

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
INFORMÁTICA INDUSTRIAL

LIA YAMAMOTO

**UM MODELO BASEADO EM METAHEURÍSTICA PARA O
SEQUECIAMENTO DE BATELADAS EM REDES DUTOVIÁRIAS**

TESE DE DOUTORADO

CURITIBA

2009

LIA YAMAMOTO

**UM MODELO BASEADO EM METAHEURÍSTICA PARA O
SEQUECIAMENTO DE BATELADAS EM REDES DUTOVIÁRIAS**

Tese de doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Doutor em Ciências” – Área de Concentração: Informática Industrial.

Orientadora: Profa. Dra. Lúcia Valéria Ramos de Arruda

CURITIBA

2009

Ficha catalográfica elaborada pela Biblioteca da UTFPR – Campus Curitiba

Y19m	<p>Yamamoto, Lia</p> <p>Um modelo baseado em metaheurística para o sequenciamento de bateladas em redes dutoviárias / Lia Yamamoto. – 2009. 128 p. : il. ; 30 cm</p> <p>Orientador: Lúcia Valéria Ramos de Arruda Tese (Doutorado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial. Área de Concentração: Informática Industrial, Curitiba, 2009 Bibliografia: p. 123-28</p> <p>1. Agenda de execução. 2. Metaheurística. 3. Petróleo – Derivados. 4. Algoritmos genéticos. 5. Bateladas. 6. Oleodutos de petróleo. 7. Informática Industrial. I. Arruda, Lúcia Valéria Ramos de, orient. II. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial. Área de Concentração em Informática Industrial. III. Título.</p> <p>CDD 621.3</p>
------	--

Título da Tese N° 45:

“Um Modelo Baseado em Metaheurística para o Sequenciamento de Bateladas em Redes Dutoviárias”

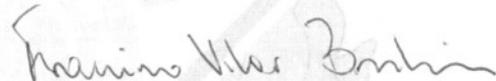
por

Lia Yamamoto

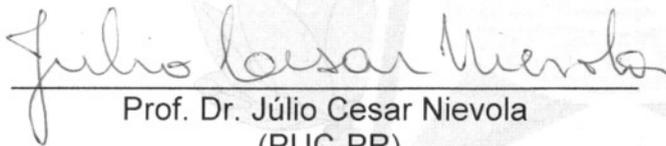
Esta tese foi apresentada, às 14h do dia 27 de outubro de 2009, como requisito parcial para a obtenção do título de DOUTOR EM CIÊNCIAS – Área de Concentração: Informática Industrial, pelo Programa de Pós Graduação em Engenharia Elétrica e Informática Industrial – CPGEI –, na Universidade Tecnológica Federal do Paraná – UTFPR. O trabalho foi aprovado pela Banca Examinadora, composta pelos professores:



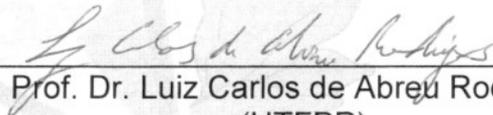
Prof^a. Dr. Lúcia Valéria Ramos de Arruda
(Orientadora - UTFPR)



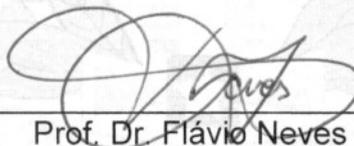
Prof. Dr. Francisco Vilar Brasileiro
(UFCG)



Prof. Dr. Júlio Cesar Nievola
(PUC-PR)



Prof. Dr. Luiz Carlos de Abreu Rodrigues
(UTFPR)



Prof. Dr. Flávio Neves Junior
(UTFPR)

Visto da coordenação:



Prof. Dr. Humberto Remígio Gamba
(Coordenador do CPGEI)

AGRADECIMENTOS

Muitas pessoas colaboraram e foram essenciais na realização deste trabalho.

Agradeço a professora Valéria pela orientação do trabalho e ajuda em horas difíceis de tomadas de decisão, ao professor Flávio pelo esforço atrás de suporte financeiro e ao Nikolas que muito me ajudou na parte de programação. Agradeço a todos os colegas do laboratório LASCA, e especialmente ao Felizari, Suelen, Leandro e Sérgio.

Aos membros da banca Prof. Dr. Francisco Vilar Brasileiro, Prof. Dr. Júlio Cesar Nievola, Prof. Dr. Luis Carlos de Abreu Rodrigues e Prof. Dr. Flávio Neves Junior pela disposição de examinarem este trabalho e pelas sugestões dadas.

À minha família, Mariana, Marcelo, Tamia e Yuji, fico grata pelo apoio, ajuda e compreensão.

Gostaria de agradecer a todos que de alguma forma contribuíram para a realização deste trabalho.

Este trabalho recebeu o apoio financeiro da Agência Nacional do Petróleo (ANP) e do CTPetro/Financiadora de Estudos e Projetos através do programa de recursos humanos para o setor de petróleo e gás natural - UTFPR/PRH10, CENPES/PETROBRAS (0050.0017859.05.3 e 0050.0027214.06.2)

RESUMO

YAMAMOTO, Lia. Um modelo baseado em metaheurística para o sequenciamento de bateladas em redes dutoviárias. 128 f. Tese – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2009.

O *scheduling* operacional de uma rede de distribuição de produtos derivados do petróleo é um problema combinatório complexo, de difícil resolução computacional. Este problema pode ser decomposto em três subproblemas de acordo com os elementos chaves do *scheduling*: alocação dos recursos, seqüenciamento das atividades, e determinação do tempo para a utilização dos recursos pelas atividades. O presente trabalho apresenta o desenvolvimento de um modelo para a solução do subproblema de seqüenciamento das atividades. O seu principal objetivo é propor uma solução baseada em metaheurística para a geração de ordenações das bateladas de produtos derivados do petróleo, criando um modelo de auxílio à tomada de decisão durante as operações de uma malha dutoviária. Da prática operacional, sabe-se que o seqüenciamento de bateladas tem grande influência no resultado final do *scheduling*. Além disto, a escolha da melhor sequencia é um problema multiobjetivo, dependente dos critérios considerados na escolha (atendimento à demanda ou à produção, ocupação dos dutos, etc.). Neste trabalho realizou-se a implementação de várias técnicas metaheurísticas para a solução do problema de seqüenciamento. Foram analisados os resultados alcançados com o uso de diferentes funções objetivo na otimização. Os resultados mostraram um bom desempenho das metaheurísticas estudadas no seqüenciamento, o que possibilitou a obtenção de boas soluções em um curto tempo computacional. Estudou-se mais profundamente o modelo baseado em algoritmo genético que permite fornecer um conjunto de soluções que representam diferentes opções de possíveis sequencias a serem escalonadas na rede. Este modelo foi, em seguida, estendido para tratar o caso de múltiplos objetivos. Desenvolve-se então uma abordagem baseada em MOGA (*Multiobjective Genetic Algorithm*) que apresentou bons resultados além de uma maior diversificação de sequencias quando comparado com a solução com objetivo único. Os modelos propostos foram validados em instâncias reais da rede de dutos brasileira e constituem uma ferramenta de auxílio à tomada de decisão operacional, resultando num uso mais eficiente da malha dutoviária estudada.

Palavras-chave: *Scheduling*. Sequenciamento. Metaheurística. Bateladas. Dutos.

