linhas gerais, os terminais fornecem um valor ao sistema enquanto que as funções processam um valor que esteja no sistema. Juntos, funções e terminais formam os nós [27]. Apesar de esta terminologia ter surgido da representação de programas por árvores, seu uso também foi estendido às estruturas lineares. As representações mais comuns serão apresentadas na seção 3.3.

O conjunto de símbolos terminais é composto pelas entradas do programa, pelas constantes fornecidas e, algumas vezes, pelas funções sem argumentos executadas pelo programa. São chamados terminais ou folhas porque encerram um fluxo de execução em uma árvore. De fato, há um terminal no final de cada ramo de um cromossomo em árvore.

O conjunto de funções é composto pelas funções disponíveis para o sistema de PG. Este conjunto é específico à aplicação e selecionado visando adequar-se ao domínio do problema. A variedade de funções disponíveis é muito grande, podendo ser constituída, por exemplo, por funções “booleanas”, aritméticas, trigonométricas etc., e por sentenças condicionais, de laço etc. Muitas vezes é necessário incluir nos conjuntos primitivos funções desenvolvidas especialmente para o domínio do problema. Por exemplo, para um programa que movimenta um robô, será necessário incluir no conjunto funções que representem ações como: mover, girar, parar, etc. [28].

As funções e os símbolos terminais devem ser capazes de representar uma solução para um dado problema. Esta característica é chamada propriedade de suficiência. Se, por exemplo, um conjunto de funções for constituído apenas pela função de soma, ele provavelmente não resolverá muitos problemas. No entanto, deve-se ter cuidado em não usar um conjunto de instruções muito grande, pois aumenta o espaço de busca e pode, às vezes, tornar a busca por uma solução mais difícil. Ou seja, é importante uma abordagem sensata ao se escolher o conjunto de instruções [25].

3.2.2. Avaliação e Seleção

O conjunto de terminais e o conjunto de funções definem, indiretamente, o espaço de busca que a PG irá explorar. Isto inclui todos os programas que podem ser construídos pela combinação dos conjuntos. No entanto, ainda é necessário orientar a evolução para encontrar os programas que representam uma boa solução, o que é feito através da seleção de indivíduos e definição da função objetivo.

O sistema de PG deve escolher quais membros da população serão submetidos aos operadores genéticos. Ao fazer esta escolha, a PG implementa uma das mais importantes partes do seu modelo de aprendizado evolucionário: a seleção baseada em aptidão [27].

A métrica de avaliação da PG é chamada de função de avaliação da aptidão e representa a expressão que permite ao algoritmo reconhecer sua proximidade à solução ótima, sinalizando o quão bem um programa aprendeu a prever as saídas a partir das entradas, ou seja, as características do domínio de aprendizagem. Funções de avaliação são específicas de cada problema e representam o que é uma boa solução para este problema. Podem ser medidas de várias maneiras, como por exemplo: erro entre um valor medido e desejado, custo/lucro, tempo, etc. [28]. Seu objetivo é oferecer uma medida de aptidão de cada indivíduo na população corrente, indicando a habilidade do indivíduo em questão de solucionar o problema. Os valores das funções de avaliação de todos os indivíduos irão dirigir o processo de busca. Indivíduos com maior valor de função de avaliação têm maior probabilidade de persistirem na população tanto na sua forma direta como através de seus descendentes.

A maneira pela qual a função de avaliação afeta a seleção dos indivíduos para os operadores genéticos é considerada como sendo o algoritmo de seleção da PG [22]. Um mecanismo de seleção é caracterizado pela pressão seletiva ou intensidade de seleção que o mesmo introduz no algoritmo genético. O operador de seleção é um componente essencial de algoritmos genéticos. A literatura identifica cinco principais mecanismos de seleção: proporcional, por torneios, com truncamento, por normalização linear e por normalização exponencial. Há inúmeros tipos de algoritmos de seleção usados em PG, sendo mais comumente utilizada a seleção por torneio [28].

3.2.3. Algoritmo Básico

Há duas formas de se conduzir uma execução de PG: abordagem com gerações e abordagem *steady-state* (ou “sem gerações”). Na PG com gerações, a cada ciclo, toda uma nova geração é obtida a partir da geração anterior. A nova geração substitui a antiga e o ciclo continua. Na PG *steady-state*, não há intervalos fixos de gerações, mas sim um fluxo contínuo de indivíduos encontrando-se, cruzando-se e produzindo descendentes. Os descendentes substituem indivíduos existentes na mesma população.

Os ciclos de execução dos diferentes tipos de PG são similares e independem da representação e dos operadores. O algoritmo básico de PG segue os seguintes passos, também apresentados na Figura 3:

1. Geração de uma população inicial pela combinação aleatória de terminais e funções;
2. Iterativamente, execução dos passos deste item até que um critério de terminação seja satisfeito:
 - a. Execução total ou parcial dos programas da população (dependendo da modelagem desenvolvida) e determinação dos da aptidão.
 - b. Preservação do melhor indivíduo
 - c. Seleção de indivíduos e aplicação os operadores evolutivos
 - d. Uso dos critérios de substituição da população.
3. Apresentação do melhor indivíduo como solução da evolução.

Figura 3 - Algoritmo básico de execução da PG.

3.3. Estruturas de Representação

Diversas representações para as estruturas de dados podem ser utilizadas em PG, como por exemplo: árvores, grafos, listas. A seleção de qual é a estrutura de dados mais adequada para a solução de um problema depende principalmente dos tipos de operações que serão realizadas e da linguagem dos indivíduos a serem evoluídos.

3.3.1. Programação Genética em Árvore

A estrutura original e mais difundida é a representação em árvores, organizadas sobre o conceito de nós criados a partir de funções e terminais [28]. A informação da estrutura de dados é organizada na forma de um conjunto de nós e cada nó também deve ter sua organização interna.

Assim como no AG, os operadores de cruzamento e mutação são frequentemente utilizados na PG e, no caso de árvores, devem ser aplicados sobre os nós, como exemplificado na Figura 4. No entanto, diversos operadores podem ser implementados e utilizados em PG.

Figura 4 – Representação em árvore e operadores [28]

Há alguns pontos importantes que devem ser considerados:

- A propriedade de fechamento, que é a garantia, através da preocupação com a elaboração dos terminais e funções, de que todos os nós tenham o mesmo tipo de dado de maneira que, após a aplicação de operadores, não sejam geradas inconsistências na estrutura.
- A definição do tamanho máximo do indivíduo, evitando seu crescimento descontrolado e consequente aumento de carga computacional.
- A aplicação de técnicas para evitar o fenômeno em que o crescimento do indivíduo não é compensado por melhora significativa no valor da função objetivo, conhecido como *bloat*.

3.3.2. Programação Genética Linear

Na programação genética linear a representação dos programas é uma sequência linear de instruções. Sua principal característica quando comparada à PG baseada em estruturas em árvores, é que são evoluídos programas em uma linguagem imperativa e não expressões de uma linguagem de programação funcional [23].

Há algumas vantagens no uso de PG linear. Dentre elas, estão dois fatos: o primeiro é que quase todas as arquiteturas de computador representam programas de forma linear com as instruções sendo executadas em intervalos de tempo consecutivos e o segundo é que não é natural para computadores executar programas na forma de árvore, o que significa que compiladores e interpretadores precisam fazer parte dessa arquitetura [28]. Além destas características que refletem em velocidade de execução, uma estrutura linear simples favorece sua rápida análise e este tipo de representação também facilita a leitura do indivíduo e remoção de qualquer código não efetivo (*introns*) contido nele [22].

Cada programa é uma sequência de genes onde cada gene representa uma instrução, tal como o mostrado na Figura 5. O número de instruções pode ser fixo, significando que todos os programas da população têm o mesmo comprimento, ou variável, o que significa que diferentes indivíduos podem ter diferentes tamanhos. Os operadores evolutivos comumente utilizados são de cruzamento e mutação, sendo aplicados de forma a garantir a integridade das instruções.

Figura 5 - Exemplo de estrutura de cromossomo da PG Linear

3.3.3. Programação Genética Orientada à Gramática

A utilização de sistemas que permitam gerar automaticamente programas em linguagens arbitrárias é desejável pela conveniência de se escolher uma linguagem de programação específica para um determinado domínio de aplicação. No período que se seguiu à introdução de PG, vários sistemas foram desenvolvidos com este propósito, sendo que a maioria utiliza a notação BNF

(*Backus-Naur Form*) para gramáticas livres de contexto, de forma a especificar a sintaxe da linguagem desenvolvida. Uma BNF descreve estruturas gramaticais admissíveis na construção de uma linguagem através da quádrupla: $\{S, N, T, P\}$, onde S denota o símbolo inicial, N denota o conjunto de símbolos não terminais, T denota o conjunto de símbolos terminais e P são as produções [29]. Pela análise da Figura 6 é possível compreender a sintaxe utilizada. S é o símbolo inicial. Se estiver definido entre os caracteres “<” e “>”, são símbolos não terminais e, se estiver definido entre aspas são símbolos terminais. O sinal “::=” denota a atribuição. As barras verticais separam as alternativas de construção do símbolo. Por exemplo, para a Figura 6, <op> pode assumir os valores “+” ou “-” ou “/” ou “*”. Por exemplo, a interpretação de <code> é substituí-lo por <code> ou <line> <code>. Uma linguagem $L(G)$ gerada por uma gramática G , definida por uma BNF, é o conjunto de todas as sentenças que podem ser derivadas a partir de S . Para se criar uma sentença em uma linguagem particular é necessário escolher quais produções serão utilizadas e em que sequência elas serão utilizadas, função esta destinada ao algoritmo evolutivo [29]. A Figura 7 apresenta um exemplo de produção que poderia ser gerada, na qual a função “myfunction (float x)” foi desenvolvida a partir da BNF apresentada na Figura 6, em linguagem C.

S	::= <func>
<func>	::= <header>
<header>	::= float "myfunction(float x) {" <body> "}"
<body>	::= <declarations> <code> <return>
<declarations>	::= "float a"
<return>	::= "return(a) "
<code>	::= <line> <line> <code>
<line>	::= <var> "=" <expr>
<expr>	::= <expr> <op> <expr> <pre-op> "(" <expr> ")" <var>
<op>	::= "+" "-" "/" "*"
<pre-op>	::= "sin" "cos" "tan" "log"
<var>	::= "x"

Figura 6 - Exemplo de estrutura BNF para definir sintaxe de uma linguagem específica

```
float myfunction(float x)
{
    a = x + sin(x)
    return(a)
}
```

Figura 7 – Exemplo de produção possível

3.4.

Programação Genética Linear com Inspiração Quântica

Uma abordagem proposta por Moore [30] é a computação com inspiração quântica, cujo objetivo é desenvolver algoritmos clássicos que tirem proveito dos paradigmas da física quântica, como a superposição de estados, para melhorar o desempenho dos mesmos na solução de problemas.

No campo da computação evolucionária, a introdução de algoritmos evolutivos com inspiração quântica (AEIQs, em inglês referenciado QEAs *quantum-inspired evolutionary algorithms*) por Han e Kim pode ser considerada a primeira aplicação de maior sucesso da inspiração quântica [31]. O uso desta abordagem acontece essencialmente sobre os operadores evolutivos.

Em um computador clássico, o bit é a menor unidade de informação, podendo assumir os valores 0 ou 1. Em um computador quântico, a unidade de informação básica, chamada de *q-bit*, pode assumir os estados $|0\rangle$, $|1\rangle$ ou uma superposição dos dois estados. Esta superposição de dois estados é uma combinação linear dos estados $|0\rangle$ e $|1\rangle$ e pode ser representada pela eq. 3.1:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (3.1),$$

onde $|\psi\rangle$ representa o estado do *q-bit*, α e β são números complexos que representam a probabilidade dos estados 0 e 1 quando o *q-bit* for observado (ou medido) [32].

A programação genética com inspiração quântica usada neste trabalho é desenvolvida com base no trabalho [11]. Entretanto, nesta dissertação o modelo é usado para evoluir programas que proponham uma solução para a programação de petróleo em uma refinaria através do uso de uma linguagem imperativa arbitrária, específica para o domínio tratado. Sendo assim, é necessário definir uma gramática e sua sintaxe para que as instruções sejam geradas. Os indivíduos clássicos possuem uma representação linear onde cada gene clássico representa uma instrução, enquanto que os indivíduos quânticos são compostos por genes com uma estrutura semelhante a árvore e construídos segundo as orientações da gramática definida. Diante destas características, pode-se observar que esta dissertação propõe uma modelagem onde referências à programação genética linear, à programação genética orientada à gramática e à evolução com inspiração quântica coexistem e por isso, a modelagem proposta é nomeada como Programação Genética Linear Orientada à Gramática e com Inspiração Quântica (PGLOGIQ). Ao contrário dos

operadores de cruzamento e mutação convencionalmente utilizados, aqui a evolução fica a cargo do operador quântico que trabalha com as probabilidades de seleção das instruções e seus argumentos em cada gene do indivíduo. Neste trabalho, assim como em [11], a unidade básica de informação, que é uma generalização do bit quântico, é tratada como *qudit*. O modelo usado é detalhadamente apresentado no capítulo 4.

A PGLOGIQ é baseada nos indivíduos quântico e clássico que representam, respectivamente, o genótipo e o fenótipo de uma estrutura evolutiva, além do operador quântico e dos conjuntos de funções e de terminais. As seções 3.4.1 a 3.4.4 apresentam os papéis destas entidades. No capítulo 4 são apresentadas novamente estas entidades, dentro do contexto do trabalho desenvolvido nesta dissertação.

3.4.1. Conjunto de Funções e Terminais

O conjunto de funções estabelece as instruções características do problema tratado. O programa deve saber interpretá-las e executá-las para compor a solução final. Por exemplo, no caso estudado, para representar atividade de programação de petróleo, foram criadas instruções que implementam como alimentar uma unidade de destilação com somente um tanque de carga ou utilizando tanque de injeção. Além das funções específicas, também podem ser criadas auxiliares que visam melhorar o desempenho do modelo pelo aprendizado de correlações entre variáveis internas do programa, como por exemplo: $auxiliar_1 = volumeTQ - volumeProcessado$ ou $auxiliar_1 = auxiliar_1 * duraçãoAtividade$.

O conjunto de terminais estabelece os valores que uma função pode utilizar, como por exemplo, a função que faz com que a unidade de processo tenha carga, precisa conhecer o nome do tanque e da unidade de processo utilizados. Esses valores são elementos terminais.

3.4.2. Indivíduo Quântico

O indivíduo quântico representa a superposição de todos os programas possíveis de serem gerados pela superposição no espaço de busca definido [33]. É constituído por uma lista de estruturas denominadas genes quânticos.

Um gene quântico é composto por um *qudit* de função (QF) e pelos *qudits* de terminal (QT) correspondentes a essas funções. Um gene quântico representa a superposição de todas as instruções predefinidas pelos conjuntos de funções e terminais para o espaço de busca definido.

3.4.3. Indivíduo Clássico

É o cromossomo gerado a partir da observação do indivíduo quântico e, a partir do qual o programa que representa a solução é finalmente gerado.

Cada gene do indivíduo quântico é observado para gerar cada gene do indivíduo clássico, conseqüentemente, cada gene clássico representa uma instrução a ser executada. Como a relação entre gene clássico e gene quântico é de um para um pode-se concluir que ambos os indivíduos tem o mesmo comprimento.

3.4.4. Operador Quântico

O operador quântico proposto na PGLOGIQ atua diretamente sobre as probabilidades de um *qudit*, assim como na PGLIQ [11].

O indivíduo clássico de maior aptidão fornece o material para orientar o incremento das probabilidades dos *qudits* de cada gene do indivíduo quântico. Por exemplo, se um determinado gene do melhor indivíduo clássico tem a instrução que representa *Carga na unidade U-A* usando tanque *T-1*, o gene quântico que ele representa terá as probabilidades associadas às escolhas dos terminais que representam tanque *T-1*, unidade *U-A* e instrução *Carga na Unidade* incrementadas no tamanho de um passo *s*, arbitrado para o modelo. Após o incremento, os valores de todas as probabilidades são alterados para atender à condição de normalização.

3.5. Motivação

Podem ser encontrados na literatura alguns trabalhos que falam sobre o uso de PG em problemas de *scheduling*. Por exemplo, em [34] a PG é usada para propor o *schedule* onde os tempos de execução de cada atividade são conhecidos, bem como data limite de finalização e as relações de precedência entre as atividades. Em [35] os autores utilizam PG para gerar heurísticas de

scheduling em um problema do tipo *job shop*. No entanto, não foram encontradas referências que aplicassem PG para resolver problemas de *scheduling* para uma refinaria de petróleo.

Para outras aplicações, alguns trabalhos apresentam o bom desempenho de algoritmos evolutivos baseados em inspiração quântica quando comparados com sua contraparte clássica. Em [32] é apresentado um algoritmo evolutivo com inspiração quântica para otimização numérica que é inspirado no princípio de múltiplos universos da física quântica. Este AEIQ possui representação por números reais e apresenta um tempo de convergência menor para problemas *benchmark* quando comparado com algoritmos convencionais. Em [11] é apresentado um algoritmo evolutivo baseado em PGLIQ e para avaliar seu desempenho são utilizados testes de regressão simbólica e classificação binária, comparativamente ao AIMGP (*Automatic Induction of Machine Code by Genetic Programming*), considerado o modelo de PG mais eficiente para evolução de código de máquina [36].

Em suma, o uso de programação genética linear com inspiração quântica neste trabalho foi motivado por diversos fatores, tais como:

- PG Linear pode ser implementada com variações de representação que permitam o uso de uma linguagem específica para o problema proposto e uso de uma estrutura BNF para definição da sintaxe das suas instruções [36].
- Com esta flexibilidade, é estimulado o desenvolvimento de uma gramática e sua sintaxe de forma a gerar um conjunto de instruções que representem atividades familiares ao dia a dia da programação de petróleo na refinaria;
- Indivíduos clássicos gerados são estruturas lineares de fácil leitura e interpretação.
- Bom desempenho apresentado por algoritmos evolutivos com inspiração quântica nas aplicações aos quais foi submetido.

4 Modelo Proposto

O primeiro passo no desenvolvimento deste estudo foi representar a programação de petróleo de uma refinaria cuja autonomia de decisão estivesse desde a descarga de navios em terminal marítimo dedicado até as unidades de destilação atmosférica.

Na sequência, com base na observação das atividades realizadas pela programação de produção, foi proposta uma linguagem específica de domínio para representar as atividades da planta de processo estudada.

Por último, foi construído um modelo utilizando programação genética cujos cromossomos foram compostos de genes que contivessem, individualmente, as atividades de programação representadas. Para a evolução dos cromossomos foi utilizada uma abordagem por inspiração quântica.

4.1. Configuração da Refinaria Modelada

Esta seção tem por objetivo detalhar a planta estudada, tanto em termos de equipamentos e alinhamentos, quanto em termos das atividades comuns da programação de produção. De forma simplificada, para garantir óleo às unidades de processamento é necessário descarregar óleo no terminal marítimo, movimentá-lo até a refinaria e, por último, compor carga para as unidades de destilação atmosférica.

4.1.1. Representação Física

A refinaria apresentada foi baseada em uma refinaria real, entretanto simplificações e tratamentos matemáticos foram realizados de forma a descaracterizar os dados reais e manter a validade dos resultados para o estudo proposto.

A representação física foi decomposta em áreas lógicas de forma a indicar o sequenciamento das movimentações de óleo. Dessa forma, o modelo possui

as áreas: “Navios”, “LinhaSubmarina”, “Terminal”, “Duto” e “Refinaria”, detalhados a seguir.

A Figura 8 apresenta uma representação esquemática do escopo de *scheduling* proposto neste trabalho, incluindo navios, terminal e refinaria.

Figura 8 – Representação do problema proposto

Navios:

O suprimento de óleo para a refinaria é realizado através de navios petroleiros que atracam próximo ao terminal. O navio é amarrado, a mistura de petróleo bombeada através de uma linha submarina e recebido na tancagem do terminal.

Apenas um navio pode estar amarrado para descarregar em cada instante *t* do horizonte de programação.

O navio pode conter um ou dois compartimentos e, neste trabalho, estes compartimentos foram tratados como tanques do navio, denominados *tanqueCN*. Cada *tanqueCN* contém apenas uma mistura de petróleo. Não há limitações do modelo para o número de navios que podem ser contemplados no cenário.

Por exemplo, as configurações apresentadas na tabela 1 foram previstas.

Navios	tanqueCN1	tanqueCN2
Navio1	100% P1	Não possui
Navio2	100% P2	100% P3
Navio3	100% P1	25%P4/35%P7/40%P3
Navio4	28%P12/53%P2/19%P8	35%P1/65%P2

Tabela 1 – Exemplos de diferentes configurações de navios

Linha Submarina:

A linha submarina é um duto com volume de 11.666 m^3 . Transporta óleo do navio para os tanques do terminal e, na modelagem proposta, é assumido que está sempre completamente cheia de uma única mistura de petróleo. Neste trabalho, ela é denominada LS1.

Terminal:

O terminal possui quatro tanques dedicados exclusivamente para movimentar petróleo. Estes tanques não podem operar em pulmão, ou seja, não é possível que o tanque esteja recebendo o descarregamento de navio e bombeando para a refinaria em um mesmo instante de tempo t .

Todos os tanques possuem alinhamento para receber da linha submarina, bem como todos podem bombear para o duto.

É assumido que a mistura em tanque é perfeita, ou seja, não há gradientes de concentração ou propriedade em todo o volume do tanque. Cada unidade de volume recebida no tanque é, no mesmo instante, perfeitamente misturada ao que já está contido nele.

Além da limitação de capacidade total de petróleo que cada tanque pode armazenar, há também a restrição com respeito ao lastro. O lastro é definido como um volume mínimo abaixo do qual o tanque não pode operar, ou seja, para efeitos de movimentação, o tanque funciona como se estivesse vazio. Entretanto, este volume não pode ser desprezado para os cálculos de composição do tanque e propriedades da mistura. O volume resultante da diferença entre o total de petróleo contido no tanque e o lastro é chamado volume útil ou volume operacional.

A tabela 2 apresenta as características dos tanques do terminal.

Tanque	Lastro (m^3)	Capacidade (m^3)
TqT01	6.093	52.527
TqT02	6.938	61.338
TqT03	7.393	62.763
TqT04	7.820	63.228

Tabela 2 - Lastro e capacidade total dos tanques do terminal

Duto Terminal-Refinaria:

O duto possui volume de 19.000 m³ e é responsável por transportar petróleo do terminal para a refinaria. Está sempre cheio de óleo, mas pode conter parcelas com diferentes misturas de petróleo dentro dele.

A vazão de movimentação do duto é uniforme ao longo de seu comprimento e determinada pelo item de petróleo que está sendo bombeado de um dos tanques do terminal.

Neste trabalho, ele é denominado DUTO1.

Refinaria:

A refinaria possui quatro tanques dedicados exclusivamente para movimentar petróleo. Estes tanques não podem operar em pulmão, ou seja, não é possível que o tanque esteja recebendo corrente do duto e compondo carga para as unidades de destilação em um mesmo instante t .

Todos os tanques possuem alinhamento para receber óleo do duto, bem como todos podem compor carga das unidades de destilação.

Nos tanques da refinaria também é assumida mistura perfeita, assim como no terminal. Também são válidos os conceitos de capacidade total, lastro e volume operacional, explicados para os tanques do terminal.

A tabela 3 apresenta as características dos tanques da refinaria.

Tanque	Lastro (m ³)	Capacidade (m ³)
TqR01	8.339	62.822
TqR02	5.193	37.268
TqR03	8.369	62.796
TqR04	8.827	63.255

Tabela 3 - Lastro e capacidade total dos tanques da refinaria

Unidades de Destilação Atmosférica (UDA):

É composta por duas torres de destilação UDA-01 e UDA-02, com vazões máximas de 360 m³/h e 700 m³/h, respectivamente. A vazão das unidades é um dado de entrada e não faz parte das decisões do modelo.

As UDAs operam independentemente, ou seja, podem ter vazões diferentes, composição de cargas diferentes e perfis de rendimento e propriedades diferentes.

Cada UDA pode ser alimentada por um ou dois tanques da refinaria simultaneamente. Neste trabalho, quando a UDA é alimentada por dois tanques,

o que estiver contribuindo com maior vazão para compor a carga total da unidade é chamado de tanque principal, ou tanque base, e o tanque que completa esta demanda é chamado de tanque de injeção.

O mesmo tanque pode compor a carga das duas unidades simultaneamente durante um mesmo período de tempo.

Para que o tanque possa compor a carga da UDA é preciso garantir que foi respeitado o tempo de preparação do tanque. Este tempo é o necessário para que, após um recebimento de óleo bombeado do terminal, seja decantada parte da água e do sal presentes na mistura e posterior drenagem dessa salmoura do tanque. Se este procedimento não for feito e a salmoura chegar aos equipamentos de processo posteriores (dessalgadora e torre de destilação) em concentração maior que o permitido acarretará em instabilidades operacionais que comprometem o adequado funcionamento das unidades. Este tempo de preparação pode variar em função de características da mistura, da capacidade do tanque e, principalmente, do percentual dessa salmoura presente no óleo. O tempo usualmente encontrado e também utilizado neste trabalho é de 24 horas.

4.1.2. Atividades de Programação

Esta seção explica as atividades sobre as quais o programador de produção deve tomar decisões e explicita quais informações são dados de entrada em cada etapa.

4.1.2.1. Descarregamento de Navio para o Terminal

Quando o programador recebe informação sobre cada navio que irá atracar no terminal, o conjunto de dados conhecido contempla a data/hora de chegada do navio, volume e composição de cada mistura de petróleo que ele esteja transportando.

Se um navio informa sua chegada ao terminal e seu descarregamento atrasa de forma a ultrapassar um tempo de espera permitido, a consequência é a penalização financeira da refinaria através do custo de sobreestadia. O tempo de espera permitido varia em função, por exemplo, do volume transportado e do tipo de contratação do navio em questão. É uma informação conhecida pelo programador de produção e, neste estudo, foram fornecidos como dado de entrada o horário de chegada de cada navio e a tolerância (em horas) para

descarregamento. Assim está representada a janela de tempo que um navio pode descarregar sem implicar em penalização por sobreestadia.

A ordem de descarregamento dos navios é seguida de acordo com a ordem de chegada desses ao terminal. Entretanto, o programador deve decidir, para cada navio, qual será a sequência de descarregamento dos *tanques* (quando o navio tiver mais de um). Várias considerações devem ser feitas para essa decisão, como por exemplo, a necessidade de determinado tipo de óleo na refinaria em menor horizonte de tempo, a compatibilidade de mistura entre os óleos a serem descarregados e os tanques do terminal e também a composição e os valores de propriedade da mistura final após cada volume descarregado. Neste trabalho, a ordem dos navios respeita a data/hora de chegada ao terminal, mas cabe ao modelo decidir sobre a ordem de descarregamento das misturas de petróleo. Existe a viabilidade técnica de descarregar parcialmente uma mistura de petróleo, descarregar um volume da outra mistura e depois concluir a primeira. O conjunto de operações necessário para sua realização é muito complexo e raro. Desta forma, essa flexibilidade não foi contemplada na modelagem.

A vazão de descarregamento do navio tem variações em função das propriedades da mistura de petróleo que está sendo bombeada. Neste modelo, ela foi representada como um dado de entrada do sistema.

4.1.2.2.

Movimentação de Óleo do Terminal para a Refinaria

A movimentação de óleo do terminal para a refinaria é um conjunto de decisões onde, para cada uma, é necessário definir o volume a ser movimentado, o tanque do terminal de onde será retirado esse volume e o tanque da refinaria que irá recebê-lo.

As considerações feitas para essa decisão costumam envolver a necessidade de determinado tipo de óleo na refinaria, a disponibilidade dos tanques para a movimentação (um tanque que já esteja sendo usado em outra atividade não pode ser usado na movimentação), a compatibilidade de mistura entre o óleo movimentado e os tanques da refinaria e também a composição e os valores de propriedade da mistura final após cada volume movimentado.

A movimentação acontece através de um duto cujo volume deve ser considerado, o que significa que um volume que começa a ser bombeado do

terminal no instante t_1 começará a chegar ao tanque da refinaria aproximadamente 15 horas depois.

Por se tratar de um duto de grande capacidade (19.000 m^3) é comum que ele contenha parcelas de diferentes composições. A modelagem proposta limita a quantidade de parcelas que podem existir através de um dado volume mínimo que poderia ser movimentado e da flexibilidade de configuração de restrição associada ao tempo mínimo que um tanque do terminal deve permanecer bombeando para o duto. O modelo respeita a integridade do volume total do duto e a ordem das parcelas dentro dele. Isto significa que ao bombear, por exemplo, 5.000 m^3 de um tanque do terminal para o duto, o modelo irá destinar 5.000 m^3 da parcela mais próxima à extremidade no lado da refinaria para um tanque desta.

A vazão de movimentação dentro do duto é definida pela vazão da mistura de petróleo (item) que está sendo bombeado para o duto no terminal. Esta vazão tem variações em função das propriedades da mistura de petróleo. Neste estudo, a relação de vazão com propriedades do óleo não foi modelada, tendo sido assumida uma simplificação de que todos os itens são bombeados através do duto com a vazão constante, independente da composição.

O volume total e a composição inicial de cada tanque são informados no conjunto de dados de entrada. A parte evolutiva do modelo (otimização) é responsável pelas decisões de volume da movimentação e alocação dos tanques do terminal e da refinaria para cada atividade, enquanto o simulador de planta é responsável pelo cálculo atualizado do volume, composição e propriedade dos tanques envolvidos após cada movimentação.

Para definir o volume da movimentação, a heurística usada considera que o valor máximo possível é dado pelo menor dos valores entre: o volume disponível no tanque do terminal selecionado para a atividade; o espaço disponível no tanque da refinaria selecionado; e o volume da parcela do duto mais próxima da refinaria que será a recebida no tanque, como representado na eq 4.1:

$$VolMaxMovim = \min(VolDispTQT0x, EspaçoDispTQR0x, VolParcela) \quad (4.1)$$

A parte evolutiva do modelo decide usar este volume máximo ou um percentual dele. A Figura 9 apresenta uma representação esquemática da movimentação do duto e da definição de volume da atividade.

Figura 9 – Representação esquemática da movimentação do duto

4.1.2.3.

Carga das Unidades de Destilação

As decisões do programador de produção são orientadas pelo objetivo principal de manter continuamente as unidades de destilação operando na sua vazão de carga programada. Esta vazão pode sofrer oscilações características da variabilidade operacional, mas este nível de representação não é contemplado neste trabalho.

As unidades de destilação possuem modos de operação que representam ajustes de variáveis operacionais como, por exemplo, temperatura e pressão nos estágios de separação, taxa de refluxo circulante, dentre outras, e que se traduzem em perfis de rendimento e propriedades para os cortes destilados retirados da torre, tipicamente: gás liquefeito de petróleo (GLP), nafta leve (NL), nafta pesada (NP), querosene (Q), diesel leve (DL), diesel pesado (DP) e resíduo atmosférico (RAT).

É atribuição do programador, com base nas metas de produção estabelecidas no plano de produção, definir a vazão de carga e a campanha em que cada unidade de destilação operará diariamente. Estas informações são tratadas como dados de entrada do modelo.

Para manter as unidades operando, o programador de produção também decide sobre o uso da tancagem da refinaria. Várias considerações devem ser feitas para essa decisão como, por exemplo, volume disponível, composição e propriedades das misturas de petróleo em tanque de maneira a respeitar não somente a necessidade de produção como também limitações operacionais da torre, como a acidez total da mistura a ser processada.

Para definir a atividade de carga das unidades, fazem parte das decisões do modelo evolutivo: a seleção da unidade de processo; se um ou dois tanques são usados na carga; quais os recursos; e o volume da atividade. Para determinação deste volume, o algoritmo evolutivo otimiza entre o valor máximo possível para atividade, calculado com base no volume útil disponível nos tanques selecionados (base e injeção, se houver), ou uma fração deste.

Os arquivos de configuração e a parte de simulação implementada no modelo permitem que o otimizador conheça, para cada petróleo, os valores de suas propriedades (densidade, enxofre e acidez) e o perfil de rendimentos e propriedades dos cortes destilados que cada um terá na unidade de destilação atmosférica e campanha em que estiver operando.

4.1.3. Informações de Entrada e Decisões do Modelo

Seguindo o que foi discutido na seção 4.1.2, são apresentadas, na tabela 4, quais informações são dados de entrada do sistema e quais são decisões do modelo.

Neste trabalho a abordagem adotada para determinar o tempo das atividades é indireta, acontecendo através das decisões do modelo evolutivo sobre o volume das atividades. Uma vez que todas as vazões são conhecidas, o tempo é, então, determinado para os equipamentos envolvidos na atividade.

4.2. Linguagem Específica de Domínio

Linguagem Específica de Domínio (*Domain Specific Language – DSL*) é uma linguagem de programação ou linguagem de especificação que oferece, através de notações adequadas e abstrações, poder expressivo voltado e,

geralmente, restrito a um determinado domínio de problema, sendo, para tal, baseada nos conceitos relevantes e características desse domínio [38].

Um pré-requisito para o desenvolvimento de uma DSL é uma análise detalhada e estruturação do domínio da aplicação. Desta forma, para o desenvolvimento da DSL deste modelo, foram avaliados diversos cenários de programação de maneira a identificar as principais atividades executadas.

Como descrito na seção 4.1.2, estas atividades estão associadas ao descarregamento de navios, movimentação terminal-refinaria e carga de unidades de destilação. Para representá-las, foi implementada uma linguagem imperativa específica para o domínio do problema proposto. A lista de funções e argumentos que a constituem é apresentada na tabela 5.

Foi necessário formalizar esta linguagem através de uma sintaxe que garantisse a criação de instruções válidas pelo sistema. A notação escolhida para definir sua gramática é baseada na BNF. A Figura 10 apresenta a BNF para CEN05, que foi o cenário utilizado para a definição dos parâmetros do modelo evolutivo usado nesta dissertação. A interpretação com base na quádrupla apresentada na seção 3.3.3 demonstra que o símbolo <func> representa as funções da PG e pode assumir os valores NOP, RcvItemMax etc. NOP é o único elemento terminal do símbolo <func>, todas as demais funções possuem símbolos que remetem a outras atribuições. O símbolo <navioeTnq> representa o navio que está conectado à linha submarina e, dependendo do símbolo <navio1Tnq> ou <navio2Tnq>, qual é o *tanqueCN* que está descarregando deste navio. Os demais símbolos, como por exemplo, <tnqT>, <uda> e <itemProp>, definem os elementos terminais. Um exemplo de produção (*P*) desta estrutura é a instrução FeedCDUMax (TqR04, UDA-02), cuja interpretação é que o TqR04 é o único tanque na carga da unidade UDA-02 e que esta atividade é executada com volume máximo.

Nesta representação, o único símbolo variável entre os cenários do sistema foi o <navioeTnq>. Este símbolo concentra as informações do número de navios a ser descarregado no cenário e a quantidade de *tanqueCN* transportados em cada navio. Na Figura 10, “navio1” e “navio2” significam que CEN05 possui dois navios e os terminais vinculados, os símbolos <navio1Tnq> e <navio2Tnq>, indicam que o primeiro navio possui dois *tanquesCN* e por isso transporta duas misturas de petróleo diferentes e o segundo navio apenas um *tanqueCN* e, conseqüentemente, uma mistura de petróleo.

Estrutura	Dados
Dados de entrada (estacionários)	Lista de petróleos e suas propriedades (densidade, enxofre, acidez).
	Rendimento dos cortes de destilação para cada petróleo e as respectivas propriedades (densidade, enxofre, acidez) nas campanhas das unidades.
	Restrições (podem ser configuradas a qualquer momento, mas foram usadas as mesmas nos 5 cenários avaliados neste trabalho).
	Planta: Tanques do terminal e refinaria: lastro e capacidade total. Duto: capacidade volumétrica Representação origem-destino entre equipamentos.
Dados de entrada (para cada cenário)	Para cada navio: Data/hora de chegada e tolerância para descarga. Vazão de descarregamento. Para cada <i>tanqueCN</i> : volume e composição da mistura de petróleos.
	Inventário dos tanques do terminal e refinaria, duto e linha submarina (volume e composição da mistura de petróleos) no início do cenário.
	Vazão de bombeio do terminal para a refinaria.
	Modo de operação das unidades de destilação: vazão programada e campanha.
Decisões do modelo	Sequência de atividades a serem executadas.
	Descarregamento de navio: Tanque de origem do navio (<i>tanqueCN</i>). Tanque de destino no terminal. Volume da atividade (volume no navio, espaço disponível na refinaria ou fração).
	Movimentação Terminal – Refinaria: Tanque de origem no terminal. Tanque de destino na refinaria. Volume da atividade (volume no tanque de origem, espaço no tanque de destino, parcela dentro do duto ou fração).
	Carga de unidade de destilação: Usar somente um tanque de carga para dar carga alimentar a unidade de processo ou usar tanque de carga base e tanque de injeção. Tanque de carga base e, se for o caso, tanque de injeção. Percentual de participação volumétrica do tanque de injeção, se for o caso. Volume da atividade.

Tabela 4 - Estrutura dos tipos de dados utilizados no sistema

Função	Argumentos	Descrição
NOp		Não executa nenhuma atividade. É usada para controlar, indiretamente, o tamanho do indivíduo.
RcvItemMax	TanqueCN Linha submarina Tanque do Terminal	Transfere do navio para o terminal o volume máximo possível para a atividade o que equivale ao menor valor entre o espaço disponível no tanque do terminal e o tanqueCN.
RcvItemProp	TanqueCN Linha submarina Tanque do Terminal Proporção do Item	Similar à função RcvItemMax. A diferença é que o volume transferido é uma proporção do máximo.
FeedCduMax	Unidade de destilação Tanque de carga	Transfere óleo de apenas um tanque da refinaria para alimentar uma unidade de destilação. O volume máximo possível para a atividade, o que equivale a todo o volume útil disponível no tanque.
FeedCduProp	Unidade de destilação Tanque de carga Proporção de carga	Similar à função FeedCduMax. A diferença é que o volume transferido é uma proporção do máximo.
FeedCduMaxInj	Unidade de destilação Tanque de carga Tanque de injeção Proporção de injeção	Transfere óleo de dois tanques (base e injeção) da refinaria para alimentar uma unidade de destilação. O volume máximo possível para a atividade, o que é definido pela relação entre o volume útil disponível em cada um dos tanques e o percentual de injeção definido pela tarefa.
FeedCduPropInj	Unidade de destilação Tanque de carga Tanque de injeção Proporção de injeção Proporção de carga	Similar à função FeedCduMaxInj. A diferença é que o volume transferido é uma proporção do máximo.
PipeTransfMax	Tanque do terminal Tanque da refinaria	Transfere do terminal para a refinaria o volume máximo possível para a atividade, o que é definido pela relação entre o volume útil disponível no tanque do terminal, o espaço disponível no tanque da refinaria e o volume da parcela que será movimentada.
PipeTransfProp	Tanque do terminal Tanque da refinaria Proporção de volume do duto	Similar à função PipeTransfMax. A diferença é que o volume transferido é uma proporção do máximo.