ABSTRACT

YAMAMOTO, Lia. A model based on metaheuristic to the batches sequencing on the pipeline networks. 128 f. Tese – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2009.

The scheduling of activities to distribute oil derivate products through a pipe network is a complex combinatorial problem, that presents a difficult computational solution. This problem can be decomposed on three sub-problems according to the key elements of scheduling: assignment of resources, sequencing of activities, and determination of resource timing utilization by these activities. This work develops an approach to the sequencing sub-problem. The main objective is to develop an approach based on metaheuristic for the order generation of oil derivate products batches. The approach goals is to aid the operational decision making of the scheduling problem in a pipeline network. From the operational practice, the batches sequencing has great influence on the final scheduling result. Moreover the best order choice is a multiobjective problem and depends on the considered criteria (fulfillment of demand and production, pipeline occupation rate, etc.). Some metaheuristic techniques were implemented to the sequencing problem. Results in using different objective functions on sequencing optimization were also analyzed. The metaheuristic showed a good performance in sequencing, which resulted on good solutions in a small computational time. A model based on genetic algorithm was intensively studied and it has provided a set of solutions that means different options of possible sequences to be schedule in the pipeline network. This model was also extended to treat multiple objectives. A model based on MOGA (Multiobjective Genetic Algorithm) was developed and it showed good results and also good batches sequencing diversification. The proposed models were validated for real scenarios in the Brazilian pipeline network and have become a tool to aid the operational decision making, resulting in an improvement of the study of this pipeline network.

Keywords: Scheduling. Sequencing. Metaheuristic. Batches. Pipelines.

LISTA DE FIGURAS

Figura 1 – Malha dutoviária brasileira	22
Figura 2 – Espaço de estratégias de operação	29
Figura 3 – Classificação do sistema híbrido segundo a estratégia de controle	38
Figura 4 – As várias estruturas diferentes de vizinhança do VNS	48
Figura 5 – Sistemática de mudança das vizinhanças	48
Figura 6 – A escolha do caminho mais curto pela colônia de formigas	50
Figura 7 – Processo de <i>crossover</i> ou recombinação	54
Figura 8 – Processo de mutação	55
Figura 9 – Soluções Ótimas de Pareto para um problema com dois objetivos	60
Figura 10 – Rede em estudo	70
Figura 11 – Estrutura do <i>scheduling</i>	73
Figura 12 – Determinação dos tempos em que há a necessidade de envio e recebimento de produtos em uma área	74
Figura 13 – Determinação das janelas de tempo para os terminais de demanda e oferta	75
Figura 14 – Operação de reversão	78
Figura 15 – Operação de pulmão	79
Figura 16 – Representação de uma solução	85
Figura 17 – Tempo computacional médio obtido pelo processamento das diferentes técnicas metaheurísticas	88
Figura 18 – Resultados obtidos pelo processamento das diferentes técnicas metaheurísticas (cenário 1)	89
Figura 19 – Resultados obtidos pelo processamento das diferentes técnicas metaheurísticas (cenário 2)	89
Figura 20 – Tempo total do <i>scheduling</i> obtido pelo seqüenciamento usando diferentes funções objetivo	92
Figura 21 – Soma das taxas de utilização dos dutos obtidas pelo seqüenciamento usando diferentes funções objetivo	93
Figura 22 – Média e desvio padrão das taxas de utilização dos dutos obtidas pelas diferentes funções objetivo	93
Figura 23 – Número total de transições entre diferentes produtos	94
Figura 24 – Número total de violações ocorridas considerando as diferentes funções objetivo	95
Figura 25 – Número de iterações para o Algoritmo Genético	96
Figura 26 – Melhora do percentual de violações das janelas de tempo e transições entre produtos devido os diferentes pesos da função objetivo	97
Figura 27 – Número de indivíduos mantidos em novas populações do Algoritmo Genético	98
Figura 28 – Gráficos de Gantt para o cenário 1	100

Figura 29 – Gráficos de Gantt para o cenário 3	101
Figura 30 – Gráficos de Gantt para o cenário 1	103
Figura 31 – Método do ranqueamento	106
Figura 32 – Exemplo de superfície tridimensional	111
Figura 33 – Paralelepípedo definido por (m_1, m_2, m_3) e (M_1, M_2, M_3) , que limita a área da superfície	111
Figura 34 – Exemplificação da Área "A". Cada ponto está a uma distância σ_{share} dos vizinhos	112
Figura 35 – População=20 indivíduos	113
Figura 36 – População=50 indivíduos	113
Figura 37 – População=100 indivíduos	113
Figura 38 – População=150 indivíduos	114
Figura 39 – População=200 indivíduos	114
Figura 40 – Solução 1, 2 e 3 do cenário 1	116
Figura 41 – Solução 1, 2 e 3 do cenário 2	117

LISTA DE TABELAS

Tabela 1 – Diferenças entre <i>scheduling</i> e <i>planning</i>	27
Tabela 2 – Tabela resumo da literatura sobre aplicação em dutos	35
Tabela 3 – Relações entre o processo físico e o processo de otimização funcional ...	45
Tabela 4 – Tabela de geração de subconjuntos	58
Tabela 5 – Exemplo de dados de saída do módulo de alocação	75
Tabela 6 – Tabela de demanda de produtos para o cenário 1	82
Tabela 7 – Tabela de demanda de produtos para o cenário 2	82
Tabela 8 – Tabela de demanda de produtos para o cenário 3	82
Tabela 9 – Tabela de produção dos derivados para o cenário 1	83
Tabela 10 – Tabela de produção dos derivados para o cenário 2	83
Tabela 11 – Tabela de produção dos derivados para o cenário 3	83
Tabela 12 – Algumas bateladas pertencentes à carteira de bateladas do cenário 1	84
Tabela 13 – Nomenclatura das metaheurísticas implementadas	85
Tabela 14 – Parâmetros dos algoritmos	86
Tabela 15 – Nomenclatura dos objetivos usados	90
Tabela 16 – Pesos da função objetivo	96
Tabela 17 – Parâmetros dos algoritmos	98
Tabela 18 – Número de vizinhança para o VNS	102
Tabela 19 – Número de aperfeiçoamentos para cada nova solução do VNS	102
Tabela 20 – <i>Características das Soluções</i>	115