Tabela 5 – Funções e argumentos desenvolvidos para a linguagem

```

<func> ::= "NOP" |
           "RcvItemMax" <navioeTnq> <tnqT> |
           "RcvItemProp" <navioeTnq> <tnqT> <itemProp> |
           "FeedCDUMax" <tnqR> <uda>
           "FeedCDUProp" <tnqR> <uda> <feedProp> |
           "FeedCDUMaxInj" <tnqR> <tnqInj> <injProp> <uda> |
           "FeedCDUPropInj" <tnqR> <tnqInj> <injProp> <feedProp> <uda> |
           "PipeTransfMax" <tnqT> <tnqR> |
           "PipeTransfProp" <tnqT> <tnqR> <pipeProp>

<navioeTnq> ::= "navio1" <naviolTanq> | "navio2" <navio2Tanq>
<naviolTanq> ::= "tanq1" | "tanq2"
<navio2Tanq> ::= "tanq1"
<tnqT>      ::= "TqT01" | "TqT02" | "TqT03" | "TqT04"
<tnqR>      ::= "TqR01" | "TqR02" | "TqR03" | "TqR04"
<tnqInj>    ::= "TqR01" | "TqR02" | "TqR03" | "TqR04"
<uda>       ::= "UDA-01" | "UDA-02"
<itemProp>  ::= "0.1" | "0.2" | "0.3" | "0.4" | "0.5" | "0.6" | "0.7" | "0.8" | "0.9"
<pipeProp>  ::= "0.2" | "0.4" | "0.6" | "0.8"
<feedProp>  ::= "0.2" | "0.4" | "0.5" | "0.6" | "0.8"
<injProp>   ::= "0.1" | "0.2" | "0.3" | "0.4" | "0.5"

```

Figura 10 – Estrutura BNF que representa a gramática da linguagem específica para o problema proposto.

4.3. Representação dos Indivíduos

Independentemente da técnica de modelagem utilizada para a otimização, ao se propor resolver problemas de *scheduling* quatro representações básicas devem ser feitas, sendo elas: seleção de um número de atividades a serem executadas dentro de um horizonte de tempo de modo a satisfazer certas exigências; alocação de recursos para execução destas atividades respeitando as restrições de disponibilidade destes; definição de uma sequência para as atividades selecionadas respeitando as restrições de precedência; e a determinação do instante de tempo da utilização dos recursos para as atividades [7].

Baseando-se na gramática desenvolvida para o estudo de caso, e compatível com este roteiro, foi proposto que cada indivíduo clássico representasse uma solução completa de *scheduling* e que cada gene clássico

representasse uma atividade de programação com recursos alocados e volume definido, dessa forma o número de atividades a serem executadas e seu sequenciamento estariam definidos.

A PGLOGIQ é baseada nas seguintes entidades:

- cromossomo do “indivíduo quântico”, que representa a superposição de todos os programas possíveis para o espaço de busca predefinido;
- cromossomo do “indivíduo clássico”, gerado a partir da observação do indivíduo quântico e através do qual o programa que constrói o *scheduling* é criado.
- *token* de função (TF): identificador único que representa internamente as funções da PGLOGIQ descritas na seção 4.2.
- *token* de terminal (TT): identificador único que representa internamente os terminais da PGLOGIQ descritos na seção 4.2.

4.3.1. Indivíduo Quântico

Como apresentado na seção 3.4, a unidade básica de informação adotada pela PGLIQ é o *qudit*. Esta informação pode ser descrita por um vetor de estado em um sistema mecânico quântico de d níveis, o qual equivale a um espaço vetorial d -dimensional, onde d é o número de estados nos quais o *qudit* pode ser medido. Isto é, d representa a cardinalidade do *token* que terá seu valor determinado pela observação do seu respectivo *qudit*.

O estado de um *qudit* (ψ) é uma superposição linear dos d estados e pode ser representado pelo somatório das probabilidades de que o *qudit* seja encontrado em cada estado quando observado (eq 4.2). Um *qudit* representa todas as probabilidades até o momento em que ele é observado, indicando o valor do *token* associado (função ou terminal) a essa observação.

$$|\psi\rangle = \sum_{i=0}^{d-1} p_{(i)} \quad (4.2)$$

O estado de um *qudit* é implementado como uma estrutura de dados similar a uma roleta. A Figura 11 ilustra a implementação de um *qudit* com distribuição não uniforme das probabilidades de cada estado para as principais funções implementadas na PGLOGIQ.

Figura 11 – Representação de um *qudit* com probabilidades não uniformemente distribuídas.

A implementação do processo de observação do *qudit* pode ser definida pela função apresentada na Figura 12. Nesta figura, r é um valor real entre 0 e 1 gerado aleatoriamente como o primeiro passo da observação, e $T(r)$ retorna o valor observado para o *token*, indicando qual a função ou terminal observado.

$$T(r) = \begin{cases} 0, & \text{se } 0 \leq r < p'_0 \\ 1, & \text{se } p'_0 \leq r < p'_1 \\ \vdots & \\ N, & \text{se } p'_{N-1} \leq r < p'_N \end{cases}$$

Figura 12 – Função que define o valor do *token* associado à observação do *qudit*

A interpretação da Figura 13 diz que, para a função principal (símbolo <func>) do gene em questão, há probabilidade 0,336 de ser observada a instrução “NOP”, 0,011 de ser observada a instrução “RcvItemMax”, 0,590 de ser observada a instrução “RcvItemProp” (vide *RcvItPr* na Figura 13) e assim sucessivamente. A mesma estrutura é utilizada para a representação dos terminais associados aos argumentos de cada função principal.

0	0,336	0,347	0,937	0,948	0,958	0,969	0,979	0,990
0	1	2	3	4	5	6	7	8
p_{NOP}	$p_{RcvItMax}$	$p_{RcvItPr}$	$p_{FdCduMax}$	$p_{FdCduMaxIj}$	$p_{FdCduMaxIj}$	$p_{FdCduMaxIj}$	$p_{FdCduMaxIj}$	$p_{PipeTrProp}$

Figura 13 – Exemplo de distribuição de probabilidades do símbolo <func> em um gene quântico

O cromossomo do indivíduo quântico, ou “cromossomo quântico”, é representado por uma lista de estruturas denominadas “genes quânticos”. Um gene quântico é composto por um *qudit* de função (QF), que representa a superposição de todas as funções predefinidas pelo conjunto de funções e cada

uma dessas funções referencia *qudits* de terminal (QT), num processo contínuo até que todas as funções e terminais definidos pela gramática sejam representados. A estrutura desenvolvida para o gene quântico se assemelha à representação em árvore da programação genética tradicional. No entanto, neste trabalho, ela é aplicada ao gene quântico e não ao programa gerado. A Figura 14 apresenta a estrutura em árvore que representa o gene quântico implementado neste modelo e a Figura 15 explicita as possibilidades de cada estado.

Figura 14 – Estrutura do gene quântico

<func>																	
TF	Prob	Nome	TT	TnqT	Prob	Conteúdo	navio1	TT	Prob	Nome	navio2	TT	Prob	Conteúdo	TT	Prob	Conteúdo
0	p ₀ '	Nop															
1	p ₁ '	RcvItemMax		TnqT			shipAndTank										
			0	p ₀ '	TqT01		navio1				navio2						
			1	p ₁ '	TqT02		0	p ₀ '	tanq1		0	p ₀ '	tanq1				
			2	p ₂ '	TqT03		1	p ₁ '	tanq2								
2	p ₂ '	RcvItemProp	3	p ₃ '	TqT04												
				TnqT			shipAndTank								itemProp		
			0	p ₀ '	TqT01		navio1				navio2				0	p ₀ '	0.1
			1	p ₁ '	TqT02		0	p ₀ '	tanq1		0	p ₀ '	tanq1		1	p ₁ '	0.2
3	p ₃ '	FeedCDUMax	2	p ₂ '	TqT03		1	p ₁ '	tanq2						2	p ₂ '	0.3
			3	p ₃ '	TqT04										3	p ₃ '	0.4
				TnqR			TT	Prob	Conteúdo						4	p ₄ '	0.5
			0	p ₀ '	TqR01		0	p ₀ '	UDA-01						5	p ₅ '	0.6
4	p ₄ '	FeedCDUProp	1	p ₁ '	TqR02		1	p ₁ '	UDA-02						6	p ₆ '	0.7
			2	p ₂ '	TqR03										7	p ₇ '	0.8
			3	p ₃ '	TqR04										8	p ₈ '	0.9
				TnqR			TT	Prob	Conteúdo								
5	p ₅ '	FeedCDUMaxInj	0	p ₀ '	TqR01		0	p ₀ '	UDA-01								
			1	p ₁ '	TqR02		1	p ₁ '	UDA-02								
			2	p ₂ '	TqR03												
			3	p ₃ '	TqR04												
6	p ₆ '	FeedCDUPropInj		TnqR			TT	Prob	Conteúdo								
			0	p ₀ '	TqR01		0	p ₀ '	UDA-01								
			1	p ₁ '	TqR02		1	p ₁ '	UDA-02								
			2	p ₂ '	TqR03												
7	p ₇ '	PipeTransfMax	3	p ₃ '	TqR04												
				TnqR			TT	Prob	Conteúdo								
			0	p ₀ '	TqR01		0	p ₀ '	TqR01								
			1	p ₁ '	TqR02		1	p ₁ '	TqR02								
8	p ₈ '	PipeTransfProp	2	p ₂ '	TqR03		2	p ₂ '	TqR03								
			3	p ₃ '	TqR04		3	p ₃ '	TqR04								
				TnqT			TT	Prob	Conteúdo								
			0	p ₀ '	TqT01		0	p ₀ '	TqR01								
9	p ₉ '	PipeTransfProp	1	p ₁ '	TqT02		1	p ₁ '	TqR02								
			2	p ₂ '	TqT03		2	p ₂ '	TqR03								
			3	p ₃ '	TqT04		3	p ₃ '	TqR04								
				TnqR			TT	Prob	Conteúdo								
10	p ₁₀ '	PipeTransfProp	0	p ₀ '	TqR01		0	p ₀ '	TqR01								
			1	p ₁ '	TqR02		1	p ₁ '	TqR02								
			2	p ₂ '	TqR03		2	p ₂ '	TqR03								
			3	p ₃ '	TqR04		3	p ₃ '	TqR04								

Figura 15 – Todas as possibilidades de estados do *qudit* de um gene quântico

4.3.2. Indivíduo Clássico

Um gene clássico é o resultado de todas as observações de um gene quântico necessárias para definir a função principal e seus argumentos, o que equivale a dizer que cada gene clássico representa uma instrução completa para o programa. O cromossomo de um indivíduo clássico é composto pela

disposição linear de todos os genes clássicos. A execução do programa gerado pelo indivíduo clássico é feita da primeira para a última linha, como é característico da programação genética linear, onde cada uma dessas linhas é um gene clássico. Depois de submetido aos testes de restrições, o indivíduo clássico representa o *scheduling* do cenário.

Como cada gene quântico é observado para gerar um gene do indivíduo clássico (ou seja, uma instrução completa), estes indivíduos necessariamente possuem o mesmo comprimento. No entanto, apesar do cromossomo do indivíduo clássico possuir comprimento fixo, o programa efetivo que ele representa possui comprimento variável devido à existência da instrução “NOP” no conjunto de funções principais. Isto acontece porque essa instrução representa uma “não operação”, o que faz com que o processo de geração do programa ignore os genes nos quais o NOP seja a instrução equivalente ao *token* observado, ou seja, valor do *token* de função igual a 0.

Para melhor compreensão de como acontece a observação do gene quântico e a consequente definição do valor do *token* associado, será utilizado o exemplo apresentado na Figura 12, que representa o estado de um indivíduo quântico após algumas iterações evolutivas. Considere-se que para a função principal foi gerado aleatoriamente o valor 0,796 e, portanto, pela aplicação da equação representada na Figura 11, o *token* de função observado valerá 2, representando a função *RcvItemProp*. Esta função demanda os argumentos *navioeTnq*, *tnqT* e *itemProp*. Respeitando a ordem de chegada dos navios, o primeiro a ser descarregado é o navio1 e, pelas Figuras 9 e 14, sabe-se que este navio possui dois *tanquesCN* que devem ser descarregados. Pela observação deste gene, é gerado aleatoriamente o valor 0,33 e, portanto, pela aplicação da equação representada na Figura 11, o *token* de terminal observado valerá 0, representando o primeiro *tanqueCN*. Processo equivalente é feito para os argumentos *tnqT* e *itemProp*, para os quais é suposto que foram gerados aleatoriamente os valores 0,645 e 0,6, respectivamente. Pela aplicação da equação da Figura 11, o *token* de terminal observado para o tanque do terminal valerá 3 e para a proporção do item valerá 1, correspondendo aos elementos terminais: *TqT04* e *itemProp* 0,4. A finalização do processo de observação deste gene quântico implica na geração do gene clássico que é dado pela instrução *RcvItemProp*(navio1-tanqueCN1, TqT04, 0,4), ou seja, descarregue do navio 1-tanqueCN 1 para o *TqT04* o volume equivalente a 40% do máximo possível para esta atividade.

4.3.3. Avaliação

Como apresentado na seção 3.1, a avaliação de um indivíduo em PG contempla a execução completa do programa que ele representa. Na modelagem desenvolvida, para atingir essa execução, o indivíduo clássico precisa ter cada uma de suas instruções avaliada iterativamente.

Nem todas as instruções geradas pela observação do indivíduo quântico e que compõem o indivíduo clássico podem ser executadas. Além da instrução “NOP” discutida na seção 4.3.2, outras instruções reais de programação podem não ser executadas por infringirem restrições da modelagem. Por exemplo, um gene clássico pode conter a instrução `FeedCDUMax(UDA-01, TqT03)` e ao tentar executá-la, o simulador de planta identifica que ela viola uma restrição de tempo mínimo de atividade, pois o `TqT03` não tem volume suficiente para se manter na carga de UDA-01 por, pelo menos, 24 horas.

No sistema proposto, o indivíduo clássico é percorrido linearmente do primeiro até o último gene e cada uma das suas instruções é submetida às diferentes premissas e restrições da modelagem implementada. As atividades executadas são transferidas para uma estrutura que representa o *schedule* efetivamente. A execução linear é feita iterativamente, ou seja, após a tentativa de execução da última instrução, o indivíduo é novamente percorrido por completo. É interrompido este processo cíclico quando uma varredura completa do indivíduo não acarreta na execução de nenhuma instrução. Esta etapa da avaliação pode ser entendida como a construção do *schedule* e este objeto é, então, submetido à função de avaliação.

Devido à dinâmica da atividade e à grande quantidade de variáveis que precisariam ser avaliadas, na programação de produção não é tarefa fácil trabalhar com diretrizes que converjam para um único objetivo, fazendo com que referências operacionais sejam utilizadas para orientar o que pode ser classificado como uma boa solução.

Neste trabalho a evolução é orientada para o atendimento dos objetivos priorizados segundo descrição a seguir:

1. Atendimento à operação de carga da UDA: as unidades de destilação devem operar continuamente durante todo o cenário. Soluções que minimizam o tempo total de parada das UDAs são classificadas como melhores. Para o cálculo deste tempo é considerada cada uma das UDAs, por exemplo, se uma UDA parar

por 10 horas e outra parar por 15 horas, o valor deste objetivo será 25 horas.

2. Descarregamento de navio: o navio deve ser completamente descarregado dentro do período especificado. Primeiramente é avaliado se o navio foi completamente descarregado. Não havendo volume residual, é avaliado se houve atraso para completar essa atividade. Este é o segundo objetivo mais importante a ser atendido e mede o desvio em relação ao atendimento perfeito, cujo valor é zero.
3. Tempo de operação de duto: é uma medida indireta para aumentar a movimentação entre o terminal e a refinaria, facilitando o atendimento de carga da UDA e a manutenção de um nível de estoque seguro na refinaria. Sempre que o duto fica parado é facilitado o aumento da formação de interface entre as parcelas. A interface é a mistura entre os óleos de diferentes parcelas que ganha características intermediárias entre essas, mas que são desconhecidas porque não se conhece o quanto dos volumes originais se envolveu na mistura. Por essa e outras razões operacionais, soluções que minimizem o tempo que o duto fica parado são melhores.
4. Trocas de tanque: cada operação de troca de tanque provoca um regime transiente no sistema, além do custo (não quantificado neste trabalho) associado à operação. Se não houver benefício na execução de uma troca de tanque (por exemplo, atender restrições), a evolução deve optar por minimizá-la.
5. Tanque de injeção: o uso de tanque de injeção na carga de unidade implica na alocação de dois recursos durante o período da atividade, diminuindo a disponibilidade de tanques na refinaria, o que pode dificultar a programação. Como último critério de desempate entre dois indivíduos igualmente avaliados até o momento, será considerado melhor o que minimizar o uso deste recurso.

Esta priorização dos objetivos foi definida para o estudo de caso. No entanto, é permitido configurar o modelo de maneira a reordená-las de acordo com os critérios de cada refinaria.

4.3.4. Evolução

A evolução dos indivíduos é baseada na atuação do operador quântico (OPQ) sobre as probabilidades de observação do *qudit*. É orientada pela aptidão destes indivíduos através da medida da função de avaliação.

Para os melhores indivíduos clássicos avaliados, o operador quântico incrementa, no indivíduo quântico correspondente, o valor de um passo (parâmetro do modelo) para a probabilidade de observação das funções e terminais desses indivíduos em cada gene quântico. Para garantir a consistência do somatório de probabilidades, é feita a normalização dos valores após a aplicação do operador. A atuação do OPQ é apresentada em (eq. 4.3).

$$p_{(i,g)} = p_{(i,g-1)} + OPQ \quad (4.3),$$

onde $p_{(i,g-1)}$ representa a probabilidade de seleção da função ou terminal i na geração g , que represente a função ou terminal observado no melhor indivíduo na geração $g-1$ e OPQ representa o incremento na probabilidade de observação da função ou terminal.

Independentemente das instruções que não foram executadas, como discutido na seção 4.3.3, todos os genes clássicos dos melhores indivíduos são usados para orientar a evolução dos seus respectivos genes quânticos.

4.3.5. Funcionamento do Modelo

A PGLOGIQ manipula duas populações, uma quântica e outra clássica, ambas possuindo o mesmo número de indivíduos que, por sua vez, possuem o mesmo tamanho. Também manipula uma população clássica auxiliar, de mesmas dimensões das anteriores, que resulta das observações dos indivíduos quânticos da geração atual e é usada para definição dos melhores indivíduos clássicos obtidos pela ordenação conjunta desta geração com os oriundos da geração anterior. A Figura 16 e as etapas descritas a seguir caracterizam o ciclo de uma geração da PGLOGIQ.

1. A população quântica é observada, gerando os indivíduos clássicos auxiliares.

2. Os indivíduos clássicos auxiliares e os clássicos oriundos da geração anterior são ordenados segundo os critérios da função de avaliação multiobjetivo. A população clássica agora é composta pelos 6 melhores indivíduos.
3. Aplicação do operador quântico, ou seja, o incremento de um passo na probabilidade de observação de cada função e terminal que representa o indivíduo clássico, ao indivíduo quântico correspondente. Neste trabalho, significa dizer que os 6 melhores indivíduos clássicos, ordenadamente, orientam o incrementos das probabilidades dos 6 indivíduos quânticos. É nesta etapa que efetivamente ocorre a evolução, pois é quando o indivíduo quântico é atualizado de maneira a aumentar a probabilidade de que a evolução gere indivíduos clássicos mais parecidos com os melhores gerados até então.
4. Avaliação e comparação do melhor indivíduo clássico obtido nesta geração com o melhor indivíduo clássico obtido na evolução até o presente momento, representado na Figura 16 como C_M . Se um novo indivíduo for melhor que C_M , este indivíduo será copiado, substituindo o C_M atual.

O desenvolvimento do modelo começa com a criação da população quântica e a distribuição de probabilidades das funções e terminais. Neste trabalho, foi adotado que, para o conjunto de funções da PGLOGIQ, o valor da probabilidade da instrução NOp seria um parâmetro do modelo e as demais instruções teriam sua probabilidade distribuídas igualmente, ou seja, se o número total de funções for nove e o valor inicial da probabilidade de observação da função NOp for 0,8, as demais funções principais terão valor de probabilidade inicial dado por $0,2/8$, igual a 0,025, onde 0,2 é a diferença entre o total (1,0) e a probabilidade de NOp (0,8) e 8 é o número de instruções, excluindo NOp. No caso dos argumentos dessas funções, todos receberiam inicialmente valor igualmente distribuído de suas probabilidades, ou seja, se a instrução FeedCDUMax possui para o argumento "tnqR" os terminais "TqR01", "TqR02", "TqR03" e "TqR04", cada um destes terminais terá sua probabilidade de observação iniciada em 0,25.

Figura 16 – Ciclo de uma geração da PGLOGIQ

A evolução ocorre através da observação dos indivíduos quânticos, avaliação e comparação dos indivíduos clássicos e aplicação do operador quântico.

Para aumentar a capacidade do modelo de busca no espaço de solução e evitar uma convergência prematura, foi definido que sempre que a população clássica convergisse de forma que a avaliação de todos os seus indivíduos clássicos fosse igual, a população quântica e a população clássica seriam iniciadas novamente persistindo somente o melhor indivíduo clássico na população. Esse mecanismo de reiniciar a população é chamado de *reset* neste trabalho. O critério de parada adotado foi o atingimento do número de gerações definidos para a execução. As Figuras 17 e 18 apresentam, respectivamente, o ciclo simplificado e detalhado de funcionamento do modelo.

É muito importante compreender que o valor de probabilidade de observação é um atributo associado ao gene quântico de cada indivíduo independentemente de a lista de argumentos compartilhar os mesmos elementos terminais. Para melhor entendimento, considere-se um exemplo em que, após algumas gerações, os genes quânticos 27 e 102 possuam a distribuição de probabilidades segundo a tabela 6.

Figura 17 – Ciclo simplificado do modelo proposto

Instrução	Distrib. de Probab Gene 27	Distrib. de Probab Gene 102
NOp	0,400	0,360
RcvItemMax	0,100	0,030
RcvItemProp	0,080	0,160
FeedCDUMax	0,286	0,310
FeedCDUProp	0,033	0,039
FeedCDUMaxInj	0,028	0,028
FeedCDUPropInj	0,017	0,015
PipeTransfMax	0,031	0,023
PipeTransfProp	0,025	0,035

Tabela 6 – Exemplo de distribuição de probabilidades para dois genes quânticos

Suponha-se que, ao observar cada um destes genes quânticos, tenha sido apontada a mesma instrução FeedCDUMax. Esta função possui os argumentos “tnqR” e “uda”, representados pelos terminais apresentados nas seções 4.2. e 4.3. Entretanto, a distribuição de probabilidades destes terminais poderá ser diferente entre os genes quânticos 27 e 102, como apresentado na tabela 7.

Argumento	Terminais	Probabilidades Gene 27	Probabilidades Gene 102
tnqR	TqR01	0,439	0,088
	TqR02	0,018	0,135
	TqR03	0,291	0,103
	TqR04	0,252	0,674
uda	UDA-01	0,679	0,771
	UDA-02	0,321	0,229

Tabela 7 – Exemplo de distribuição de probabilidades para os mesmos elementos terminais em genes quânticos diferentes

4.3.6. Simulador de Planta

O simulador de planta é o responsável por garantir a integridade de todo indivíduo gerado, seja ele uma boa solução ou não. Dessa forma, para cada gene clássico, ou seja, para cada atividade que se pretende executar, é preciso que o simulador de planta:

- Avalie o atendimento a todas as restrições pertinentes;
- Estabeleça, com base no volume da atividade e nas informações do próprio simulador sobre os equipamentos de origem e destino envolvidos, o instante inicial, a duração e, consequentemente, o tempo final de cada atividade;
- Calcule o inventário dos equipamentos da planta após a execução da atividade.

No modelo proposto, as atividades são executadas ordenadamente a partir da leitura linear dos genes de um indivíduo clássico. Entretanto, a definição do instante de tempo em que a atividade ocorre é conhecida no simulador, que possui a informação sobre a disponibilidade de todos os equipamentos. A atividade em questão será executada no primeiro instante de tempo em que os equipamentos estiverem disponíveis. Por exemplo, um gene clássico indica que o TqR01 deve alimentar UDA-01 com todo seu volume operacional. O simulador de planta alocará esta atividade no primeiro horário em que TqR01 e UDA-01 estejam disponíveis. Ou seja, após executar atividades representadas em genes anteriores TqR01 está disponível na hora 23 e UDA-01 na hora 33, esta atividade será alocada na hora 33. Após considerar volume útil do tanque e

vazão da unidade, os equipamentos estarão disponíveis a partir da hora 67 para outras atividades representadas pelos próximos genes clássicos.

Figura 18 – Fluxograma de funcionamento do modelo proposto

5 Estudos de Caso

5.1. Descrição

Este trabalho se propôs a desenvolver um modelo capaz de executar o *scheduling* de petróleo para um horizonte de tempo variável, contemplando todas as atividades associadas a ele e que estivessem dentro do escopo de decisão da refinaria, ainda que não restritas a seus limites físicos.

Uma vez definido o algoritmo e toda a estrutura de modelagem proposta, foi realizada a avaliação de alguns parâmetros e definição da melhor configuração dentre as opções testadas. Para tal, foi utilizado como caso base o cenário chamado CEN 05, detalhado no item 5.2 a seguir.

Após a análise que definiu os parâmetros de configuração, e sob essas condições, foram executados todos os cenários de programação para avaliação dos resultados do modelo. Os 5 cenários usados neste estudo são baseados em casos reais de uma refinaria e, a título de avaliação do desempenho do modelo proposto, os resultados otimizados são comparados com os cenários reais do ponto de vista do valor de cada um dos objetivos da função de avaliação.

A implementação do modelo foi totalmente construída em ambiente Visual Studio.net, linguagem C#. Os arquivos de dados são do tipo XML (*eXtensible Markup Language*) e manipulados via interface gráfica desenvolvida usando a mesma linguagem.

Os cenários estudados também foram executados em um sistema baseado em AG para comparação de desempenho com a modelagem usando PGLOGIQ. Por se tratar de uma evolução não determinística, para cada avaliação foram realizadas diversas rodadas do modelo.

Todos os experimentos foram executados em um computador baseado em um processador Intel i7-3960X 3,30GHz e com 16.0 GB de memória RAM. O tempo médio de execução de cada rodada do modelo variou entre 15 e 20 minutos.

5.2. Cenários

Todos os cenários avaliados possuem um conjunto básico de informações em comum, sendo eles:

- Lastro e capacidade máxima de cada tanque, cujos valores são encontrados nas tabelas 2 e 3.
- Base de dados de petróleo: qualidades, cujos valores são encontrados na tabela 8.
- Base de dados de cortes de destilação: perfil de rendimento e qualidade dos cortes para cada petróleo. A Figura 19 tem a representação de um subconjunto desses valores para compreensão da estrutura das informações.
- Restrições de caráter operacional ou de qualidades, descritas na seção 5.2.1.
- Pre-Def: podem ser compreendidas como atividades que já estavam acontecendo no início do cenário e, portanto serão executadas até a sua conclusão. O modelo não pode alterá-las, seja em seu volume, duração ou equipamentos envolvidos. Essas atividades representam movimentações para carga das unidades de destilação ou de transferência entre o terminal e a refinaria. Todos os cenários possuem pelo menos uma Pre-Def, característica de cada cenário.

Em todos os cenários a vazão de movimentação do DUTO1 é constante para todo o horizonte com valor igual a 1.250 m³/h.

Como já mencionado, todos os cenários apresentados neste trabalho possuem correlação com cenários reais de uma refinaria e, em razão da demanda corrente neste período, as unidades de destilação estão trabalhando somente em um modo de operação, aqui chamado CAMPANHA01. Dessa forma, em todos os cenários as unidades UDA-01 e UDA-02 estão configuradas para esta campanha.

Todas as informações referentes aos dados iniciais de inventário para cada cenário são apresentadas no Apêndice A.

Óleo	API	Densidade (g/cm ³)	IAT (mgKOH/g óleo)	Enxofre (%m)
PT01	47,6	0,7901	0,052	0,05
PT02	45,2	0,8008	0,050	0,06
PT03	28,3	0,8855	0,150	0,44
PT04	19,9	0,9346	1,190	0,74
PT05	22,8	0,9170	0,740	0,72
PT06	23,1	0,9153	0,590	0,67
PT07	20,6	0,9303	1,240	0,64
PT08	20,6	0,9303	1,240	0,64
PT09	27,0	0,8927	0,410	0,59
PT10	28,3	0,8855	0,090	0,58
PT11	28,3	0,8855	0,090	0,58
PT12	22,1	0,9212	1,820	0,42

Tabela 8 – Propriedades dos petróleos utilizados

Corte	PT01	PT02	...	PT11	PT12
Rendimento (%)					
GLP	3,3	2,4		2,0	0,7
NL	22,8	26,5		9,4	3,6
DP	5,8	5,1		6,2	7,5
RAT	11,8	13,5		43,4	58,1
Densidade (g/cm³)					
GLP	0,5613	0,5618		0,5451	0,5487
RAT	0,9351	0,9343		0,9805	0,9703
Enxofre (%m)					
NL	0,003	0,006		0,001	0,002
RAT	0,204	0,242		0,762	0,565
Acidez (mgKOG/g óleo)					
DL	0,10	0,05		0,12	2,15
RAT	0,32	0,09		0,08	0,74

Figura 19 – Subconjunto dos rendimentos volumétricos e propriedades dos petróleos para CAMPANHA01 da UDA-01

5.2.1. Restrições

Em todos os cenários foi adotado o mesmo conjunto de restrições, descritos a seguir:

- **Lastro:** durante todo o horizonte do cenário, não foi permitido que o modelo fizesse uma movimentação que implicasse em que o volume total dentro do tanque fosse menor que o volume do lastro. Essa regra foi aplicada para todos os tanques.
- **Capacidade Máxima:** durante todo o horizonte do cenário, não foi permitido que o modelo fizesse uma movimentação que implicasse em que o volume total dentro do tanque fosse maior que a capacidade volumétrica máxima do tanque. Essa regra foi aplicada para todos os tanques.
- **Duração mínima de carga da UDA:** um tanque de petróleo, quando alocado para fornecer carga a uma UDA, deveria permanecer nesta atividade por pelo menos 24 horas.
- **Duração mínima da transferência entre terminal e refinaria:** uma movimentação que transferisse óleo do terminal para a refinaria somente poderia ser realizada se tivesse pelo menos 3 horas de duração utilizando os mesmos recursos alocados.
- **Restrição de qualidade de carga e qualidade dos cortes intermediários das unidades de destilação atmosférica:** independentemente da configuração de tanques alocados para compor a carga de uma unidade de destilação, os limites apresentados na tabela 9 deveriam ser respeitados em todo o horizonte de programação.

UDA	Corrente	Qualidade	Limite
UDA-01 e UDA-02	Carga	Acidez	Máximo 1,3
UDA-01	DL e DP	Acidez	Máximo 1,9
UDA-02	DL e DP	Acidez	Máximo 1,4
UDA-01	RAT	Acidez	Máximo 1,7
UDA-02	RAT	Acidez	Máximo 1,5

Tabela 9 – Restrições de qualidade aplicadas aos cenários

O óleo PT12 possui valor de acidez cerca de 40% acima do limite permitido, demandando atenção em relação à sua composição na mistura que alimenta as unidades. Os petróleos PT04, PT07, PT08 possuem valor próximo ao limite de especificação para esta mesma restrição, podendo levar o programador de produção a ter uma postura conservadora sobre a utilização destes óleos. Postura semelhante pode acontecer sobre o corte de RAT do PT04, também um valor limite.

Os petróleos PT08 e PT12 possuem alta acidez nos cortes DL e DP e, portanto, demandam muito cuidado em relação à sua composição na mistura usada nas unidades de destilação. O petróleo PT07 possui valor limite para os cortes DL e DP e, assim como acontece com a preocupação do óleo, também pode levar o programador a ter uma postura conservadora na sua utilização.

5.2.2. Seleção dos Cenários

Os cenários foram escolhidos de maneira que o conjunto tivesse maior diversidade de configurações segundo os critérios: duração do horizonte de programação, programação de navios a ser descarregado, volume total de óleo no início do cenário, configuração do duto que conecta terminal e refinaria e configuração das Pre-Def. A tabela 10 apresenta o valor de alguns desses critérios para os cinco cenários. Os valores desses termos em cada cenário serão discutidos nas seções correspondentes.

Cenário	Duração (h)	Navios (tanques/navio)	Volume a ser descarregado (m ³)	Volume útil terminal + refinaria (m ³)
CEN01	192	navio 1(1 tanque)	66.666,6	202.221,6
CEN02	192	navio 1(1 tanque) navio 2(1 tanque)	112.500,0	184.966,8
CEN03	216	navio 1(2 tanques)	133.333,3	174.956,3
CEN04	240	navio 1(1 tanque)	129.166,7	275.486,1
CEN05	240	navio 1(2 tanques) navio 2(1 tanque)	262.500,0	102.160,5

Tabela 10 – Duração, inventário e configuração de navios dos cenários estudados.

5.2.2.1. CEN01

Este cenário, comparativamente dentre os selecionados, pode ser considerado de mais baixa complexidade por possuir menor horizonte de programação, alto volume de inventário inicial em tanque, o que garante a carga das unidades até a hora 60 quando ocorre recebimento de óleo, configuração de um navio com apenas um tanque a ser descarregado e baixo volume de descarga. O navio não descarrega óleo ácido, mas ele é recebido no terminal pelo deslocamento da linha LS1. No início do cenário este tipo de óleo (PT12) está concentrado no TqT02 e TqT04, formando aproximadamente 30.000 m³. O DUTO1 está completamente cheio da mesma mistura, ou seja, existe apenas uma parcela dentro do duto. No que diz respeito às Pre-Def, possui atividade de carga para as unidades de destilação e movimentação entre terminal e refinaria, conforme apresentado na tabela 11. A tabela 12 apresenta os valores de vazão das unidades de processo e descarga do navio.

Tipo de Atividade	Equipto Origem	Equipto Destino	Hora Final
Carga de Unidade	TqR04	UDA-01	12
Carga de Unidade	TqR03	UDA-02	12
Transferência Duto	TqT01	TqR01	4

Tabela 11 – Pre-Def de CEN01

Equipamento	Parâmetro	Valor
Navio1 (1 tanqueCN)	Vazão de descarga	3.750 m ³ /h
	Hora chegada/saída	60/80
UDA-01	Vazão	347,2 m ³ /h
UDA-02	Vazão	677,0 m ³ /h

Tabela 12 – Vazões de navio e UDAs de CEN01

5.2.2.2. CEN02

Em relação ao cenário CEN01, CEN02 traz como complicador a programação dos navios e a composição da mistura de petróleos. No mesmo horizonte de programação, são dois navios a serem descarregados totalizando

um volume quase 70% maior. O inventário inicial do terminal está baixo (cerca de 50.000 m³), sinalizando que as atividades do duto provavelmente ocorrerão somente após o início da descarga do navio. O segundo navio transporta um óleo de alta acidez, o que deve dificultar a programação para atender a esta restrição na carga das unidades de destilação. Inicialmente, o DUTO1 está cheio com a mesma mistura, ou seja, existe apenas uma parcela no seu inventário. No que diz respeito às Pré-Def, possui somente atividade de carga para as unidades de destilação, conforme apresentado na tabela 13, o que vai de acordo com a sinalização de que as movimentações de duto devem estar concentradas mais a frente do horizonte. A tabela 14 apresenta os valores de vazão das unidades de processo e descarga do navio.

Tipo de Atividade	Equipto Origem	Equipto Destino	Hora Final
Carga de Unidade	TqR04	UDA-01	24,0
Carga de Unidade	TqR03	UDA-02	29,5

Tabela 13 - Pre-Def de CEN02

Equipamento	Parâmetro	Valor
Navio1 (1 tanqueCN)	Vazão de descarga	3.750 m ³ /h
	Hora chegada/saída	12/32
Navio2 (1 tanqueCN)	Vazão de descarga	3.750 m ³ /h
	Hora chegada/saída	58/78
UDA-01	Vazão	347,2 m ³ /h
UDA-02	Vazão	677,0 m ³ /h

Tabela 14 - Vazões de navio e UDAs de CEN02

5.2.2.3. CEN03

Em relação aos casos apresentados anteriormente, o CEN03 traz um maior horizonte de programação e diferente configuração de navios. O volume total a ser descarregado é 18% maior que em CEN02 e ainda existe o agravante de que todo este volume está em apenas um navio, ou seja, todo este volume deve ser alocado em menor intervalo de tempo. De onde se espera que haja boa

movimentação do duto para que haja espaço nos tanques do terminal para receber este descarregamento. Possui significativa quantidade de óleo ácido em 3 tanques da refinaria. Inicialmente, o DUTO1 está completamente cheio da mesma mistura, ou seja, existe apenas uma parcela dentro do duto. No que diz respeito às Pré-Def, possui atividade de carga para as unidades de destilação e movimentação entre terminal e refinaria, conforme apresentado na tabela 15. A tabela 16 apresenta os valores de vazão, unidades de processo e descarga do navio.

Tipo de Atividade	Equipto Origem	Equipto Destino	Hora Final
Carga de Unidade	TqR03	UDA-01	24
Carga de Unidade	TqR04	UDA-02	24
Transferência Duto	TqT03	TqR01	4

Tabela 15 – Pre-Def de CEN03

Equipamento	Parâmetro	Valor
Navio1 (2 tanquesCN)	Vazão de descarga	3.750 m ³ /h
	Hora chegada/saída	58/98
UDA-01	Vazão	347,2 m ³ /h
UDA-02	Vazão	659,7 m ³ /h

Tabela 16 - Vazões de navio e UDAs de CEN03

5.2.2.4. CEN04

Este cenário possui o maior horizonte de programação e grande volume de estoque total no início do horizonte, com o terminal próximo de sua capacidade máxima (cerca de 210.000 m³ em inventário para 240.000 m³ máximo). Dessa forma, é esperada uma boa movimentação de duto no início do cenário para que haja espaço para o descarregamento do navio.

O DUTO1 possui duas misturas, ou seja, existem duas parcelas no seu inventário. No que diz respeito às Pré-Def, possui atividade de carga para as unidades de destilação e movimentação entre terminal e refinaria, conforme

apresentado na tabela 17. A tabela 18 apresenta os valores de vazão, unidades de processo e descarga do navio.

Tipo de Atividade	Equipto Origem	Equipto Destino	Hora Final
Carga de Unidade (tanque base)	TqR02 (70%)	UDA-01	4
Carga de Unidade (tanque injeção)	TqR01 (30%)		
Carga de Unidade	TqR03	UDA-02	4
Transferência Duto	TqT02	TqR04	3,7

Tabela 17 - Pre-Def de CEN04

Equipamento	Parâmetro	Valor
Navio1 (1 tanqueCN)	Vazão de descarga	5.000 m ³ /h
	Hora chegada/saída	82/114
UDA-01	Vazão	347,2 m ³ /h
UDA-02	Vazão	659,7 m ³ /h

Tabela 18 - Vazões de navio e UDAs de CEN04

5.2.2.5. CEN05

Este cenário pode ser considerado de alta complexidade, pois possui o maior horizonte de programação, 2 navios de grande capacidade, sendo que um destes possui dois tanquesCN e o modelo deve decidir como proceder o descarregamento, além da correta alocação de recursos para recebimento de todo o volume. Baixo inventário inicial de petróleo em tanque, sendo que o terminal está com volume inicial útil em cerca de 11.000 m³ e a refinaria com aproximadamente 3 dias de óleo em estoque, o que é considerado um valor limite para garantir a manutenção da vazão de carga das unidades.

Inicialmente, o DUTO1 está completamente cheio da mesma mistura, ou seja, existe apenas uma parcela dentro do duto. No que diz respeito às Pre-Def, possui atividade de carga para as unidades de destilação e movimentação entre

terminal e refinaria, conforme apresentado na tabela 19. A tabela 20 apresenta os valores de vazão, unidades de processo e descarga do navio.