SUMÁRIO

1	INTRODUÇÃO	21
1.1	MOTIVAÇÃO	23
1.2	OBJETIVO	24
1.3	CONTRIBUIÇÕES	25
1.4	ORGANIZAÇÃO DO TRABALHO	26
2	LEVANTAMENTO BIBLIOGRÁFICO	27
2.1	<i>SCHEDULING</i>	27
2.1.1	Revisão Bibliográfica	31
2.1.2	Métodos Híbridos	33
3	FUNDAMENTAÇÃO TEÓRICA	37
3.1	SISTEMAS HÍBRIDOS	37
3.2	METAHEURÍSTICAS	39
3.2.1	Métodos de Busca Local	41
	<i>Greedy Randomized Adaptive Search Procedures</i>	41
	<i>Simulated Annealing</i>	44
	Busca Tabu	46
	<i>Variable Neighborhood Search</i>	47
3.2.2	Métodos de Busca Populacional	49
	Colônia de Formigas	49
	Algoritmo Genético	52
	<i>Scatter Search</i>	56
3.3	TÉCNICAS DE OTIMIZAÇÃO MULTIOBJETIVO	59
3.3.1	Problemas Multiobjetivos (PMO)	59
3.3.2	Otimalidade de Pareto	60
	Definição de soluções dominadas	61
3.3.3	Algumas Técnicas Importantes de Aproximação	61
	Métodos Evolucionários	64
4	REDE DE DUTOS	69
4.1	MODELO DA REDE	69
4.2	ABORDAGEM HÍBRIDA ADOTADA	73
5	MODELOS PROPOSTOS	81
5.1	SEQÜENCIAMENTO	81
5.1.1	Seqüenciamento de Bateladas através de Modelos baseados em Metaheurísticas	82
	Comparações dos Modelos baseados em Metaheurísticas	84
	Análise e Comparações de Diferentes Funções Objetivo	90
	Algoritmo Genético	95
	Aplicações do AG	99
	VNS	101
	Aplicações do VNS	102
	Conclusões	104
6	SEQÜENCIAMENTO DE BATELADAS ATRAVÉS DO MODELO MOGA	105

6.1 MOGA	105
6.1.1 Discussões	108
6.1.2 Aplicações	114
7 CONCLUSÕES E TRABALHOS FUTUROS	119
7.1 CONCLUSÕES	119
7.2 CONTRIBUIÇÕES	120
7.3 TRABALHOS FUTUROS	121
REFERÊNCIAS	123

1 INTRODUÇÃO

Devido à busca constante de uma maior eficiência na utilização dos recursos das indústrias, há uma grande preocupação e necessidade de se implantar boas técnicas/práticas para o planejamento e programação da produção, bem como para a distribuição de bens e produtos. Segundo Rejowski e Pinto (2003), a indústria petrolífera, em especial, pode ter um ganho econômico considerável através de uma boa execução e prática de um planejamento (planning) e uma programação de atividades (scheduling) otimizada.

O refino de petróleo possui características técnicas peculiares que se refletem em atributos econômicos marcados pelas economias de escala e escopo. Por ser uma atividade integrada à produção de petróleo e à distribuição de derivados, a escala ótima de refino em um determinado mercado depende diretamente das escalas destas duas outras atividades, no Brasil. É preciso combinar estas três atividades eficientemente a fim de otimizar a cadeia de suprimentos do petróleo e atender da melhor maneira possível o mercado desejado, gerando lucros à empresa. Por outro lado, o mercado apresenta um consumo crescente e uma demanda por produtos de melhor qualidade com rapidez e segurança.

Visando a otimização dos processos de produção e distribuição dos derivados do petróleo, faz-se necessário o planejamento e a programação das operações de transporte e armazenagem desses produtos. Entre os diversos modais de transporte, o transporte de derivados através de dutos mostra-se altamente confiável e econômico, principalmente para o transporte de grandes quantidades e em longas distâncias (SASIKUMAR et al., 1997).

O transporte através da malha de dutos proporciona aumento da eficiência e da segurança operacional, assim como a redução de custos. O transporte dutoviário possibilita agilidade, segurança e capacidade de fluxo na movimentação dos combustíveis, além de gerarem enormes ganhos econômicos e sociais, baseados na responsabilidade socioambiental (TRANSPETRO, 2009).

Objetivando o uso eficiente dos recursos dutoviários brasileiros faz-se necessária a análise da infra-estrutura existente no país. A rede dutoviária é muito concentrada na região Sudeste do país. Existem alguns projetos para melhorias e ampliação de portos, em curto prazo, mas estes projetos não serão suficientes para suportar um aumento em nove vezes das quantidades de derivados de petróleo transportados até 2015 (TAVARES, 2005). Talvez seja necessária

uma maior integração da rede de transportes existente, a partir da ampliação da rede dutoviária e de uma maior participação dos modais hidroviário e ferroviário para o transporte de derivados de petróleo. As limitações decorrentes de uma infra-estrutura de transporte e armazenamento de petróleo e derivados pouco desenvolvida são particularmente relevantes, tornando limitadas as condições de aplicação do princípio do livre-acesso determinado em lei (TAVARES, 2005).

O transporte do petróleo processado nas refinarias brasileiras, proveniente de importações ou da produção nacional é realizado por intermédio do sistema dutoviário ou do transporte aquaviário (marítimo ou interior, por meio de navios petroleiros ou balsas). Com relação ao transporte pelo modal dutoviário, os principais dutos, em extensão e movimentação de produtos, interligam terminais localizados na costa brasileira às refinarias. O Brasil conta com uma infra-estrutura de refinarias e dutos concentrada basicamente na Região Sudeste e na parte litorânea, como pode ser verificada na figura 1 (ANP, 2006) que mostra a malha dutoviária brasileira. O transporte de derivados é feito em sua maioria por barcos, trens e caminhões. No entanto, no estado de São Paulo (quadro inferior esquerdo da figura 1), onde se concentram a maior parte das refinarias, existe uma rede de dutos complexas por onde trafega 48% da produção de derivados do país (TRANSPETRO, 2009).

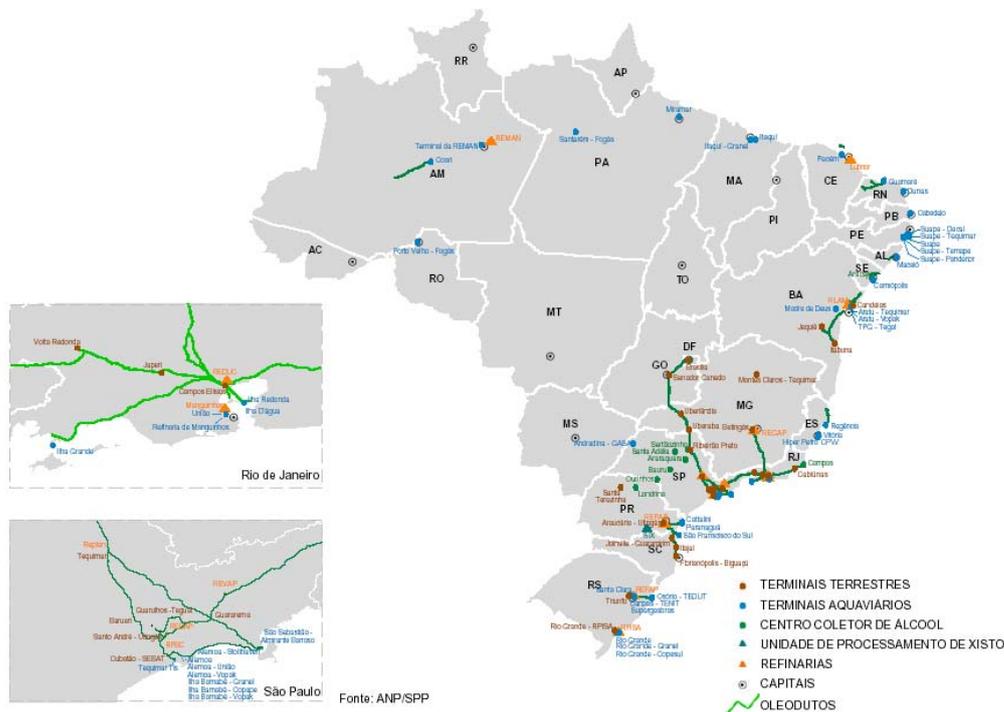


Figura 1 – Malha dutoviária brasileira (Fonte:ANP, 2006)

De acordo com Tavares (2005), em 2002, o Brasil possuía uma infra-estrutura dutoviária composta por 411 dutos destinados à movimentação de petróleo, derivados, gás natural e outros produtos. Destes, 10,2 mil km de dutos eram para transporte, isto é, destinados à movimentação de petróleo e derivados ou gás natural por percursos considerados de interesse geral

e 4,9 mil km de dutos eram destinados à transferência, destinados à movimentação de petróleo, seus derivados ou gás natural por percursos considerados de interesse específico e exclusivo do proprietário ou explorador das facilidades. Esses dutos somam um total de 15,1 mil km de extensão.