Tipo de Atividade	Equipto Origem	Equipto Destino	Hora Final
Carga de Unidade	TqR02	UDA-01	63,5
Carga de Unidade (tanque base)	TqR01 (88,3%)	UDA-02	63,5
Carga de Unidade (tanque injeção)	TqR02 (11,7%)		
Transferência Duto	TqT03	TqR03	8,0

Tabela 19 - Pre-Def de CEN05

Equipamento	Parâmetro	Valor
Navio1 (2 tanquesCN)	Vazão de descarga	3.750 m ³ /h
	Hora chegada/saída	11/48
Navio2 (1 tanqueCN)	Vazão de descarga	4.166,7 m ³ /h
	Hora chegada/saída	89/121
UDA-01	Vazão	354,1 m ³ /h
UDA-02	Vazão	591,6 m ³ /h

Tabela 20 - Vazões de navio e UDAs de CEN05

5.3. Definição dos parâmetros do modelo

O cenário CEN05 foi utilizado como caso base para a definição dos parâmetros do modelo por apresentar boa diversidade de atividades a serem representadas, considerando duração, características de inventário e configuração de navios, além de Pre-Def de duto e UDA, sendo uma operação com tanque de injeção.

A inexistência de trabalhos aplicando programação genética e inspiração quântica para solução de problemas de *scheduling* até o desenvolvimento dos experimentos deste trabalho dificultou a obtenção de valores de referência para os parâmetros em casos similares. Com base no trabalho [11], foram selecionadas e avaliadas as combinações apresentadas na tabela 21.

Foram realizados testes com 240 mil avaliações, valor com o qual já se observava que os objetivos de atendimento à carga da UDA e descarregamento de navio sem penalização já tinham estabilidade. O gráfico de evolução dos demais objetivos em rodadas diferentes apresentou características diferentes: rodadas em que a convergência prematura era alcançada e rodadas nas quais, até o final das 240 mil avaliações, a evolução ainda estava acontecendo. Estes comportamentos podem ser vistos nas Figuras 20 e 21, nas quais estão plotadas a evolução do melhor indivíduo para duas rodadas diferentes. O eixo das abcissas é dado pelo número de gerações e o eixo das ordenadas é dado somente pelos objetivos com maior variabilidade, cujos respectivos valores estão trabalhados de maneira que tenham a mesma escala e possam compartilhar o mesmo eixo. Os objetivos de carga na UDA e de operação de duto estão representados na forma percentual do tempo total de parada desses equipamentos em relação ao horizonte total do cenário, o objetivo de descarregamento do navio também está representado pelo mesmo critério. No entanto, gerou-se uma descontinuidade para representar as gerações em que o volume total dos navios não havia sido entregue, o eixo das ordenadas para esse ponto assume o valor de 100%. O número de trocas de tanque está representado em valores absolutos para todas as movimentações do cenário.

Figura 20 – Curva evolutiva do melhor indivíduo de uma rodada com convergência prematura

Figura 21 – Curva evolutiva do melhor indivíduo de uma rodada sem convergência prematura

Figura 22 – Curva evolutiva da média da população na rodada sem convergência prematura

Através da Figura 22, que apresenta a média da mesma população cujo melhor indivíduo foi apresentado na Figura 21, pode ser observada a contribuição dos *resets* da população, cujas gerações estão identificadas pelos marcadores em todas as curvas. Em geral, antes dos *resets* a evolução havia

convergiu para um patamar e, ao reiniciar a população, foi possível que os indivíduos buscassem novos espaços de solução e a otimização continuasse.

Foram realizadas algumas rodadas com 480 mil avaliações e, sob o critério de atendimento aos objetivos de carga das UDAs e descarregamento dos navios, não houve aumento significativo do percentual de soluções aceitas (conceito discutido na seção 5.3.1) e o tempo computacional para execução de cada rodada foi, em média, o dobro. Dessa forma, optou-se por manter todo o modelo e, conseqüentemente, todos os testes utilizando 240 mil avaliações.

Parâmetro	Valores
Tamanho dos indivíduos	128 / 256 genes
Tamanho da população	4 / 6 indivíduos
Percentual de instrução NOp na população inicial	80% / 90%
Tamanho do passo na evolução (× 1.000)	4 / 6 / 8 / 10

Tabela 21 – Parâmetros avaliados para o modelo evolutivo

A combinação dos parâmetros apresentados na tabela 21 gera 32 configurações diferentes. Foram executadas 10 rodadas conhecendo-se a semente do gerador de números pseudoaleatórios de cada configuração e selecionadas 5 combinações, sendo as 2 que obtiveram melhor desempenho e outras 3 de classificação intermediária com o intuito de validar a pré-seleção com menos rodadas. Com estes casos foram executadas 40 rodadas com semente conhecida para o gerador de números pseudoaleatórios. A tabela 22 apresenta as 5 configurações que foram pré-selecionadas e submetidas a avaliação com 40 rodadas.

Parâmetro	Config1	Config2	Config3	Config4	Config5
Tamanho dos indivíduos	128	128	256	256	256
Tamanho da população	6	6	4	4	6
Percentual de instrução NOp na população inicial	80	80	80	80	90
Tamanho do passo na evolução	0,008	0,010	0,004	0,006	0,008

Tabela 22 – Configurações pré-selecionadas dos parâmetros

A execução com mais rodadas manteve a ordem de classificação entre as configurações e confirmou que a combinação com melhor desempenho foi a *Config1*.

A partir desta definição, todas as avaliações subsequentes do modelo foram feitas com tamanho de indivíduo (clássico e quântico) igual a 128 genes, tamanho de população (clássica, quântica e auxiliar) igual a 6 indivíduos, probabilidade de observação da instrução NOp na população inicial igual a 80% e tamanho do passo do operador quântico igual a 0,008. O número de gerações foi mantido em 40.000.

Por ser um algoritmo evolutivo baseado no sorteio que caracteriza as observações do indivíduo quântico para geração do indivíduo clássico, pode-se afirmar que, dependendo da semente utilizada pelo gerador de números pseudoaleatórios do computador, a solução final (*schedule*) obtida será diferente. Por isso, para definir a configuração de parâmetros do modelo, utilizou-se como critério que seria a combinação que desse o maior percentual de boas soluções, fazendo-se necessário definir o que é uma boa solução, ou seja, quais serão aceitas.

5.3.1. Critério Usado para Definir as Soluções Aceitas

A melhor solução dentre todas de uma dada combinação de parâmetros é a que minimiza todos os termos da função objetivo. No entanto, uma solução aceita é caracterizada por um *schedule* que atinge principalmente os dois primeiros objetivos: operação contínua das unidades de destilação e descarregamento completo dos navios sem atraso. Sobre este critério foi aplicada uma tolerância na avaliação de maneira a não invalidar soluções que não atingissem a meta de valor zero desses objetivos, mas que tivessem valores não significativos do ponto de vista operacional. Por exemplo, uma solução que não opere qualquer combinação das torres de destilação até 2% do tempo equivalente ao horizonte do cenário (no caso 4,8 horas), não é ideal, no entanto, dadas as variabilidades da atividade no dia a dia, essa não conformidade provavelmente poderia ser absorvida e as unidades não parariam realmente. Da mesma maneira, foi considerado que um atraso de até 3 horas no descarregamento completo do navio significaria que a atividade é concluída dentro do mesmo dia planejado e a diferença provavelmente poderia ser

absorvida sem incorrer em penalizações. A tabela 22 apresenta os critérios para classificação das soluções.

A Figura 23 apresenta a distribuição do percentual de soluções aceitas apresentado nesta seção para a melhor configuração de parâmetros, Config1. A título de ilustração, a Figura 24 apresenta essa distribuição para a segunda melhor configuração de parâmetros, Config2.

Classificação	Critério	Denominação
Aceita	Parada UDAs 0h E Atraso Navio 0h	Ideal
	0h < Parada UDAs <=1% E/OU 0h < Atraso Navio <= 2h	Boa
	1%< Parada UDAs <=2% E/OU 2h < Atraso Navio <= 3h	Limite
Não Aceita	Parada UDAs > 2% E/OU Atraso Navio > 3h	Não Aceita

Tabela 23 – Critério de aceitação de uma solução de *scheduling*

Figura 23 – Distribuição de soluções para o melhor conjunto de parâmetros, Config1

Figura 24 - Distribuição de soluções para o segundo melhor conjunto de parâmetros, Config2

5.4. Avaliação do Conjunto de Terminais

Como demonstrado na seção 4.3.1, o gene quântico é a superposição de todos os estados que podem representar um gene clássico. Para dar maior flexibilidade ao modelo, inicialmente optou-se por disponibilizar várias possibilidades de estado para os valores dos argumentos que representassem proporções da alternativa máxima de uma decisão volumétrica: feedProp, itemProp, injProp, pipeProp. Diferentemente de parâmetros como tamanho da população, tamanho do indivíduo ou tamanho do passo da evolução, que são característicos do modelo PGLOGIQ, feedProp e os demais “Prop” são característicos do problema estudado e interferem diretamente na interpretação da gramática que o modelo usa. Os valores originalmente disponibilizados para cada um destes argumentos são encontrados nas Figuras 10 e 15.

Uma vez estabelecidos os parâmetros do modelo, decidiu-se avaliar o efeito de uma diminuição das opções disponibilizadas para argumentos do problema de maneira a concentrar os elementos terminais em um subconjunto que orientasse a resolução da maior parte dos casos. A proposta foi avaliar cada argumento e disponibilizar somente um conjunto de valores que, baseado em justificativas operacionais, representasse esse subconjunto mais provável. Esta modelagem facilitaria a seleção dos parâmetros mais comuns, mas seu ônus estaria nas soluções que, para chegarem a ser aceitáveis, precisariam de valores terminais que não estivessem nesse conjunto.

O argumento injProp não foi limitado porque o próprio conceito de tanque de injeção já limita sua participação em 50% da composição da carga.

Dessa forma foram avaliados os seguintes subconjuntos dos argumentos:

- itemProp: 0,3|0,4|0,5|0,6|0,7. O valor máximo desta atividade é comumente limitado pela capacidade do tanque de recebimento. A principal motivação para que um item não ocupe completamente

este espaço estaria em limites de composição ou propriedade em tanque. Uma movimentação que bombeia menos de 20% desse volume máximo não se justifica na maioria das situações, pois demanda um conjunto de operações para executá-la em curto espaço de tempo e não causará impactos expressivos de composição ou qualidade que não possam ser compensados pela mistura terminal ou na refinaria. Raciocínio semelhante se aplica para valores acima de 0,7, que deixa menos de 30% do espaço originalmente disponível em tanque para uma próxima mistura.

- feedProp: 0,6|0,7|0,8. O valor máximo desta atividade é essencialmente limitado pelo volume disponível nos tanques que compõem a carga da unidade (somente base ou com injeção). É incomum que uma configuração de tanques que atenda todas as restrições de qualidade de carga seja alocada para essa tarefa e nela permaneça por um período suficiente para consumir menos da metade da capacidade volumétrica do conjunto. Trocas de tanque de carga provocam um período de regime transiente na unidade, o que deve ser evitado. Portanto, se a configuração de tanques atende aos requisitos, é pouco provável que se retire. Uma motivação para a retirada do conjunto antes do seu volume máximo estaria no caso de se desejar aproveitar a composição do(s) tanque(s) para melhor enquadramento de uma nova mistura nele(s) após recebimento de óleo do terminal.
- pipeProp: 0,6|0,7|0,8. O valor máximo desta atividade é limitado pelos tanques de envio e de recebimento e pelo tamanho da parcela dentro do duto. Raciocínio análogo foi aplicado aqui. Uma motivação para que a movimentação não aconteça em seu volume máximo estaria em limites de composição ou propriedade nos tanques. Programar uma atividade de duto sem aproveitamento máximo de sua continuidade e gerando uma parcela menor que a original dentro do duto, se justificaria nestes casos e, sendo assim, valores maiores para este parâmetro permitiriam impactos de composição sem permitir movimentações de duração muito curta.

Para cada uma das 8 configurações geradas, foram executadas várias rodadas com as mesmas sementes controladas usadas no estudo de parâmetros apresentado na seção 5.3.

A comparação de resultados foi feita com a configuração mais flexível apresentada nas Figuras 10 e 15, e permitiu observar que a restrição de oferta de valores para os parâmetros do problema não melhorou o desempenho do modelo, permanecendo a configuração mais flexível com o maior percentual de boas soluções. A diferença de desempenho não foi significativa na maior parte dos casos e, em algumas configurações, o melhor indivíduo gerado teve exatamente os mesmos valores na função objetivo com parâmetros mais restritos ou mais flexíveis.

A análise do melhor indivíduo de algumas das soluções ideais permitiu observar que, em várias rodadas, mesmo quando as opções de valores mais baixos para `pipeProp` e `feedProp` estavam disponíveis, a observação do respectivo *token* de terminal não assumia valores que endereçavam a estas, embasando a argumentação dada nesta seção para a escolha do subconjunto de valores destes parâmetros.

A observação mais importante deste estudo é que a evolução da PGLOGIQ é robusta ao ponto de não ser prejudicada pela existência de mais estados representados no gene quântico, mesmo quando estes não representam os mais frequentes, sabendo valer-se de todas as suas probabilidades na evolução. Desta forma, optou-se por manter os parâmetros `itemProp`, `feedProp` e `pipeProp` podendo assumir mais valores, como apresentado na seções 4.2 e 4.3.

5.5. Resultados dos Cenários

Uma vez definidos os parâmetros do modelo e os argumentos do problema, foi necessário avaliar o desempenho do modelo em condições mais próximas ao que seria seu uso em ambiente operacional. Para tal, foram executadas diversas rodadas de cada um dos cinco cenários inicialmente propostos, mas desta vez, a semente do gerador de números pseudoaleatórios foi deixada livre para manter a característica não determinística dos resultados e avaliar o desempenho do modelo em ambiente não controlado.

Como método de comparação de desempenho utilizou-se duas referências:

- O valor, de acordo com a programação dos cenários reais da refinaria, de cada um dos componentes da função objetivo. Esta referência permite a avaliação do *scheduling* gerado.

- O percentual de soluções aceitas, segundo critérios estabelecidos na seção 5.3.1, geradas pela otimização utilizando algoritmos genéticos (AG) segundo modelo descrito na seção 5.5.6. Esta referência é usada para balizar a capacidade da modelagem proposta neste trabalho em resolver problemas de *scheduling*.

5.5.1. CEN01

Considerando a menor complexidade do cenário, era esperado que o percentual de soluções aceitas em CEN01 fosse maior que o alcançado no cenário utilizado para a obtenção dos parâmetros do modelo, CEN05. Tal hipótese foi confirmada e a Figura 25 apresenta a distribuição de soluções aceitas para o modelo de otimização usando PGLOGIQ.

CEN01 tem uma característica que implica em alguma dificuldade para a programação: a quantidade de óleo ácido. Os tanques do terminal TqT02 e TqT04 possuem relevante quantidade de óleo ácido PT12. Este mesmo óleo também é deslocado pela movimentação da LS1 quando o navio1 começa seu descarregamento. Este óleo puro não enquadra as restrições de acidez de carga e nem dos cortes DL e DP. Consequentemente, é preciso limitar sua participação na carga das unidades de destilação de maneira que seu efeito não seja restritivo. Essa limitação deve acontecer através da preocupação durante as movimentações para garantir sua diluição em tanque até valores de composição aceitáveis ou através do uso de tanque de injeção.

Figura 25 – Distribuição de soluções aceitas para CEN01

Foi selecionada uma das soluções ideais para apresentação e discussão de resultados da programação completa gerada para o cenário. As Figuras 26 e 27 apresentam os gráficos de Gantt e de estoques para esta solução (as figuras

estão alinhadas na margem direita para facilitar a relação temporal entre os gráficos e, conseqüentemente, a análise das atividades ao longo do horizonte).

O navio foi descarregado sem ultrapassar a data limite para sua liberação, não incorrendo em sobreestadia. A operação de descarga foi contínua e para tal foram utilizados somente dois tanques. O primeiro volume movimentado é o deslocamento da linha LS1 que está cheia do óleo ácido PT12 para o TqT02. Este tanque já tinha óleo ácido, mantido na concentração do inventário inicial, e com este recebimento, a concentração aumenta, no entanto, não existe restrição de acidez para os tanques do terminal. Essa movimentação é tão válida quanto a seleção de qualquer dos outros tanques. Outras decisões de movimentação mantem este tanque no recebimento até que esteja cheio, quando passa a ser alocado o TqT03. Ao final do processo de descarregamento, a LS1 permanece cheia com o óleo PT09, recebido do navio1. No instante que o navio começa a descarregar, o terminal está com seu estoque total significativamente baixo. O atraso no recebimento deste navio implicaria em deixar o terminal impossibilitado de bombear óleo para a refinaria em um curto horizonte de tempo.

As atividades do terminal utilizam todos os tanques disponíveis, no entanto estão mais concentradas nos tanques TqT02 e TqT03, isto acontece porque, para alocar o volume de óleo descarregado, a capacidade de dois tanques é suficiente. Todos os tanques da refinaria são utilizados para receber óleo do terminal. O DUTO1 esteve em operação por 136,8 h em todo o horizonte. A seleção dos tanques de movimentação foi contínua em boa parte das operações.

No tempo 96,6 começa um conjunto de movimentações que pode ser discutida. O recebimento de óleo na refinaria que se segue a partir deste momento pode, num primeiro momento, parecer uma troca de tanques desnecessária, pois o duto está destinando para TqR01, troca para TqR02 e volta para TqR01. No entanto, alguns fatores são analisados sobre esse conjunto de decisões, por exemplo: uma alternativa seria a retomada de operação do duto com recebimento no TqR02, no entanto essa decisão implicaria em retomar a movimentação do duto depois, como o tempo de operação do oleoduto é um objetivo que se sobrepõe ao número de troca de tanques, a evolução privilegia a solução que começa antes.

Figura 26 – Gráfico de gantt de uma solução ideal para CEN01

Figura 27 – Gráfico de estoque de uma solução ideal para CEN01

As unidades de destilação operam por todo o horizonte de programação com a vazão de carga mantida no valor programado. Houve boa utilização dos recursos de tancagem para carga das unidades, o que pode ser observado pelas barras laranjas vinculadas aos tanques da refinaria no gráfico de Gantt. Durante as primeiras 12 horas essa atividade é garantida pelas Pre-Def e a partir de então uma sequencia de alocações usando os tanques TqR02 e TqR-04 garante a carga da UDA-01 e dos tanques TqR03, TqR01, TqR03 e TqR02 garante a carga da UDA-02. Não houve decisão do modelo evolutivo para uso de tanque de injeção, pois foram encontradas soluções que minimizavam esse objetivo. As

atividades de preparação de tanque, representadas pelas barras cinza, foram corretamente executadas.

Sob o ponto de vista dos objetivos avaliados, a solução proposta pode ser comparada com a solução real programada pela refinaria. Esta comparação é apresentada na tabela 24.

Pelos dados apresentados na tabela 24 pode-se observar que a solução otimizada e a programada pela refinaria têm o valor de seus objetivos semelhantes com exceção do número de trocas de tanques envolvidos na atividade de movimentação terminal-refinaria e tempo de operação do duto.

Objetivo	Critério	Solução PG	Solução Real
Operação das UDAs	UDA-01	192 h	192 h
	UDA-02	192 h	192 h
Sobreestadia		0 h	0 h
Operação do duto		136,8 h	90 h
Troca de Tanque	Descarga Navio	1	1
	Bombeio Duto	8	5
	Recebimento Duto	7	3
	Carga de UDA-01	2	2
	Carga de UDA-02	3	3
Uso de tanque de injeção para UDA		0	0

Tabela 24 – Tabela comparativa dos objetivos entre programação real e modelo proposto para CEN01

Na programação da refinaria as movimentações do duto são concentradas no início do cenário, neste caso, as 90 horas de operação do duto estão concentradas nas primeiras 110 horas do horizonte. Este comportamento se deve à incerteza na previsão sobre a chegada de navios nos últimos dias do cenário, e se reflete na tendência a antecipar toda movimentação para a refinaria de maneira a garantir carga das unidades de processo. No entanto, no terminal é preciso lidar com o dilema de manter espaço livre na tancagem ou guardar óleo

em tanque para montar a mistura que será bombeada quando houver novo descarregamento. Na solução da PG, a evolução dispõe e usa todo o horizonte do cenário para fazer todas as movimentações.

Os gráficos que apresentam, para a solução evoluída pela PG, o perfil das principais variáveis acompanhadas para a aceitação do *scheduling* são encontrados no Apêndice A. Neles pode ser observado o atendimento às restrições de acidez apresentadas na seção 5.2.1.

Para demonstrar a estrutura do indivíduo clássico, é apresentado, como exemplo na tabela 25, o indivíduo que gerou esta solução, já excluídos os genes com instrução NOp. A tabela está estruturada em 3 colunas por uma opção de compactação do texto neste trabalho, mas a interpretação deve ser feita percorrendo as colunas linearmente, ou seja, no indivíduo clássico, a instrução descrita na primeira linha da segunda coluna (PipeTransfProp TqT03, TqR03, 0.8) representa o gene clássico seguinte ao gene clássico representado na última linha da primeira coluna (FeedCduMaxInj TqR02, TqR01, 0.4, UDA-02). O mesmo vale entre as colunas 2 e 3. As tarefas Pre-Def não são representadas na tabela.

No total, para gerar esta solução de *scheduling*, 63 genes quânticos geraram instruções executáveis. No entanto, como discutido na seção 4.4.3, o processo de construção do *schedule* testa cada uma das instruções, avaliando todas as restrições apresentadas na seção 5.2.1. A execução do indivíduo clássico é iterativa e ocorre até que, ao percorrer todo o indivíduo, nenhuma instrução é executada. Para exemplificar melhor: dado um indivíduo que começa a ser interpretado, suponha-se que na primeira iteração, a instrução PipeTransfMax(TqT04, TqR01) foi executada, mas a instrução FeedCduMax (TqR02, UDA-01) não pôde ser executada porque no horário associado a essa atividade, TqR02 está sendo utilizado no recebimento de óleo do terminal. Consequentemente, a instrução não será transferida para o objeto que representa o *schedule*. A interpretação do indivíduo segue esta primeira iteração tentando executar todas as instruções, até a última PipeTransfMax (TqT04, TqR04). Como houve atividades executadas na primeira iteração, o indivíduo clássico é novamente percorrido. Na segunda iteração, no horário associado à mesma instrução FeedCduMax (TqR02, UDA-01), TqR02 está liberado e atende todas as demais restrições. Consequentemente, nesta segunda iteração a atividade é executada e acontece em um horário diferente do que havia sido tentado na primeira iteração. Novas iterações acontecem até que o horizonte do cenário acabe ou mais nenhuma atividade possa ser executada. Mesmo

considerando que nem todas as instruções apresentadas na tabela 25 foram executadas, é possível observar que a maior parte das instruções representadas no indivíduo clássico usa o volume máximo disponível para a atividade (43 são “Max”), o que é uma característica bem-vinda do ponto de vista de continuidade operacional.

Instrução	Instrução	Instrução
PipeTransfMax (TqT04, TqR01)	PipeTransfProp (TqT03, TqR03, 0.80)	PipeTransfMax (TqT03, TqR02)
PipeTransfMax (TqT04, TqR04)	RcvItemProp (navio1 tank1, TqT02, 0.90)	FeedCduProp (TqR04, 0.80, UDA-02)
PipeTransfMax (TqT02, TqR04)	RcvItemMax (navio1 tank1, TqT02)	PipeTransfMax (TqT03, TqR02)
PipeTransfMax (TqT02, TqR04)	PipeTransfMax (TqT04, TqR04)	FeedCduMax (TqR03, UDA-02)
FeedCduMax (TqR02, UDA-01)	FeedCduMax (TqR04, UDA-02)	PipeTransfMax (TqT04, TqR04)
PipeTransfMax (TqT04, TqR04)	FeedCduMaxInj (TqR04, TqR02, 0.5, UDA-01)	RcvItemMax (navio1 tank1, TqT04)
PipeTransfMax (TqT03, TqR04)	FeedCduMax (TqR01, UDA-02)	PipeTransfMax (TqT04, TqR02)
PipeTransfMax (TqT04, TqR04)	FeedCduProp (TqR03, 0.50, UDA-01)	PipeTransfProp (TqT04, TqR03, 0.60)
FeedCduMax (TqR03, UDA-02)	PipeTransfMax (TqT03, TqR03)	FeedCduMax (TqR03, UDA-02)
PipeTransfMax (TqT02, TqR04)	FeedCduMax (TqR02, UDA-01)	PipeTransfMax (TqT04, TqR03)
FeedCduMax (TqR02, UDA-02)	PipeTransfMax (TqT04, TqR03)	PipeTransfProp (TqT04, TqR04, 0.60)
PipeTransfMax (TqT04, TqR03)	PipeTransfMax (TqT04, TqR03)	FeedCduPropInj (TqR04, TqR03, 0.5, 0.80, UDA-02)
FeedCduMax (TqR04, UDA-01)	FeedCduPropInj (TqR04, TqR03, 0.4, 0.80, UDA-02)	FeedCduProp (TqR02, 0.50, UDA-01)
PipeTransfMax (TqT03, TqR03)	FeedCduProp (TqR03, 0.60, UDA-01)	PipeTransfMax (TqT04, TqR02)
FeedCduPropInj (TqR04, TqR03, 0.5, 0.80, UDA-02)	PipeTransfMax (TqT03, TqR03)	PipeTransfProp (TqT03, TqR02, 0.80)
PipeTransfMax (TqT04, TqR04)	RcvItemMax (navio1 tank1, TqT03)	FeedCduPropInj (TqR04, TqR03, 0.5, 0.80, UDA-02)
FeedCduPropInj (TqR03, TqR04, 0.4, 0.80, UDA-02)	PipeTransfProp (TqT02, TqR01, 0.40)	PipeTransfMax (TqT03, TqR01)
PipeTransfMax (TqT04, TqR04)	FeedCduPropInj (TqR04, TqR02, 0.3, 0.80, UDA-02)	PipeTransfMax (TqT01, TqR01)
FeedCduMax (TqR04, UDA-02)	FeedCduPropInj (TqR03, TqR04, 0.3, 0.80, UDA-02)	PipeTransfMax (TqT03, TqR02)
RcvItemProp (navio1 tank1, TqT02, 0.80)	FeedCduPropInj (TqR04, TqR02, 0.4, 0.50, UDA-02)	PipeTransfProp (TqT04, TqR04, 0.80)
FeedCduMaxInj (TqR02, TqR01, 0.4, UDA-02)	PipeTransfMax (TqT02, TqR02)	PipeTransfMax (TqT04, TqR04)

Tabela 25 – Programa gerado pelo modelo evolutivo para CEN01

5.5.2. CEN02

A Figura 28 apresenta a distribuição de soluções aceitas para o modelo de otimização usando PGLOGIQ.

Este cenário apresentou o maior percentual de soluções aceitas, sendo quase 90% destas classificadas como ideais. Apesar da avaliação de maior complexidade comparativamente ao CEN01, CEN02 possui duas características que pode ser atribuída ao seu melhor desempenho: o perfil de concentração de óleo leve e o perfil contínuo do duto. O inventário inicial nos tanques, duto e linha submarina tem alta concentração de óleo com baixa acidez, ainda que o inventário útil não seja alto, o lastro de óleo leve facilita as diluições. O primeiro navio a ser descarregado também compartilha dessa característica. Somente devido ao segundo navio passa a existir óleo ácido no sistema. No entanto, as movimentações associadas ao deslocamento e processamento desse óleo, ficam mais no final do cenário, além do fato de que, para que se avalie a ordem de grandeza, a quantidade trazida no navio é menor que a que havia em CEN01. Com essas características não é um grande desafio encontrar condições de diluição que não violem os limites de propriedades. Como este cenário possui apenas uma parcela no duto e não existem Pre-Defs para este equipamento, é esperado que as atividades de movimentação do duto sejam mais fáceis.

Figura 28 - Distribuição de soluções aceitas para CEN02

Foi selecionada uma das soluções ideais da PG para apresentação de resultados e discussão sobre a programação completa gerada para o cenário. As Figuras 29 e 30 apresentam os gráficos de Gantt e de estoques para esta solução.

Os navios foram descarregados completamente e de maneira contínua dentro do período especificado. Foram utilizados 2 tanques para o navio1 e apenas um tanque para o navio2, de pequeno porte. Considerando a situação de estoque inicial e a capacidade dos navios movimentados, pode-se inferir que

este cenário de programação reflete um período de dificuldade na disponibilização de óleo, mas não um cenário de crise, uma vez que as vazões das unidades de processo estão na faixa normal de produção.

Quando o navio1 começa a descarregar, o terminal possui 3 tanques praticamente vazios, facilitando a operação. Quando o navio2 atraca, o terminal está com maiores restrições de tancagem uma vez que TqT04 está movimentando, TqT01 está cheio e TqT02 com boa ocupação. Com este cenário, somente duas opções são possíveis para a descarga: uso total de TqT04 ou parcial deste tanque e parcial de TqT02.

Figura 29 - Gráfico de gantt de uma solução ideal para CEN02

Figura 30 - Gráfico de estoque de uma solução ideal para CEN02

As atividades de movimentação para a refinaria se concentram mais ao final do horizonte de programação. Este comportamento pode ser atribuído fato de que no início do período, o terminal tem baixo inventário e a refinaria razoável

estoque de petróleo, sendo necessário recuperá-lo depois da hora 80. A seleção dos tanques de movimentação é contínua, mas ainda acontecem trocas de tanques duvidosas, como pode ser visto em torno da hora 130 no terminal entre TqT01, TqT02 e retorno ao TqT01 e também em torno da hora 38, envolvendo os tanques TqR03 e TqR04. Tanto os tanques do terminal como da refinaria são bem utilizados nas movimentações.

As unidades de destilação operam por todo o horizonte de programação com a vazão de carga mantida no valor programado. Todos os tanques da refinaria tiveram boa utilização para alimentar as unidades. Durante as primeiras horas essa atividade é garantida pelas Pre-Def e, a partir de então, uma sequencia de alocações usando os tanques TqR02 e TqR04 garante a carga da UDA-01 e dos tanques TqR01, TqR03 e TqR02 garante a carga da UDA-02. Não houve decisão para uso de tanque de injeção.

A tabela 27 apresenta a comparação desta solução ideal gerada pelo modelo evolutivo com a solução real da refinaria.

Objetivo	Critério	Solução PG	Solução Real
Operação das UDAs	UDA-01	192 h	192 h
	UDA-02	192 h	192 h
Sobreestadia		0 h	0 h
Operação do duto		130,9 h	75 h
Troca de Tanque	Descarga Navio	2	3
	Bombeio Duto	4	3
	Recebimento Duto	6	2
	Carga de UDA-01	2	2
	Carga de UDA-02	3	4
Uso de tanque de injeção para UDA		0	0

Tabela 26 - Tabela comparativa dos objetivos entre programação real e modelo proposto para CEN02

Pelos dados apresentados na tabela 26 pode-se observar que a solução otimizada e a programada pela refinaria têm o valor de seus objetivos

semelhantes com exceção do número de trocas de tanques envolvidos na atividade de movimentação terminal-refinaria e tempo de operação do duto.

Neste cenário também há mais operação do duto na solução otimizada do que na real sendo que, nesta última, as 75 horas do objetivo estão nas primeiras 110 horas. Devido ao inventário de óleo no cenário, na solução real também não houve movimentação de duto no início do período. O maior valor obtido no objetivo de duto para a solução otimizada pode explicar parte da diferença entre o número de trocas de tanque.

Os gráficos que apresentam, para a solução evoluída pela PG, o perfil das principais variáveis acompanhadas para a aceitação do scheduling são encontrados no Apêndice A. Neles pode ser observado o atendimento às restrições de acidez apresentadas na seção 5.2.1.

5.5.3. CEN03

Comparativamente aos cenários anteriores, CEN03 possui 24 horas a mais de horizonte de programação, o que aumenta sua complexidade e pode afetar o desempenho de soluções aceitas, pois se torna necessário um conjunto maior de decisões de movimentação para programar todo o horizonte. Tem grandes quantidades de óleo ácido, significativamente mais do que CEN01, mas também tem expressiva quantidade de óleo leve, facilitando a diluição. A evolução deve ser capaz de gerar as movimentações de mistura que resultem em adequada composição desses óleos para atendimento às restrições. Este cenário também tem a característica complicadora do grande volume a ser descarregado pelo navio em um período curto do horizonte. O desempenho do modelo evolutivo neste cenário, apresentado na Figura 31, reflete a maior dificuldade em lidar com estas características. O percentual de soluções aceitas foi menor e, dentre estas, o percentual de soluções ideais também foi menor do que em CEN01.

Figura 31 - Distribuição de soluções aceitas para CEN03

Foi selecionada uma das soluções ideais do PG para apresentação de resultados e discussão sobre a programação completa gerada para o cenário. As Figuras 32 e 33 apresentam os gráficos de Gantt e de estoques para esta solução.

O navio foi descarregado sem ultrapassar a data limite para sua liberação, não incorrendo em sobreestadia. A operação de descarga foi contínua, sendo necessários três tanques que operaram aproveitando o máximo de suas capacidades. O primeiro volume movimentado é o deslocamento da linha LS1 que está cheia do óleo PT06 para o TqT04 que segue recebendo o óleo do navio enquanto há espaço disponível. O destino seguinte é TqT02 e, por último TqT01, que não chega a atingir sua capacidade máxima. A mudança de *tanqueCN* acontece durante o recebimento em TqT02. O modelo evolutivo seguiu a orientação de minimizar o número de trocas de tanques nestas movimentações. No final do horizonte LS1 fica cheia com o óleo PT03, recebido do *navio1tanqueCN1* (esta solução descarregou primeiro *tanqueCN2* e depois *tanqueCN1*).

As atividades do terminal utilizam todos os tanques disponíveis e todos os tanques da refinaria são utilizados para receber óleo do terminal. Este comportamento sinaliza boa utilização de recursos pelo algoritmo evolutivo.

O DUTO1 esteve em operação por 140,8 h em todo o horizonte. A seleção dos tanques de movimentação entre o terminal e a refinaria teve número expressivo de trocas de tanque, especialmente no horário em torno do descarregamento do navio. Dois fatores podem contribuir para este aumento: a mistura dos tanques na refinaria para balancear composição dos óleos ácido e leve, minimizando o uso de tanque de injeção e, a tarefa Pre-Def do DUTO1 que criou uma parcela dentro do duto já no começo do horizonte. Como a definição do volume das atividades do duto considera o mínimo entre volume útil para envio no *TqTx*, espaço disponível para receber no *TqRx* e volume da parcela, um volume de parcela menor deve implicar em mais movimentações e, potencialmente, mais trocas de tanque pela escolha de outro recurso para executar a atividade (instrução). No primeiro fator considerado, as trocas de tanque são justificáveis, mas no segundo caso sinaliza que a heurística de decisão para movimentação do duto implementada no modelo deve ser mais explorada.

Figura 32 - Gráfico de gantt de uma solução ideal para CEN03

Figura 33 - Gráfico de estoques de uma solução ideal para CEN03

As unidades de destilação operam por todo o horizonte de programação com a vazão de carga mantida no valor programado. Como nos demais cenários, em CEN03 também houve boa utilização dos recursos de tancagem para carga das unidades, tanto do ponto de vista de diversidade de tanques como pela continuidade da operação de cada um. Durante as primeiras 24 horas essa atividade é garantida pelas Pre-Def e a partir de então uma sequência de alocações usando os tanques TqR04 e TqR02 garante a carga da UDA-01 e dos tanques TqR02, TqR01, TqR03 e TqR01 garante a carga da UDA-02. Não houve decisão do modelo evolutivo para uso de tanque de injeção, pois foram encontradas soluções que minimizavam esse objetivo. As atividades de preparação de tanque, representadas pelas barras cinza, foram corretamente executadas.

Sob o ponto de vista dos objetivos avaliados, a solução proposta pode ser comparada com a solução real programada pela refinaria. Esta comparação é apresentada na tabela 27.

Pela análise desta tabela, pode-se observar total atendimento aos critérios considerados críticos, ou seja, operação das UDAs e não penalização por sobreestadia. Neste cenário também a solução proposta pela PGLOGIQ operou o DUTO por mais tempo, chegando a 46% de horas a mais. Como já argumentado para os demais cenários, é esperado maior número de trocas de tanques aqui também, no entanto, a análise detalhada da solução permite a identificação de trocas desnecessárias.

Objetivo	Critério	Solução PG	Solução Real
Operação das UDAs	UDA-01	216 h	216 h
	UDA-02	216 h	216 h
Sobreestadia		0 h	0 h
Operação do duto		140,8 h	96 h
Troca de Tanque	Descarga Navio	2	4
	Bombeio Duto	7	4
	Recebimento Duto	7	3
	Carga de UDA-01	2	3
	Carga de UDA-02	4	2
Uso de tanque de injeção para UDA		0	0

Tabela 27 - Tabela comparativa dos objetivos entre programação real e modelo proposto para CEN03

Os gráficos que apresentam, para a solução evoluída pela PGLOGIQ, o perfil das principais variáveis acompanhadas para a aceitação do *scheduling* são encontrados no Apêndice A.

5.5.4. CEN04

Ao contrário da expectativa inicial, CEN04 se revelou o cenário mais desafiador para a evolução. A Figura 34 apresenta a distribuição de soluções aceitas para CEN04. Algumas considerações podem ser feitas para entender este comportamento.

O maior horizonte implica em mais atividades uma vez que, por exemplo, para garantir a carga das unidades de destilação é necessário mais petróleo na refinaria, necessitando movimentá-lo do terminal e, possivelmente mais tanques envolvidos. Há também o fato de que este cenário manipula uma quantidade muito grande de petróleo. No início do horizonte, o inventário do terminal está quase no limite de sua capacidade o que faz com que, para conseguir receber o descarregamento correto do navio, exista a necessidade de muita movimentação de duto. Por último há o fato de que DUTO1 tem originalmente 2 parcelas o que, pela discussão apresentada em 5.5.3 deve levar a mais instruções de movimentação.

Figura 34 - Distribuição de soluções aceitas para CEN04

Foi selecionada uma das soluções ideais do PG para apresentação de resultados e discussão sobre a programação completa gerada para o cenário.

As Figuras 35 e 36 apresentam os gráficos de Gantt e de estoques para esta solução. A observação destas figuras evidencia a grande movimentação entre o terminal e a refinaria durante todo o horizonte do cenário. No início do horizonte as atividades do duto eram mais contínuas, com maior volume associado. Ao redor da hora 80, quando o navio inicia o descarregamento, o terminal ainda possui considerável quantidade de petróleo em estoque e os quatro tanques são necessários para conseguir receber todo o volume. Estas atividades são executadas de forma contínua e deixam o terminal no limite de sua capacidade.

A partir deste momento, o modelo evolutivo gera diversas atividades de movimentação do duto para atender aos objetivos de operação das UDAs e duto. No entanto, estas atividades não tem boa continuidade operacional, pois existem diversas trocas de tanque, como pode ser visto entre as horas 90 e 100, onde a sequencia de bombeio é dada por TqT02 e TqT04 sem aproveitamento máximo de suas capacidades. Dado que este cenário não manipula óleos que infrinjam restrições de qualidade, as decisões deveriam priorizar a minimização de trocas de tanques. O mesmo padrão de comportamento pode ser observado a partir da hora 140. Alguns testes, feitos com este cenário sem o uso de Pre-Def e com parcela única no duto, apresentaram comportamento de maior continuidade operacional. No entanto estes testes não são apresentados nesta dissertação porque foram realizados em condições que não refletiam o cenário real.

Figura 35 - Gráfico de gantt de uma solução boa para CEN04

Figura 36 - Gráfico de estoque de uma solução boa para CEN04

As unidades de destilação operaram por todo o horizonte do cenário. A despeito do mau desempenho no uso dos tanques nas movimentações de duto, a utilização destes recursos para a atividade de carga foi contínua com a utilização de todo o volume útil disponível. No entanto, como este volume não representava toda a capacidade do tanque, foi necessária maior troca de tanques para manutenção da carga. Durante as primeiras 4 horas a carga das unidades é garantida pelas Pre-Def e, a partir de então, uma sequência de alocações usando os tanques TqR01 e TqR03 garante a carga da UDA-01 e dos tanques TqR02, TqR04, TqR02, TqR04, TqR01 e TqR02 garante a carga da UDA-02.