Segundo Tavares (2005), a capacidade total dos dutos considerados como de transporte era de $334.848 \text{ m}^3/\text{dia}$, sendo que apenas 55 % deste volume têm sido usados em média nos últimos seis anos. Em 2008, a Transpetro (TRANSPETRO, 2009) movimentou 670 milhões de m^3/ano , o que mostra um aumento considerável de movimentações no último ano. Assim, faz-se necessário estudos que possibilitem uma melhor utilização da malha dutoviária.

O planejamento (*planning*) e a programação (*scheduling*) das operações de uma rede dutoviária devem ter por objetivo a utilização eficiente dos recursos, resultando em um melhor desempenho operacional e em maior segurança para os seres humanos e para o meio ambiente.

O *scheduling* é necessário sempre que exista uma competição entre atividades para o uso de recursos limitados, dentro de um determinado período de tempo (REKLAITIS, 1992), sendo assim, uma estratégia de operação a ser otimizada, a qual envolve três elementos-chave, que são: a atribuição dos recursos, o seqüenciamento de atividades, e a determinação do tempo do uso dos recursos pelas atividades. A componente de atribuição envolve a seleção do conjunto de recursos para a realização de uma dada atividade. A componente de seqüenciamento se constitui do ordenamento das atividades atribuídas aos recursos. E a componente de temporização envolve a determinação dos tempos específicos de início e fim de cada uma das atividades (REKLAITIS, 1992). Essas três componentes permitem uma descrição matemática completa do problema e, assim, uma possível obtenção de uma solução matemática e/ou computacional.

Este trabalho fez parte de um grupo de estudos e pesquisa, que busca desenvolver modelos de otimização que possibilitem a realização, de forma sistêmica, do *scheduling* das operações da malha dutoviária brasileira. Este grupo de pesquisa desenvolveu uma ferramenta computacional capaz de obter soluções reais otimizadas de *scheduling* da malha dutoviária, através de um sistema híbrido, que combina técnicas heurísticas e metaheurísticas a métodos baseados em programação linear inteira mista.

1.1 MOTIVAÇÃO

O problema de *scheduling* de uma rede dutoviária é um problema de otimização combinatória de difícil solução, que envolve complexas restrições operacionais, tornando necessário o desenvolvimento de sistemas computacionais que auxiliem na tomada de decisão. Atualmente no Brasil, a programação das operações é realizada por especialistas experientes. Entretanto, técnicas de pesquisa operacional têm se tornado cada vez mais uma ferramenta eficiente na otimização destes processos operacionais, auxiliando na modelagem e resolução de problemas

combinatórios.

Uma modelagem do problema de *scheduling* da rede de dutos deve satisfazer às restrições operacionais, restrições de distribuição e demanda de produtos, restrições de seqüenciamento e restrições operacionais dos dutos e áreas interligadas. É fundamental uma solução que permita não somente encontrar o ponto ótimo operacional de transferência de produtos, mas também que busque as melhores condições de estocagem em todos os pontos conectados ao duto. De acordo com Rejowski e Pinto (2003), as restrições de seqüenciamento são um grande complicador na geração de uma solução operacional ótima. O resultado da temporização é fortemente impactado pela qualidade da seqüência gerada *a priori*.

Além da otimização do sistema dutoviário ser de grande importância, pela capacidade do transporte de grandes quantidades de produtos de uma forma rápida e segura, ele diminui a sobrecarga do sistema rodoviário nacional (REJOWSKI; PINTO, 2003), a qual apresenta graves problemas de infra-estrutura.

Na literatura, os trabalhos que tratam do problema de *scheduling* operacional de uma malha dutoviária real são, em sua maioria, baseados em programação matemática. A aplicação de programação matemática em um problema real implica na adoção de hipóteses que simplificam demasiadamente a realidade ou que restringem muito o problema, o que torna muito dispendiosa a solução do ponto de vista computacional (ALLE, 2003). Soma-se a isto o fato dos modelos matemáticos serem de difícil entendimento por outros usuários que não os seus idealizadores. Uma alternativa a essas características é o uso de metaheurísticas.

O benefício do uso de técnicas metaheurísticas é a redução de restrições das rotinas de otimização na formulação do modelo, o que simplifica a representação do modelo e possibilita uma facilidade de implementação de possíveis mudanças no modelo. Assim, esta tese explora o uso de técnicas metaheurísticas as quais, nas últimas décadas, têm sido amplamente estudadas e utilizadas na resolução de variados problemas combinatoriais complexos e mostram ser muito eficientes na busca de soluções viáveis em um tempo mais curto.

O seqüenciamento das bateladas é um problema que pode ser resolvido através de metaheurísticas; entretanto é um problema que engloba uma série de critérios que otimizados podem implicar em uma melhora significativa da solução do *scheduling*. Para estes problemas, tem-se como alternativa o uso de algoritmos baseados em busca estocástica, como por exemplo os métodos evolucionários multiobjetivos.

1.2 OBJETIVO

Preliminarmente, este estudo visa a solução do *scheduling* de uma rede de dutos através de um sistema híbrido baseado numa combinação de técnica metaheurística e programação matemática.

O objetivo principal é, ao longo deste trabalho, realizar a análise, avaliação e a aplicação de técnicas metaheurísticas e de otimização multiobjetivo no seqüenciamento do problema de *scheduling* de uma malha dutoviária. Como consequência deste estudo propõe-se, ao final do trabalho, um modelo baseado em algoritmo genético multiobjetivo para o auxílio à tomada de decisão do problema de programação das operações desta malha.

Como citado anteriormente, o problema de *scheduling* das operações de distribuição de produtos derivados do petróleo é um problema muito complexo, que atualmente é realizado por um grupo de especialistas da Petrobrás. Assim, deseja-se desenvolver um modelo computacional que forneça soluções operacionais atendendo os requisitos do problema de forma otimizada.

Os resultados alcançados por este trabalho contribuem na busca de soluções otimizadas para o problema de *scheduling* das atividades operacionais do sistema dutoviário brasileiro de distribuição de produtos derivados do petróleo, fornecendo soluções alternativas que representem melhoras com relação: ao tempo total do *scheduling*, às taxas de utilização dos dutos, à diminuição do número de transições entre diferentes produtos e à entrega de produtos satisfazendo sua oferta e demanda, permitindo ao especialista a análise do comportamento de diferentes soluções na malha dutoviária e a escolha da solução mais adequada ao resultado esperado.