Sob o ponto de vista dos objetivos avaliados, a solução proposta pode ser comparada com a solução real programada pela refinaria. Esta comparação é apresentada na tabela 28.

Pela análise desta tabela, pode-se observar atendimento aos critérios considerados críticos, ou seja, operação das UDAs e não penalização por sobreestadia. O número de trocas de tanques entre as atividades de duto foi significativamente maior, impactada pelo expressivo maior tempo de operação do duto e refletindo a consequência da geração de parcelas pequenas.

Objetivo	Critério	Solução PG	Solução Real
Operação das UDAs	UDA-01	240 h	240 h
	UDA-02	240 h	240 h
Sobreestadia		0 h	0 h
Operação do duto		197,9 h	144 h
Troca de Tanque	Descarga Navio	3	2
	Bombeio Duto	17	5
	Recebimento Duto	11	4
	Carga de UDA-01	2	2
	Carga de UDA-02	6	4
Uso de tanque de injeção para UDA		1 (Pre-Def)	1

Tabela 28 - Tabela comparativa dos objetivos entre programação real e modelo proposto para CEN04

Os gráficos que apresentam, para a solução evoluída pela PGLOGIQ, o perfil das principais variáveis acompanhadas para a aceitação do *scheduling* são encontrados no Apêndice A.

5.5.5. CEN05

Como previsto originalmente, CEN05 se confirmou como um cenário de alta complexidade devido às características de horizonte de programação, além de volume e configuração dos navios. Uma característica positiva é que o elenco de petróleo possui quantidade significativa de óleo ácido somente em um dos tanques, o qual é consumido através das Pre-Def. Dessa forma as atividades não são afetadas por restrições de qualidade. A Figura 37 apresenta a distribuição percentual de soluções aceitas para este cenário.

Figura 37 - Distribuição de soluções aceitas para CEN05

É observado um desempenho um pouco melhor em relação ao CEN04, que possui a mesma duração e características semelhantes de elenco. Essa diferença pode ser associada ao seu inventário inicial de petróleo ser significativamente menor o que facilita as atividades de descarregamento de navio e não sobrecarrega a movimentação do duto como aconteceu em CEN04.

Foi selecionada uma das soluções ideais da PG para apresentação de resultados e discussão sobre a programação completa gerada para o cenário. As Figuras 38 e 39 apresentam os gráficos de Gantt e de estoques para esta solução.

O descarregamento dos navios foi realizado com aproveitamento máximo da capacidade dos tanques, maximizando a continuidade operacional. Todos os tanques do terminal estiveram envolvidos em pelo menos um navio, sinalizando boa utilização dos recursos por parte do algoritmo evolutivo. No entanto, um exemplo de decisão que pode ser questionada acontece em torno da hora 110,

pois o TqT04, usado no recebimento, estava bombeando para o terminal, parou para receber uma hora de navio e voltou a bombear. Uma decisão mais adequada seria receber no TqT02.

Após a hora 170 (aproximadamente) podem ser observadas trocas desnecessárias entre os tanques TqR02 e TqR04 para recebimento de óleo do duto.

As unidades de destilação operam por todo o horizonte de programação com a vazão de carga mantida no valor programado. Como nos demais cenários, em CEN05 também houve boa utilização dos recursos de tancagem para carga das unidades, tanto do ponto de vista de diversidade de tanques quanto pela continuidade operacional de cada atividade.

Figura 38 - Gráfico de gantt de uma solução ideal para CEN05

Figura 39 - Gráfico de estoque de uma solução ideal para CEN05

Sob o ponto de vista dos objetivos avaliados, a solução proposta pode ser comparada com a solução real programada pela refinaria. Esta comparação é apresentada na tabela 29. Em CEN05 também se pode observar o mesmo padrão obtido nos outros cenários, ou seja, adequado atendimento aos critérios de UDA e navio, adequado número de trocas de tanque para manutenção da carga das unidades, maior movimentação de duto e mais trocas de tanque do que o esperado nas atividades envolvendo duto.

Os gráficos que apresentam, para a solução evoluída pela PGLOGIQ, o perfil das principais variáveis acompanhadas para a aceitação do *scheduling* são encontrados no Apêndice A.

Objetivo	Critério	Solução PG	Solução Real
Operação das UDAs	UDA-01	240 h	240 h
	UDA-02	240 h	240 h
Sobreestadia		0 h	0 h
Operação do duto		177,1 h	155 h
Troca de Tanque	Descarga Navio	5	5
	Bombeio Duto	6	5
	Recebimento Duto	9	5
	Carga de UDA-01	3	3
	Carga de UDA-02	3	4
Uso de tanque de injeção para UDA		1(Pre-Def)	1

Tabela 29 - Tabela comparativa dos objetivos entre programação real e modelo proposto para CEN05

5.5.6. Comentários Gerais

Algumas observações comuns nos casos estudados podem ser feitas:

- O modelo evolutivo foi capaz de gerar soluções ideais em todos os 5 cenários estudados.

- Os casos originalmente considerados mais desafiadores se confirmaram como tal.
- O inventário original de petróleo se mostrou influente no desempenho da evolução.
- A divisão de atividades no duto se mostrou influente no desempenho da evolução.
- A evolução levou a um maior percentual de instruções do tipo “Max” nos cromossomos clássicos.

De maneira geral, o comportamento das soluções ideais do modelo evolutivo foi equivalente em todos os cenários apresentados, o que pode ser traduzido nas seguintes características:

- Adequado atendimento aos objetivos críticos representados pela operação das UDAs em todo o horizonte de programação e descarregamento do volume completo do navio dentro da janela de tempo especificada.
- Boa utilização dos recursos de tancagem para as atividades de descarregamento de navio e carga das unidades de processo: tanto sob o aspecto de envolver todos os tanques quanto sob o aspecto de priorizar continuidade operacional quando não é identificado benefício em troca dos recursos.
- Maior operação do duto, comparativamente à solução real do cenário, como discutido na seção 5.5.1. Este comportamento advém da necessidade de atender o objetivo de operação do duto que, indiretamente, ajuda no atendimento ao objetivo de demanda das UDAs. O ônus observado é o maior número de troca de tanques. As características dos cenários impactadas pela movimentação do duto (maior tempo de operação do duto, descontinuidade temporal na alocação total e maior número de trocas de tanque) indicam provável necessidade de reavaliar as heurísticas utilizadas na movimentação e a modelagem do objetivo em si.
- Atividade de movimentação entre o terminal e a refinaria com maior número de trocas de tanque do que o esperado, possivelmente influenciada também pela existência de parcelas dentro do duto. Estas parcelas podem existir a partir do inventário inicial do duto ou

serem criadas a partir da programação de Pre-Def, sinalizando oportunidade de melhoria na modelagem proposta.

5.5.7. Comparação com Modelo em AG

O modelo em algoritmos genéticos utilizado é resultado de desenvolvimentos derivados do trabalho [3] com o qual ainda guarda forte relação. Uma explicação simplificada de sua representação e de seus parâmetros é dada nesta seção.

Neste modelo a representação dos cromossomos é baseada em segmentos com representação cujos tipos são:

- por ordem: ordem com que as atividades são executadas;
- inteiros: segmento com os recursos de origem e segmento com os recursos de destino de cada atividade;
- reais: o volume (máximo ou um percentual deste) que será movimentado em cada atividade;
- binário: uso de tanque de injeção em caso de atividade de carga de unidade.

Para determinar o número de tarefas é utilizada a equação:

$$N_{tarefas} = \frac{\sum VolItem}{VolLoteItem} + \frac{\sum VolOleoTQ}{VolLoteTQ} + \sum TempoCargaUD \quad (eq\ 5.1)$$

onde:

- $\sum VolItem$ representa o somatório dos volumes totais de cada item;
- $\sum VolOleoTQ$ representa o somatório dos volumes totais em cada tanque;
- $\sum TempoCargaUD$ representa o somatório do tempo que cada unidade de destilação está programada para operar;
- $VolLoteItem$ representa o volume mínimo de uma atividade de recebimento de item;
- $VolLoteTQ$ representa o volume mínimo de uma atividade de retirada de óleo do tanque;
- $N_{tarefas}$ representa o número máximo de tarefas que o cenário de programação pode executar. Este valor é usado para determinar o

comprimento (número de genes) de cada segmento do cromossomo.

Em [3] é encontrada a explicação do modo de atuação dos operadores de cruzamento e mutação utilizados no modelo AG atual. Todos os tipos de segmento (por ordem, binário e real) possuem operadores de cruzamento e mutação específicos.

Para garantir equilíbrio entre as características exploratórias e de aproveitamento dos melhores indivíduos, as taxas dos operadores foram configuradas de forma que variassem ao longo da evolução. Todos os valores de parâmetros do modelo em AG, inclusive suas faixas de variação, foram resultados de diversos testes realizados ao longo da pesquisa e desenvolvimento deste modelo, que se seguiram como continuidade de sua versão inicial (que originou o trabalho [3]) até os dias de hoje. A tabela 30 apresenta as faixas aplicadas para as taxas dos operadores do modelo evolutivo usando algoritmos genéticos. O método de seleção dos indivíduos utilizado foi torneio.

Parâmetro	Faixa (0-1)
Cruzamento: segmento ordem	0,9 ao 0,5
Mutação: segmento ordem	0,1 ao 0,4
Cruzamento: segmento real	0,8 ao 0,6
Mutação: segmento real	0,1 ao 0,4
Cruzamento: segmento binário	0,9 ao 0,6
Mutação: segmento binário	0,1 ao 0,4

Tabela 30 – Taxa de aplicação dos operadores do modelo evolutivo em AG

O tamanho da população foi mantido em 100 indivíduos. Outros parâmetros utilizados no modelo em AG são: taxa de *steady state* (percentual de indivíduos que permanecem na população a cada geração) igual a 50%; e a semente entre experimentos igual a 10%. Uma rodada do modelo em AG é composta por experimentos que se repetem tantas vezes quanto for configurado. No primeiro experimento todos os indivíduos da população são iniciados aleatoriamente e todo o processo evolutivo transcorre de acordo com as taxas e tipos de operadores e taxa de *steady state* até atingir o número total de gerações configurado. Quando este ciclo é finalizado, seus 10% melhores indivíduos são preservados para que componham a população do próximo experimento. Os

demais indivíduos para compor o total da população são iniciados aleatoriamente e todo o ciclo evolutivo recomeça com todas as taxas voltando aos seus valores originais. A rodada finaliza quando todos os experimentos tiverem sido completados.

Inicialmente os testes foram realizados utilizando como critério de parada as mesmas 240mil avaliações feitas pela PGLOGIQ de maneira que no AG esse valor se representasse na relação: tamanho da população vezes número de gerações vezes número de experimentos. Sendo assim, utilizou-se 400 gerações e 6 experimentos. A taxa de *steady state* em 50% significa que, na verdade, 120 mil avaliações são realmente feitas. Entretanto, optou-se por manter essa relação, pois nessa ordem de grandeza cada rodada do modelo em AG já demorava entre 1:30 h e 2:00 h. Do ponto de vista acadêmico este tempo pode ser considerado aceitável para uma execução, porém avaliando do ponto de vista de utilização do sistema em ambiente operacional, por toda a argumentação já feita neste trabalho a respeito da dinâmica e variabilidade da atividade de programação de produção, não seria viável aguardar cerca de duas horas por uma solução, especialmente sob o risco dela não ser considerada boa. Dessa forma, optou-se por também realizar execuções com 200 gerações e 3 experimentos que demoravam entre 20 e 25 minutos, mesma faixa de duração das rodadas em PGLOGIQ.

Este cromossomo é uma representação indireta da solução, uma vez que ele por si próprio não representa um *schedule*. Portanto, era necessário decodificá-lo para fazer a interpretação do indivíduo em uma solução representativa do problema real.

A decodificação de cada indivíduo é feita a partir da leitura sequencial de todos os genes do segmento de ordem das tarefas. Uma vez identificado tipo de tarefa, são lidos os respectivos recursos de origem e destino e o volume associado à tarefa. A Figura 40 apresenta o ciclo de evolução do algoritmo para o problema proposto.

Para haver coerência na comparação entre as modelagens em AG e PGLOGIQ foi garantido que todas as premissas, restrições, dados de entrada e a função de avaliação seriam as mesmas em ambos os casos.

Figura 40 – Ciclo de evolução do modelo em algoritmos genéticos

5.5.7.1. Comparação de Desempenho sobre Soluções Aceitas

As execuções usando os modelos de otimização baseados em programação genética e em algoritmos genéticos são avaliadas segundo os critérios adotados para soluções aceitas definidos na seção 5.4.1. A Figura 41 apresenta a distribuição percentual de soluções aceitas usando cada um dos seguintes algoritmos evolutivos:

- PGLOGIQ se refere aos resultados do modelo de otimização baseado em programação genética implementado nesta dissertação;
- AG240mil se refere aos resultados do modelo de otimização baseado em algoritmos genéticos com 400 gerações e 6 experimentos;
- AG60mil se refere aos resultados do modelo de otimização baseado em algoritmos genéticos com 200 gerações e 3 experimentos.

Figura 41 – Comparação do percentual de soluções aceitas em cada cenário usando modelagens em PGLOGIQ, AG 240 mil e AG 60 mil avaliações.

Pela Figura 41 pode-se observar que a modelagem usando PGLOGIQ teve melhor desempenho no percentual total de soluções aceitas em todos os cenários estudados. O tamanho desse ganho, comparativamente ao modelo evolutivo AG 240 mil, foi mais expressivo nos cenários CEN01, CEN02 e CEN04 e menos significativo nos cenários CEN03 e CEN05. Em relação ao AG 60 mil o benefício da modelagem usando PGLOGIQ foi sensível em todos os cenários.

CEN02 se mostrou o cenário mais fácil de ser resolvido em qualquer das modelagens. A diferença mais expressiva entre este e os demais casos está na uniformidade do duto neste cenário. Uma vez que no inventário inicial de DUTO1 existe apenas uma parcela e que não existe Pre-Def, o modelo evolutivo não teve que lidar, desde o início da evolução, com atividades limitadas pelo volume de pequenas parcelas. Dessa forma, as decisões da evolução poderiam trabalhar com parcelas de volume maior, podendo ser a própria capacidade do duto. A criação de parcelas, se existisse, seria uma consequência da evolução e não uma necessidade inicial.

CEN04 se mostrou o cenário mais desafiador, no qual coexistiam características de parcelas dentro do duto e grande inventário no terminal, explicitando o menor desempenho do modelo ao lidar com esses fatores.

Como esperado, comparativamente ao modelo em AG com 60 mil avaliações, o modelo em AG com 240 mil avaliações teve maior percentual de soluções aceitas em todos os cenários. Este desempenho pode ser explicado

por dois fatores: maior número de gerações por experimento; e maior número de experimentos, que faz com que a cada ciclo de 400 gerações, 90% da população seja iniciada novamente. A primeira característica permite que cada indivíduo possa evoluir mais e a segunda característica permite a busca em maior área do espaço de solução, além de contribuir para evitar mínimos locais.

A principal avaliação feita com a comparação entre as três modelagens é que a PGLOGIQ apresentou desempenho superior a uma metodologia que já se mostrou aplicável a resolver problemas de *scheduling* de refinaria e sobre a qual ainda são feitos desenvolvimentos. Pela Figura 41, pode-se observar que, dos cinco cenários estudados, a PGLOGIQ foi significativamente melhor em três e ligeiramente melhor nos outros dois. Dessa forma, pode-se concluir que a metodologia desenvolvida nesta dissertação é válida e promissora para a resolução de problemas de *scheduling* de petróleo em refinarias.

6

Conclusões e Trabalhos Futuros

O objetivo do trabalho apresentado foi mostrar a aplicabilidade de um modelo baseado em programação genética para auxiliar a elaboração da programação de petróleo em uma refinaria cujo escopo de decisão envolve o descarregamento de navios no terminal, movimentação de óleo entre o terminal e a refinaria e a manutenção da carga das unidades de destilação atmosférica.

A modelagem proposta considerou uma série de condições operacionais como: restrições de qualidade de correntes, restrições de lastro e capacidade de equipamentos, tempo de preparação de tanque, tempo mínimo de duração de uma atividade de carga da UDA e tempo mínimo de duração de uma atividade de movimentação entre o terminal e a refinaria.

A evolução foi orientada, hierarquicamente, pelas variáveis críticas do *scheduling*, ou seja, operação ininterrupta das UDAs e descarregamento do volume completo dos navios dentro da janela de tempo permitida, e por variáveis que refletem itens de preocupação do programador de produção, ou seja, operação contínua do duto que movimenta entre o terminal e a refinaria e melhor utilização dos recursos (equipamentos), que se reflete no número de trocas de tanques desnecessárias e quantidade de atividades com tanque de injeção.

Para desenvolver o modelo foram feitos estudos sobre as estruturas de representação da programação genética e, a partir deles, observou-se que a representação mais adequada deveria ser desenvolvida usando conceitos de programação genética linear e programação genética orientada à gramática. A programação genética linear permitiu a execução direta do programa gerado, onde cada gene representou uma instrução equivalente a uma atividade de *scheduling* possivelmente executada. O conceito de programação orientada à gramática permitiu o desenvolvimento de uma linguagem específica de domínio para o problema proposto e sua formalização através de uma sintaxe que garantiu que todas as instruções criadas pelo algoritmo evolutivo seriam válidas. Essas instruções poderiam não ser executadas por não atender a alguma restrição, mas sua estrutura certamente estaria correta.

Com a representação adotada, essa dissertação propõe um modelo inovador (PGLOGIQ) e, através deste, habilita a programação genética a

resolver uma classe de problemas como o *scheduling* de petróleo de uma refinaria, reconhecidamente de grande complexidade. Para se ter uma ideia do tamanho do problema, pode-se considerar o gene quântico deste modelo, as funções e seus respectivos argumentos e determinar que, para gerar cada gene clássico, existem 449 combinações possíveis, uma vez que o modelo lida com 9 funções, onde cada uma tem seus argumentos e cada argumento representa algumas opções de elementos terminais. Por exemplo, somente a função `FeedCDUMax(tnqR,cdu)` pode originar 8 instruções diferentes dependendo dos terminais observados. Um indivíduo clássico, que efetivamente representa uma solução de *schedule*, possui $128 \times 449 = 57.472$ combinações possíveis.

Duas grandes vantagens da representação proposta são justamente a garantia de geração apenas de instruções válidas e o fato de que o indivíduo gerado é de fácil interpretação para um observador da área, uma vez que cada gene clássico representa uma instrução clara da programação de petróleo. A capacidade de análise e entendimento da solução pelo especialista a torna mais amigável e, conseqüentemente, facilita a aceitação da técnica pelo usuário.

O passo seguinte foi desenvolver a linguagem e a sintaxe propostas de maneira a representar as atividades de programação e os equipamentos da refinaria envolvidos como as funções e terminais da gramática.

A partir desta etapa foi possível identificar o melhor conjunto de parâmetros para a modelagem proposta. Uma observação importante feita neste levantamento é que a PGLOGIQ se mostrou suficientemente robusta para não ser prejudicada pela existência de elementos terminais desnecessários no seu processo evolutivo, ou seja, ela pode ser configurada de maneira flexível sem prejuízo ao seu desempenho.

Os cenários utilizados como estudos de caso foram selecionados de forma a permitir a avaliação do desempenho do modelo em diferentes graus de complexidade e representar boa diversidade de configurações.

Os resultados obtidos se mostraram bastante promissores. Em todos os cenários a modelagem proposta foi capaz de encontrar soluções aceitáveis. Nos cenários mais complexos esse percentual esteve em torno de 25% das rodadas e, nos cenários mais simples, acima de 50%, chegando a 90% em um dos casos. A combinação de alto inventário de petróleo, navio de grande capacidade e existência de parcelas dentro do duto se mostrou uma característica bastante desafiadora, o que se refletiu principalmente em aproveitamento ineficaz das movimentações entre terminal e refinaria do ponto de vista de minimizar o número de trocas de tanque. Por ser uma abordagem nova, ainda há

considerável desenvolvimento a ser feito neste modelo que pode aumentar o percentual de soluções aceitas.