1.3 CONTRIBUIÇÕES

Dentre as contribuições deste trabalho tem-se:

- A caracterização do problema de *scheduling* operacional de uma malha dutoviária.
- A decomposição do problema de *scheduling* de uma rede de dutos em três subproblemas. Cada subproblema pode ser resolvido por qualquer técnica de otimização, de forma independente.
- O estudo de diversas metaheurísticas aplicadas ao problema de sequenciamento.
- O desenvolvimento de um sistema híbrido, composto por heurística, metaheurística e programação linear, aplicado a um problema real.
- O desenvolvimento da formulação do modelo para o problema do seqüenciamento de bateladas do problema de *scheduling* através das metaheurísticas e do MOGA, além da comparação dos resultados alcançados.

Tudo isso mostra que este trabalho contribui com o desenvolvimento de uma ferramenta de auxílio ao especialista, e torna a solução do problema real de *scheduling* possível de se efetuar de forma sistêmica.

1.4 ORGANIZAÇÃO DO TRABALHO

Este trabalho é dividido em seis capítulos. O segundo capítulo traz uma descrição do problema de *scheduling*, apresenta os problemas de *planning* e *scheduling* e comenta suas diferenças, além de citar alguns trabalhos desenvolvidos utilizando técnicas metaheurísticas e sistemas híbridos na resolução de problemas de *scheduling*.

No terceiro capítulo tem-se a descrição de sistemas híbridos, e são mostradas as vantagens de sua utilização na solução de problemas de grande complexidade. Citam-se razões para o uso de técnicas metaheurísticas em problemas como o de sequenciamento de bateladas. São descritas as técnicas metaheurísticas que posteriormente foram implementadas. Os problemas multiobjetivo são caracterizados e técnicas de otimização multiobjetivo são apresentadas.

O quarto capítulo apresenta as características de uma rede real de dutos na qual serão validadas as heurísticas para o *scheduling* de operações de transporte de derivados do petróleo. Expõe a abordagem, adotada neste trabalho, de divisão deste problema através de sistemas híbridos.

No quinto capítulo tem-se o estudo da solução do problema de sequenciamento de bateladas através de diferentes técnicas metaheurísticas. Os resultados obtidos e as comparações entre as diferentes técnicas metaheurísticas são apresentados, mostrando a técnica que obteve melhores soluções em tempos curtos de processamento. Descreve-se em detalhes os resultados obtidos pela aplicação do modelo baseado em algoritmo genético, usando diferentes funções objetivo na otimização do problema de sequenciamento, indicando a função que produz uma melhor solução para o problema de *scheduling*. Posteriormente, este modelo baseado em algoritmo genético é estendido para tratar o problema multiobjetivo, e são apresentadas uma comparação e uma análise dos resultados obtidos.

A conclusão é apresentada no sexto capítulo, ressaltando as contribuições desta tese, juntamente com a proposição de estudos futuros.

2 LEVANTAMENTO BIBLIOGRÁFICO

Este capítulo descreve o problema de *scheduling* e sua representação segundo estratégias de operação. Mostra a diferença entre *planning* e o *scheduling*. Apresenta alguns trabalhos relacionados à solução do *scheduling* com a aplicação de diferentes técnicas, como metaheurísticas e sistemas híbridos.

2.1 SCHEDULING

Segundo Reklaitis (1992), em processos químicos, o *planning* e o *scheduling* de produção referem-se aos procedimentos de alocação, em um determinado período de tempo, de recursos e equipamentos para executar o processamento de tarefas. Usualmente, o *planning* produz decisões sobre escalas longas de tempo, medidas em meses, trimestres ou anos, enquanto que o *scheduling* foca decisões em escalas curtas de tempo, como dias ou semanas.

De acordo com Morton e Pentico (1993), os níveis do *scheduling* seguem a classificação da tabela 1.

Tabela 1 – Diferenças entre *scheduling* e *planning* (Morton e Pentico, 1993)

Nível	Exemplo de Problemas	Horizonte
Planejamento de longo prazo	Expansão de plantas, layout de plantas e projeto de plantas	2-5 anos
Planejamento de médio prazo	Planejamento da produção, logística	1-2 anos
Planejamento de curto prazo	Planejamento das necessidades de material, ordens de compra, estabelecidos por prazo de entrega	3-6 meses
<i>Scheduling</i>	Roteamento <i>Jobshop</i> , balanceamento de linhas de montagem, dimensionamento de bateladas	2-6 semanas
<i>Scheduling</i> Reativo	Tarefas urgentes, quebra de máquinas, materiais atrasados	1-3 dias

Segundo Reklaitis (1992), o problema de *scheduling* consiste em uma estratégia de operação, constituída por um conjunto de equipamentos, um conjunto de recursos como operadores, utilidades e materiais, um conjunto de ofertas e demandas de produtos, especificações dos recursos disponíveis e requeridos, e um ou mais objetivos a serem otimizados.

Lee e Dagli (1997) apresentaram algumas das dificuldades que estimulam o desenvolvimento de uma solução eficiente para o problema de *scheduling*:

- o *scheduling* é um problema complexo da classe NP-hard;
- é um problema restrito, em função dos recursos utilizados;
- as decisões possuem uma dependência com relação a outras decisões, ou seja, as decisões não são isoladas;
- o *scheduling*, normalmente, tende a ser um problema multiobjetivo de otimização.

Em um problema de *scheduling*, inicialmente, faz-se necessário o estudo da topologia da planta, que relaciona a disposição das unidades ou equipamentos interconectados numa planta de processamento, podendo ter estágios em série ou dispostos em uma rede.

O *scheduling* pode ser tratado através do envio de lotes ou bateladas de produtos, de vários tamanhos, assim como pode ser realizado o balanço de massa, em que é verificada a variação de massa.

A atribuição do tempo para as operações programadas pode ser realizada através da forma discreta ou da forma contínua do tempo na formulação do problema. A forma de representação do tempo possui grande influência sobre o número de variáveis usadas na modelagem do problema. A opção por intervalos fixos, ou discreta, leva a um maior número de variáveis. Com representações contínuas de tempo, o número de variáveis pode ser significativamente menor, ainda que a formulação do problema se torne mais complexa (ALLE, 2003).

Outro item importante do *scheduling* é a existência de estoques, a disponibilidade de espaço para armazenamento e a limitação ou não de sua capacidade.

Os recursos humanos, mão-de-obra e as utilidades, como a energia elétrica, se limitados, geram dificuldades na resolução do problema, de forma a impor restrições que não podem ser violadas.

Os padrões de demanda podem ser variáveis ou fixos dentro de um horizonte de tempo, caracterizando um problema de curto prazo (*short-term scheduling*) ou cíclico (*cyclic scheduling*), respectivamente.

Além desses ítems, Alle (2003) cita a classificação do problema em estocástico e determinístico, dependendo da inclusão ou não de incerteza nos parâmetros como datas de entrega, taxas de produção, preços, etc.

De acordo com Reklaitis (1992), devido à necessidade da resolução do *scheduling* nas áreas comercial e de manufatura, muitas metodologias foram amplamente investigadas nas mais variadas áreas de estudo como engenharias, ciência da computação, ciência gerencial e negócios.

Na indústria de processos contínuos, como é o caso da indústria química, as estratégias operacionais de produção podem ser representadas em um espaço bidimensional caracterizado pelo grau de similaridade da trajetória de processamento dos produtos e pelo tamanho relativo da campanha. A trajetória de processamento dos produtos representa a ordem dos estágios de processamento pela qual os produtos devem passar em sua produção e o tamanho da campanha se refere à quantidade de produto a ser processada (REKLAITIS, 1992).

A estratégia operacional para indústrias com baixa similaridade da trajetória dos processamentos e campanhas pequenas (uma ou duas bateladas) requer uma forma de organização diferente das indústrias que possuem grande similaridade entre seus estágios de processamento e produzem através de grandes campanhas. Assim, como pode ser visto na figura 2, Reklaitis (1992) descreve o *flowshop* como um problema de pequena campanha e grande similaridade entre os estágios de processo, e o *jobshop* como um problema de pequena campanha e baixa similaridade da trajetória de processo, assim como caracteriza os problemas de longa campanha como multiprodutos (*multiproduct*), multiplantas (*multiplant*) e multipropósito (*multipurpose*), para problemas com alta, média e baixa similaridade dos estágios de processo, respectivamente.

Grau de Similaridade dos Processos

		Alto	Médio	Baixo
Tamanho da Campanha	Pequeno	<i>Flowshop</i>		<i>Jobshop</i>
	Médio			
	Grande	Multiprodutos	Multiplantas	Multipropósito

Figura 2 – Espaço de estratégias de operação (Fonte: Reklaitis, 1992)

Yamada (2003) descreve o *jobshop scheduling* como sendo um problema de designação de n serviços em m máquinas, em que a seqüência das máquinas para cada serviço e a

seqüência dos serviços definidos para cada máquina podem ser diferentes, portanto não sendo possível verificar uma trajetória de estágios de processo. O *flowshop scheduling* é considerado como um caso especial de *jobshop* em que são iguais a seqüência das máquinas para todos os serviços e a ordem em que os serviços são realizados nas máquinas, apresentando um único roteiro de processamento, sendo a solução representada pela permutação dos n serviços, ao invés de m permutações de n serviços.

Os problemas de *jobshop* e *flowshop* são largamente trabalhados na literatura.

Quando as indústrias possuem grandes campanhas, independente de serem multiprodutos, multiplantas e/ou multipropósito, o *scheduling* torna-se um problema combinatório de difícil solução. Este é o caso da rede de dutos tratada nesta tese que pode ser caracterizada como um problema de longa campanha e com uma trajetória complexa.

Existem várias técnicas destinadas a resolver problemas de otimização combinatória, podendo ser destacadas:

- Programação matemática (inteira mista) linear ou não linear;
- Programação dinâmica;
- Programação lógica por restrições;
- Métodos heurísticos ou Metaheurísticos;
- Métodos híbridos, que combinam dois ou mais dos métodos citados acima.

A programação matemática e a metaheurística estão entre as mais aplicadas na solução de problemas de *scheduling*. Os adeptos da programação matemática visam alcançar uma solução ótima, e os da metaheurística procuram soluções que satisfaçam os critérios de viabilidade, relativa à qualidade e um tempo computacional aceitável (ALLE, 2003).

Segundo Morton e Pentico (1993), os métodos exatos de solução do *scheduling* incluem a programação inteira (*branch-and-bound*) e a programação dinâmica. Estes podem modelar problemas complexos de otimização, entretanto não são práticos na solução de problemas amplos, que apresentam um número grande de variáveis de decisão e restrições, o que favorece o uso de métodos metaheurísticos.

Nos trabalhos de Magalhães (2004), Stebel (2006), Rejowski e Pinto (2003) tem-se uma revisão de trabalhos que usam técnicas de programação matemática em otimização linear e não-linear na solução de problemas de *scheduling*. O estudo e aplicação de *scheduling* e controle da produção na indústria petroquímica ficam limitados a problemas cuja dimensão e complexidade muitas vezes não correspondem àquelas de problemas reais. Uma alternativa é

o uso de técnicas metaheurísticas, que permitam tornar o processo de busca mais rápido, em detrimento a obtenção da solução ótima.

A seguir, na revisão bibliográfica, tem-se algumas aplicações de metaheurísticas em *scheduling*.

2.1.1 Revisão Bibliográfica

Ku e Karimi (1991) aplicaram *simulated annealing* (SA) no problema de *scheduling* de processos operacionais. Concluíram que o SA é uma técnica promissora para a solução de uma variedade de problemas de *scheduling*, além de ser um algoritmo simples e as soluções apresentam-se próximas à ótima, embora exija um amplo esforço computacional.

Tandon et al. (1995) usaram o *simulated annealing* (SA) na solução do problema de *scheduling* de processos. Este problema visa determinar a ordem em que cada produto será processado pelas várias unidades de processos, minimizando o tempo. Através dos resultados alcançados e de comparações realizadas com outro método heurístico, o HIM (*Heuristic Improvement Method*) apresentado por Musier e Evans em 1989 (appud Tandon et al., 1995), observou-se que o SA apresenta melhores resultados em problemas maiores, com maior número de produtos, embora a qualidade da solução se deteriore com o aumento da complexidade do problema.

Mockus e Reklaitis (1996) apresentaram uma solução para o problema de *scheduling* de um grupo de processos usando aproximação heurística aleatória (*randomized heuristics approach*), através de uma representação contínua do tempo baseada em uma discretização não-uniforme do tempo. A função heurística é obtida através de escolha aleatória do conjunto de decisões, e não é necessária a atribuição de regras para a tomada de decisão. A escolha é realizada a partir de uma probabilidade. Desta forma, tem-se uma função heurística parametrizada dos pesos. Então, o problema original é substituído por um problema auxiliar de otimização contínua estocástica que pode ser solucionada pelo método Bayesiano de otimização global (MOCKUS; REKLAITIS, 1996).

Sasikumar et al. (1997) desenvolveram um método heurístico para o problema de *scheduling* de dutos para a distribuição de produtos derivados do petróleo. A solução consiste de quatro componentes: o produto a ser enviado, sua quantidade, como o produto é distribuído até seu destino e a seqüência de envio, levando em consideração restrições do tamanho da batelada, do seqüenciamento de envio, e de tancagem na origem e no destino. Com a solução desenvolvida é possível obter as datas de envio e chegada do produto nas diferentes localidades. As soluções alcançadas foram consideradas aceitáveis pelos usuários para o problema de *scheduling* mensal.

Montana et al. (1998) desenvolveram uma aproximação genérica para problemas de

scheduling usando algoritmo genético que possibilita uma relativa facilidade e agilidade na resolução de problemas com diferentes domínios. Neste trabalho são exemplificados dois problemas de *scheduling*: de programação do serviço em campo e de programação da movimentação militar por terra de um forte a um porto.

Löhl et al. (1998) aplicam o algoritmo genético (AG) no problema de seqüenciamento de um conjunto de operações de um processo de produção. Foram feitas comparações entre o algoritmo genético e o algoritmo de programação matemática (MINLP - *Mixed Integer Non-linear Program*), e foi observado que a programação matemática obtém melhores resultados que o AG. Entretanto, se o objetivo for obter uma solução viável com um menor tempo de processamento indica-se adotar a solução por AG.