Os resultados foram comparados com o desempenho obtido por uma modelagem usando algoritmos genéticos e a PGLOGIQ obteve percentual de soluções aceitas ligeiramente melhor que o AG 240mil em 2 cenários e superior em outros 3 cenários. Utilizando o mesmo critério, na comparação com o AG 60mil, a PGLOGIQ teve desempenho significativamente superior em todos os cenários. Avaliando o desempenho em termos de tempo computacional, a PGLOGIQ teve um desempenho expressivamente superior, concluindo sua evolução em aproximadamente 20% do tempo que o modelo AG240mil para todos os cenários.

Uma solução de cada rodada de cenário teve o valor de seus objetivos comparado com os valores equivalentes da programação real do cenário. Os objetivos de atendimento à UDAs e descarregamento de navios foram equivalentemente atendidos. O objetivo de minimização de troca de tanques teve pior desempenho, mas parte da responsabilidade desse resultado é atribuída à maior movimentação do duto.

Considerando que não foram encontradas referências na literatura que tenham aplicado a modelagem proposta em problemas de *scheduling* e que esta dissertação não cobriu todos os desenvolvimentos que podem ser estudados para este caso, abre-se uma perspectiva de vários trabalhos futuros para melhorar os resultados obtidos, expandir este modelo ou aplicar uma modelagem equivalente a outros tipos de problemas de *scheduling*. Pode-se sugerir como trabalhos futuros os seguintes temas:

- Remodelagem da heurística de funcionamento do duto: Na representação atual, na qual o volume da atividade de movimentação através do duto é baseado nos tanques da refinaria e do terminal, além da parcela dentro do duto, a existência de uma parcela dentro do duto impede que o algoritmo evolutivo realize atividades que impliquem em aumento do volume da parcela dentro do duto, ou até mesmo o unifique. Como representado, os volumes das parcelas de duto tendem a permanecer iguais ou diminuir. Dessa forma, aumentam a necessidade de instruções para executar as movimentações, incorrendo em possibilidade de seleção de recursos que incorra no aumento do número de troca de tanques. Essa representação não reflete a heurística de decisão do programador de petróleo que, ao programar uma tarefa de duto,

toma duas decisões quase independentes que se repetem sequencialmente de maneira que a integridade volumétrica do duto é mantida: que tanque do terminal bombear para o duto e com que volume e, em que tanques receber o item de bombeio que está na extremidade mais próxima da refinaria. Uma alternativa poderia ser desvincular as decisões de bombeio para o duto e de recebimento no terminal, garantindo, em todos os intervalos de tempo, a consistência volumétrica do duto.

- Compactação das atividades do duto e UDA: Considerando duas soluções que param a UDA em algum instante do cenário, a melhor solução será a que fizer essa parada mais ao final do horizonte. Dada a variabilidade da atividade de programação, as condições iniciais ou premissas podem variar de forma que a execução de uma nova rodada de cenário no dia seguinte já não implicará em parada. Consideração semelhante é dada para a movimentação do duto, como discutido na seção 5.5.1. Nessa dissertação foram realizados testes utilizando um termo quadrático multiplicador dos objetivos de UDA e duto de maneira a penalizar menos as soluções com paradas mais próximas ao fim do horizonte. A comparação dos resultados demonstrou melhor desempenho do duto e UDA, porém houve penalização do objetivo de descarregamento de navio, resultando em piora do percentual de soluções aceitas. No entanto, é recomendado avaliar outras abordagens para atingir essa compactação.
- Investigar o uso de mais de uma população simultaneamente (*demes*) e, periodicamente, promover a troca dos melhores indivíduos entre as populações [28]. Nesta linha, poderia se avaliar duas abordagens: a mais convencional, onde a evolução de todas as populações é submetida aos mesmos operadores, critérios e função objetivo; ou uma abordagem onde, por exemplo, cada população seria orientada para uma priorização de objetivos diferentes.
- Investigar maior varredura do número de indivíduos nas populações; essa dissertação, com base no trabalho [11] investigou algumas opções na faixa de 6 indivíduos. É recomendado varrer maior espectro de tamanho de população para avaliar se ordens de

grandeza muito superiores causam impacto significativo na qualidade da solução, apesar do custo computacional.

- Tolerância de objetivos: pelo método de priorização de objetivos, um indivíduo que tenha parada de UDA igual a 1 hora e atraso de navio igual a 40 horas será melhor que outro indivíduo que tenha parada de UDA igual a 1,1 hora e nenhum atraso de navio. No entanto, o segundo indivíduo está mais próximo de evoluir para uma solução ideal do que o primeiro. É recomendado avaliar o uso de faixas de tolerância em cada objetivo de forma que os dois indivíduos citados acima, por exemplo, estariam empatados no critério de UDA e o segundo seria melhor avaliado por ter o objetivo de duto mais adequado. Este estudo implica no aumento de parâmetros do modelo e seu benefício deve ser avaliado criteriosamente
- Investigar o desenvolvimento e inclusão de funções auxiliares na linguagem específica de domínio. Estas funções não representam uma instrução diretamente relacionada com as atividades operacionais como “FeedCDUMax” ou “PipeTransfMax”, mas podem dar flexibilidade ao modelo na criação de regras. Por exemplo, funções aritméticas e condicionais poderiam ser usadas para relacionar a vazão de descarga de um navio com a de movimentação do duto e orientar a alocação de tanques do terminal.

7

Referências Bibliográficas

- [1] Supply Chain Council. SCOR 8.0 Implementation. Disponível em <www.archive.supply-chain.org>. Acesso em: 20 de maio 2012.
- [2] SINHA, A. K. Multi-Agent Based Petroleum Supply Chain Coordination□: A Co- Evolutionary Particle Swarm Optimization Approach. *Industrial Engineering*, p. 1349-1354, 2009.
- [3] DA CRUZ, D. D. S. Programação da Produção em Refinaria usando Algoritmos Genéticos: um Estudo para o Caso do *Scheduling* de Petróleos. Dissertação de Mestrado. Universidade Federal do Rio de Janeiro, Rio de Janeiro. 2007.
- [4] PAPAGEORGIOU, L. G. Supply chain optimisation for the process industries: Advances and opportunities. *Computers & Chemical Engineering*, v. 33, n. 12, p. 1931-1938, dez. 2009.
- [5] HASSIMOTTO, M. K. Otimização da programação de curto prazo de duto bidirecional de derivados de petróleo. Dissertação de Mestrado. Universidade de São Paulo, São Paulo. 2007.
- [6] MAGALHÃES, M. V. O. Refinery Scheduling. Tese de Doutorado. Imperial College, Londres. 2004.
- [7] MORO, L. F. L. Técnicas de Otimização Inteira Mista para o Planejamento e Programação de Produção em Refinarias de Petróleo. Tese de Doutorado. Universidade de São Paulo, São Paulo. 2000.
- [8] SIMÃO, L. M. Otimização da Programação da Produção em Refinarias de Petróleo utilizando Algoritmos Genéticos e Co-evolução Cooperativa. Dissertação de Mestrado. Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro. 2004.
- [9] OLIVEIRA, F.; ALMEIDA, M.R.; HAMACHER, S. Genetic algorithms applied to scheduling and optimization of refinery operations. VI ALIO/EURO Workshop on Applied Combinatorial Optimization, Buenos Aires. 2008.
- [10] RAMTEKE, M.; SRINIVASAN, R. Large-Scale Refinery Crude Oil Scheduling by Integrating Graph Representation and Genetic Algorithm. *Industrial & Engineering Chemistry Research*, p. 120328092932001, 26 mar. 2012.
- [11] DIAS, D. M. Programação Genética Linear com Inspiração Quântica. Tese de Doutorado. Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro. 2010.

[12] BODINGTON, C. E. Planning, Scheduling and Control Integration in the Process Industries, 1 edn, Mc Graw-Hill. New York, NY. 1995.

[13] PINTO, J. M.; GROSSMANN, I. E. Assignment and Sequencing Models for the Scheduling of Process Systems. Ann of Operational Research 81, 443-466. 1998.

[14] JOLY, M. Técnicas de Otimização Mista-Inteira para o Scheduling e Gerenciamento de Produção em Refinarias de Petróleo. Dissertação de Mestrado. Universidade de São Paulo, São Paulo. 1999.

[15] WU, N.; CHU, F.; CHU, C.; ZHOU, M. Short-Term Schedulability Analysis of Multiple Distiller Crude Oil Operations in Refinery With Oil. Analysis, v. 39, n. 1, p. 1-16, 2009.

[16] KARUPPIAH, R.; FURMAN, K. C.; GROSSMANN, I. E. Global optimization for scheduling refinery crude oil operations. Computers & Chemical Engineering, v. 32, n. 11, p. 2745-2766, nov. 2008.

[17] DURAN, M.A.; GROSSMANN, I. E. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Mathematical Programming, n.36, p.307-339. 1986.

[18] SAHARIDIS, G. K. D.; MINOUX, M.; DALLERY, Y. Scheduling of loading and unloading of crude oil in a refinery using event-based discrete time formulation. Ind. Computers and Chemical Engineering, n.33, p.1413–1426. 2009.

[19] MENDEZ, C. A.; CERDA, J.; GROSSMANN, I. E.; HARJUNKOSKI, I.; FAHL, M. State-of-the-art of Optimization Methods for Short-Term Scheduling of Batch Processes. Computers & Chemical Engineering v. 30,n. 6/7(May), pp. 913–946. 2006.

[20] BOK, J.; HEEMAN, L.; CHANG, J. W.; PARK, S. Development of Refinery Scheduling System Using Mixed Integer Programming and Expert System. North, v. 19, n. 4, p. 545-551, 2002.

[21] DAVE, D.; ZHANG, N. Hybrid methods using genetic algorithm approach for crude distillation unit scheduling. p. 445-450, 2003.

[22] NUNEZ, E.; Banks, E.R.; AGARWAL, P.; MCBRIDE, M.; LIEDEL, R. High Performance Evolutionary Computation. 2005 Users Group Conference (DOD-UGC'05), p. 308-313. 2005.

[23] KOZA, J. R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA, USA: MIT Press. 1992.

[24] KOZA J. R. Human-competitive results produced by genetic programming. Genetic Programming and Evolvable Machines, v. 11, n. 3-4, p. 251-284, 25 maio. 2010.

[25] BANZAF, W. et al. Genetic Programming – An Introduction; On the Automatic Evolution of Computer Programs and its Applications. San Francisco, CA, USA: Morgan Kaufmann, dpunkt.verlag. 1998

- [26] BATENKOV, D. Hands-On Introduction to Genetic Programming. Systems Engineering, v17, n1, p46-48. 2010.
- [27] DIAS, D. M. Síntese Automática de Programas para Microcontroladores Digitais por Programação Genética. Dissertação de Mestrado. Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro. 2005.
- [28] POLI, R.; LANGDON, W. B.; MCPHEE, N.F.; contribution KOZA, J.R. A Field Guide to Genetic Programming. Published via <http://lulu.com.2008>. Disponível em: <http://www.gp-field-guide.org.uk>
- [29] ARAUJO, S. G. Síntese de Sistemas Digitais Utilizando Técnicas Evolutivas. Tese de Doutorado. Universidade Federal do Rio de Janeiro, COPPE/UFRJ, Rio de Janeiro. 2004.
- [30] MOORE, M.; NARAYANAN, A. Quantum-inspired computing. Dept. Comput. Sci., Univ. Exeter, Citeseer, Exeter, UK. 1995.
- [31] PLATEL, M. D.; SCHLIEBS, S.; KASABOV, N. Quantum-Inspired Evolutionary Algorithm: A Multimodel EDA. IEEE Transactions on Evolutionary Computation, v. 13, n. 6, p. 1218-1232, dez. 2009.
- [32] DA CRUZ, A. V. A. Algoritmos Evolutivos com Inspiração Quântica para Problemas com Representação Numérica. Tese de Doutorado. Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro. 2007.
- [33] DIAS, D. M.; PACHECO, M. A. C. C. Toward a Quantum-Inspired Linear Genetic Programming Model. Evolutionary Computation, p. 1691-1698, 2009.
- [34] ADAMNS, T. P. Creation of simple, deadline, and priority scheduling algorithms using genetic programming. Genetic Algorithm and Genetic Programming at Stanford. 2002. Disponível em <http://www.genetic-programming.org/sp2002/Adams.pdf>
- [35] BUDIN, L.; JAKOBOVIC, D. Dynamic Scheduling with Genetic Programming. Lecture notes in Computer Science. 9th European Conference, EuroGP 2006. 2006.
- [36] OLTEAN M.; GROSAN, C.; DIOSAN, L.; MIHAILA, C. Genetic Programming with Linear Representation – a Survey. International Journal on Artificial Intelligence Tools. 2008.
- [37] HAUSCHILD, M.; PELIKAN, M. A Survey on Estimation of Distribution Algorithms. Acta Automatica Sinica, v. 33, n. 2, p. 0113, 2007
- [38] DEURSEN, A. V.; KLINT, P. Domain Specific Language Design Requires Feature Descriptions. Journal of Computing and Information Technology – CIT, v. 10, n. 1, p. 1-17, 2002.

APÊNDICE A

Dados de Entrada e Outros Gráficos das Soluções Apresentadas como Exemplo em Cada Cenário

A.1

CEN01

Dados Iniciais

Eqpto	PT01	PT02	PT03	PT04	PT05	PT09	PT10	PT12	Vol. Total (m ³)
Tanques									
TqT01	18,4	53,8	23,1		2,5	2,2			17668,6
TqT02	5,6	19	40,8					34,6	36417,8
TqT03	5,4	85,5	9,1						60559,6
TqT04	24,7					5,5		69,8	23673,1
TqR01	32,5	1,2	40,3		2,6	13,2		10,2	27317,8
TqR02	22,3	37,7	17,2		5,9	5,7		11,2	36033,4
TqR03	44,2	20,3	10,2	2,3	6,1	9,3		7,6	35800,2
TqR04	37,6		24,3	9,2	14,7	14,2			24200,3
Linha Submarina									
LS1								100	11667,0
Navios									
Navio1 tqCN1						100			66667,0
Duto									
DUTO1 Parcela1	7,3	24,3	38,2		0,3	1,4	0,3	28,3	19000,0

Tabela 31 – Dados iniciais de composição e volume para CEN01

Resultados da Solução Apresentada

Figura 42 – Gráficos de Acidez das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN01.

Figura 43 - Gráficos de Enxofre das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN01.

Figura 44 – Perfil de Rendimento para todos os cortes de destilação das UDAs em CEN01

A.2

CEN02

Dados de Entrada

Eqpto	PT01	PT02	PT03	PT04	PT05	PT09	PT10	PT11	PT12	Vol. Total (m ³)
Tanques										
TqT01	77,4		2,9		10,3	9,4				12291,3
TqT02	70,6			2,2	8,0	19,2				6864,3
TqT03	3,1		9,3	34	19,1	34,5				6917,6
TqT04	81,9					18,1				54232,3
TqR01	58,5		2,9	1,2	6,8	29	1,6			61678,7
TqR02	43,7		7,9	5,2	22,4	20,8				31109,8
TqR03	35		5,8	16,8	19,9	22,5				28452,0
TqR04	13,5		8,7	18,5	37,3	20,7		1,3		41956,3
Linha Submarina										
LS1			100							11667,0
Navios										
Navio1 tqCN1		100								66667,0
Navio2 tqCN1									100	45833,0
Duto										
DUTO1 Parcela1	7,3	24,3	38,2		0,3	1,4	0,3		28,3	19000,0

Tabela 32 - Dados iniciais de composição e volume para CEN02

Resultados da Solução Apresentada

Figura 45– Gráficos de Acidez das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN02

Figura 46 - Gráficos de Enxofre das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN02

Figura 47 – Perfil de Rendimento para todos os cortes de destilação das UDAs em CEN02

A.3

CEN03

Dados de Entrada

Eqpto	PT01	PT02	PT03	PT04	PT06	PT08	PT09	PT12	Vol. Total (m³)
Tanques									
TqT01	3,4					59,4		37,2	40884,3
TqT02	2,0		25,6				4,4	68,0	6938,3
TqT03	86,4	13,6							44268,5
TqT04	89,9						10,1		10241,9
TqR01	42,9	4,2	28,8				19,5	4,6	9802,8
TqR02	4,3	3,3	36,6				7,0	48,8	36450,2
TqR03	21,1	3,3	9,2				11,8	54,6	30389,9
TqR04	40,7	4,7	17,8				2,7	34,1	54834,6
Linha Submarina									
LS1					100				11667,0
Navios									
Navio1 tqCN1			100						75000,0
Navio1 tqCN2				100					58333,0
Duto									
DUTO1 Parcela1	10,1						89,9		19000,0

Tabela 33 - Dados iniciais de composição e volume para CEN03

Resultados da Solução Apresentada

Figura 48– Gráficos de Acidez das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN03

Figura 49 - Gráficos de Enxofre das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN03

Figura 50 – Perfil de Rendimento para todos os cortes de destilação das UDAs em CEN03

A.4

CEN04

Dados de Entrada

Eqpto	PT01	PT02	PT03	PT04	PT05	PT09	PT10	PT11	Vol. Total (m ³)
Tanques									
TqT01	10,6		11,3		40,9	37,2			42468,3
TqT02	2,3		8,4	19,5	69,8				45610,6
TqT03	3,0		9,3	33,9	19,2	34,6			60157,5
TqT04	38,6					56,9	4,5		61805,5
TqR01	41,2	1,4	37,1		1,7	11,4	4,0	3,2	60249,8
TqR02	10,6		21,1	62,0		6,3			29358,2
TqR03	57,9	1,1	18,6			11,9	3,3	7,2	10895,7
TqR04	65,4	2,0			5,7	12,5	8,9	5,5	23713,3
Linha Submarina									
LS1							100		11667,0
Navios									
Navio1 tqCN1	100								75000,0
Duto									
DUTO1 Parcela1	16,0		17,0		59,2	5,6		2,2	4583,3
DUTO1 Parcela2	2,2		8,4	69,9	19,5				14416,7

Tabela 34 - Dados iniciais de composição e volume para CEN04

Resultados da Solução Apresentada

Figura 51– Gráficos de Acidez das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN04

Figura 52 - Gráficos de Enxofre das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN04

Figura 53 – Perfil de Rendimento para todos os cortes de destilação das UDAs em CEN04

A.5

CEN05

Dados de Entrada

Eqpto	PT01	PT02	PT03	PT04	PT06	PT07	PT08	PT09	PT12	Vol. Total (m ³)
Tanques										
TqT01	11,0				12,2	9,9	36,8		30,1	6286,5
TqT02	2,0		25,6					4,4	68,0	6938,3
TqT03	86,4	13,6								17707,6
TqT04	89,9							10,1		8251,5
TqR01	73,4	6,5	4,5				5,8	6,2	3,6	60882,0
TqR02	2,9	4,7	36,6					7,0	48,8	32376,1
TqR03	15,7	1,5	4,2		6,6	5,3	19,9	5,4	41,4	19363,9
TqR04	40,7	4,7	17,8					2,7	34,1	9329,8
Linha Submarina										
LS1					100					11667,0
Navios										
Navio1 tqCN1			100							75000,0
Navio1 tqCN2				100						58333,0
Navio2 tqCN1	100								100	129167,0
Duto										
DUTO1 Parcela1	10,9				12,2	9,9	36,9		30,1	19000

Tabela 35 - Dados iniciais de composição e volume para CEN05

Resultados da Solução Apresentada

Figura 54– Gráficos de Acidez das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN05

Figura 55 - Gráficos de Enxofre das correntes de carga, DL, DP e RAT das unidades de destilação atmosférica em CEN05

Figura 56 – Perfil de Rendimento para todos os cortes de destilação das UDAs em CEN05