Crane et al. (1999) apresentaram um modelo aplicando algoritmo genético para a solução do problema simplificado de *scheduling* de dutos. A rede tratada é composta por 8 nós terminais com 2 tanques em cada, ligados por 7 dutos unidirecionais, não permitindo a reversão do sentido do fluxo, 2 produtos e 3 níveis de capacidade de armazenamento dos produtos nos tanques. Foi obtida a solução ótima para este problema, mostrando, assim, a possibilidade de solução de um problema complexo como o *scheduling* de dutos através da aplicação de algoritmo genético.

Castro (2001a) utilizou um software comercial cujo otimizador baseia-se em algoritmo genético e desenvolveu um modelo para o sistema de produção e armazenamento de gás liquefeito de petróleo da Refinaria Henrique Lage da Petrobrás.

Cowling et al. (2002) apresentaram o uso de hiper-heurística baseada em algoritmo genético aplicando em um problema de *scheduling*. Pôde-se verificar que essa hiper-heurística, Hyper-GA, é uma técnica genérica e promissora para problemas de otimização. Utiliza o algoritmo genético como um seletor de alto nível, e cada indivíduo da população é representado por uma seqüência de valores inteiros e cada valor representa uma heurística de baixo nível. As heurísticas de baixo nível têm o intuito de adicionar os eventos ainda não programados, obedecendo prioridades ou de forma aleatória observando possíveis conflitos entre eles. Para o problema considerado o Hiper-GA obteve ótimo desempenho, segundo Cowling et al. (2002).

Han et al. (2002) apresentaram um modelo desenvolvido a partir da hiper-heurística Hyper-GA, em que cada indivíduo ou cromossomo possui tamanho variável. O tamanho do cromossomo se adapta através da identificação do desempenho de uma heurística ou de uma combinação de heurísticas, selecionada pelo cromossomo. Este modelo foi aplicado no problema de *scheduling* apresentado por Cowling et al. (2002), e obteve melhor desempenho que o Hyper-GA.

Cruz et al. (2003) apresentaram um modelo com otimização multiobjetivo e um algoritmo evolucionário com restrições, para o problema de distribuição de produtos derivados do

petróleo por uma malha dutoviária. O modelo foi aplicado em um problema simplificado, com o objetivo principal de satisfazer a demanda de produtos em um tempo mínimo, minimizando a interface entre diferentes produtos.

Garcia et al. (2004) usaram duas técnicas para a solução do problema de *scheduling*: um método heurístico baseado em algoritmo evolucionário multiobjetivo, e um método de programação matemática baseado em restrições e programação linear inteira mista. Seus resultados mostraram que os métodos evolucionários facilmente solucionam problemas de otimização combinatória não-lineares. São capazes de evitar ótimos locais, e ambas as técnicas alcançaram soluções similares. Entretanto o uso do método híbrido mostrou melhores soluções em um menor tempo de execução.

Westphal (2006) usou o algoritmo genético multiobjetivo em um problema de *scheduling* em uma rede simplificada de distribuição de derivados de petróleo. O AG desenvolvido apresentava uma estratégia de elitismo baseada em nichos e usava uma aproximação do método da ponderação de objetivos (COELLO, 1999b) para gerar prioridades entre os critérios de otimização. Seus resultados alcançaram boas soluções em curto espaço de tempo.

2.1.2 Métodos Híbridos

Uma outra alternativa para tornar possível a solução de problemas de grande complexidade, através de um tempo computacional menor, é o uso de sistemas híbridos, utilizando técnicas metaheurísticas combinadas entre si ou com métodos de programação matemática.

Em particular a programação por restrições e a programação linear inteira mista devem ser mencionadas como dois campos de pesquisa com longas histórias e também muitos sucessos na solução de problemas difíceis de otimização combinatória. Assim como a metaheurística, os métodos dessas áreas também apresentam suas vantagens específicas e limitações. Especialmente nos últimos anos, combinações entre essas técnicas e metaheurísticas têm se mostrado promissoras e geralmente se apresentado bem sucedidas. Essas combinações são conhecidas na literatura como métodos híbridos. Puchinger e Raidl (2005) apresentaram um levantamento bibliográfico de trabalhos sobre sistemas híbridos, que combinam métodos exatos e metaheurísticas.

Buzzo e Moccellini (2000) usaram métodos híbridos com o algoritmo genético e o *simulated annealing* no problema de *flowshop scheduling* permutacional com o objetivo de minimizar o *makespan*. Realizaram uma comparação dos métodos AG e SA puros e métodos híbridos. Observaram que o método híbrido pode ser desenvolvido de forma a agregar as vantagens dos dois métodos puros.

Yamada (2003) trabalhou com metaheurísticas aplicadas no problema de *jobshop scheduling* e *flowshop scheduling*, onde usou o algoritmo genético (AG), *simulated annealing* e

busca tabu, incorporando outras técnicas em seus desenvolvimentos como a inclusão do algoritmo GT¹, proposto por Giffler e Thompson no final dos anos 50 (appud Yamada, 2003), em um passo do algoritmo genético, a utilização da heurística *shifting bottleneck* (SB) combinada ao *simulated annealing*, assim como o uso da lista "*pruning pattern*" na redução do espaço de busca para o método da busca tabu.

Garcia et al. (2004) aplicaram uma metodologia heurística e programação matemática para a solução do problema de distribuição de derivados de petróleo em uma rede simplificada de dutos.

Relvas et al. (2009) usaram um sistema híbrido, que combina um modelo PLIM e uma heurística, para um problema de *planning* e *scheduling* de uma cadeia de suprimentos, que engloba o gerenciamento de um duto para multiprodutos e um inventário de produtos.

Uma extensa revisão da literatura sobre algoritmos híbridos pode ser encontrada em Talbi (2002).

Na tabela 2 a seguir tem-se um levantamento bibliográfico específico sobre a aplicação em dutos obtida do trabalho de SouzaFilho et al. (2007). Pode-se observar a complexidade da obtenção de uma solução viável e operacional de um problema de *scheduling* real que englobe uma rede com vários dutos e que seja composta por várias fontes e destinos de distribuição.

No próximo capítulo, são apresentadas as técnicas metaheurísticas, utilizadas nesta tese.

¹O algoritmo GT (Giffler e Thompson) seleciona as operações a serem programadas de forma sequencial considerando a prioridade das operações de cada máquina, e os seus tempos iniciais e finais, resultando em um *active schedule*, ou seja, uma programação em que não há mais possibilidade das operações serem antecipadas.

Tabela 2 – Tabela resumo da literatura sobre aplicação em dutos (Fonte: Souza Filho et al., 2007)

Autor	Dutos	Destinos	Fontes	Técnica	Horizonte de Tempo e Solução Alcançada
Shah (1996)*	Único	Único	Única	MILP (com discretização uniforme do horizonte de tempo)	1 mês Solução viável
Magatão et al. (2004)*	Único	Único	Única	MILP (com discretização uniforme do horizonte de tempo)	120 horas Solução ótima
Hane e Ratliff (1995)*	Único	Vários	Única	Algoritmo Guloso + Branch-and- Bound	– Solução ótima
Sasikumar et al. (1997)	Único	Vários	Única	Beam Search (Inteligência Artificial)	1 mês Solução viável
Rejowski (2001* e 2003)	Único	Vários	Única	MILP (com discretização uniforme do horizonte de tempo)	75 horas Solução viável
Rejowski e Pinto (2004)*	Único	Vários	Única	MILP (com discretização uniforme do horizonte de tempo)	75 horas Solução ótima
Cafaro e Cerdá (2004)*	Único	Vários	Única	MILP (com representação contínua de tempo)	75 horas Solução ótima
Jittamai (2004)*	Único	Vários	Única	Heurística de Fluxo Reverso	– Solução viável
Sanginetto (2006)*	Único	Vários	Única	Algoritmo Genético	168 horas Solução viável
Souza Filho (2007)*	Único	Vários	Única	VNS	168 e 336 h Solução viável
Alves (2007)*	Rede	Vários	Vários	Algoritmo Genético	7 e 14 dias Solução Viável
Camponogara (1995)*	Rede	Vários	Vários	Heurística A-Team	120 h Solução Inviável
Crane et al. (1999)	Rede	Vários	Única	Algoritmo Genético	– Solução Viável
Milidiú et al. (2001)*	Rede	Vários	Vários	GRASP	120 h Solução viável
Braconi (2002)*	Rede	Vários	Vários	PL (com representação contínua de tempo) + Heurística	5 a 30 dias Solução viável
De la Cruz et al. (2003 e 2005)*	Rede	Vários	Vários	Algoritmo Evolucionário Multiobjetivo e MILP	65 períodos de tempo Solução viável
Pessoa (2003)*	Rede	Vários	Vários	Heurística	Solução viável/ótima (resultados teóricos)
Liporace (2005)*	Rede	Vários	Vários	Heurísticas (Inteligência Artificial)+ Simuladores	– Solução viável
Más e Pinto (2003)*	Rede	Vários	Única	MILP de tempo contínuo baseado em eventos	168 horas Solução viável
Neiro e Pinto (2004)*	Rede	Vários	Vários	MINLP (com discretização do horizonte de tempo)	2 períodos de tempo Solução viáveis

*(appud, Souza Filho et al., 2007)

3 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta algumas vantagens do uso de metaheurísticas e descreve as técnicas metaheurísticas que serão implementadas na solução do problema de seqüenciamento de bateladas, mostrando seus algoritmos simplificados, e apresenta técnicas de otimização multi-objetivo e sistemas híbridos.

3.1 SISTEMAS HÍBRIDOS

Os problemas de otimização combinatória de difícil solução aparecem em grande número de aplicações de problemas reais, como roteamento, atribuição, *scheduling*, *cutting and packing*, projetos de redes, alinhamento de proteínas, e em muitos campos diferentes como econômicos, industriais e de importância científica. As técnicas aplicadas a esses problemas podem ser classificadas em duas principais categorias: métodos exatos e heurísticos. Os algoritmos exatos garantem obter a solução ótima e provar sua otimalidade em qualquer instância do problema. Entretanto, o tempo de processamento aumenta drasticamente com o aumento do tamanho do problema, e apenas pequenos problemas ou problemas de tamanho moderado podem ser resolvidos dessa forma. Neste caso, a única possibilidade de solução de grandes problemas é através de algoritmos heurísticos, sacrificando a garantia da obtenção de soluções ótimas em vista de obter-se boas soluções em um tempo limitado (PUCHINGER; RAIDL, 2005).

Em um sistema híbrido, inicialmente, distingue-se o que será hibridizado, quais algoritmos serão utilizados. Isso pode ocorrer através da combinação de: (a) diferentes estratégias metaheurísticas, (b) metaheurísticas com algoritmos específicos para o problema considerado, ou (c) metaheurísticas juntamente com outras técnicas como de pesquisa operacional (PO) e inteligência artificial (IA) (RAIDL, 2006).

Assim, deve-se, também, distinguir o nível de combinação dos algoritmos:

- Combinação de alto nível - Os algoritmos permanecem com sua identidade original, com uma interface bem definida, entre eles não apresentando nenhuma relação interna entre algoritmos.
- Combinação de baixo nível - Os algoritmos dependem fortemente um do outro e há uma troca de funções ou componentes entre eles.

Outra propriedade do sistema híbrido que deve ser apresentada é a ordem de execução. No modelo "em lotes" ou sequencial um algoritmo é executado após o outro, e a informação é passada de um algoritmo ao outro em uma única direção. Nesse modelo, um algoritmo pode funcionar como pré-processamento de dados para o outro algoritmo ou ainda como um pós-processamento dos resultados do primeiro algoritmo. Outro exemplo dessa estratégia de sequenciamento pode ser encontrada em problemas multinível. Esse tipo de problema é resolvido considerando-se um nível após o outro com algoritmos de otimização dedicados a cada nível do problema. Ao contrário do modelo "em lotes", existem modelos paralelos e intercalados, em que os algoritmos devem interagir de uma forma mais sofisticada. Atualmente, o estudo sobre metaheurísticas paralelas é um campo de pesquisa importante e amplo (RAIDL, 2006). Pode-se ainda distinguir as metaheurísticas híbridas de acordo com suas estratégias de controle. De acordo com Puchinger e Raidl (2005), existem combinações integrativas (coercivas) e colaborativas (cooperativas) que por sua vez podem ser executadas de forma sequencial ou paralela, como mostra a figura 3.

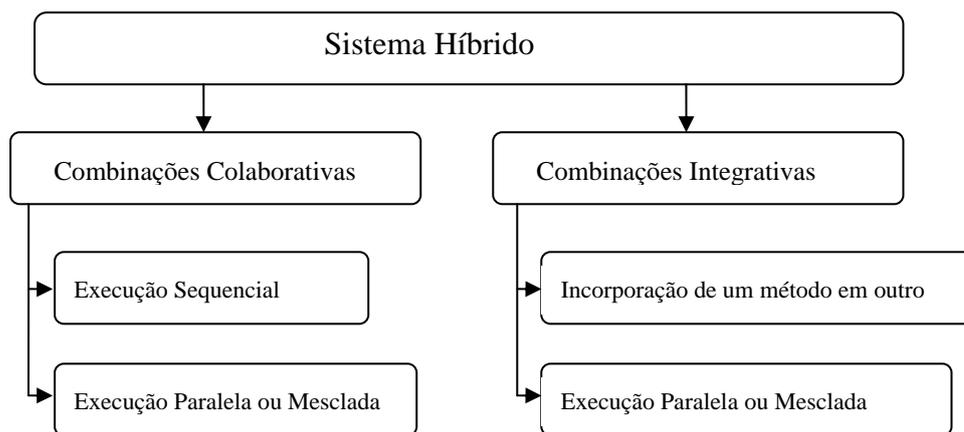


Figura 3 – Classificação do sistema híbrido segundo a estratégia de controle (Fonte: Puchinger e Raidl, 2005)

Em combinações integrativas, uma técnica é subordinada a outra, funcionando como um componente de outra técnica.

Em combinações colaborativas, a questão é qual o espaço de busca que deverá ser explorado por cada algoritmo. Por colaboração, os algoritmos trocam informações entre eles, mas um não faz parte de outro. Os algoritmos podem ser executados sequencialmente, intercalados ou em paralelo. De acordo com Raidl (2006) pode-se distinguir a decomposição do problema entre a forma implícita, que resulta de diferentes soluções iniciais, diferentes valores de parâmetros, etc., e a explícita, em que cada algoritmo trabalha explicitamente em um subespaço definido. A decomposição de problemas amplos, na prática, é geralmente de crucial importân-

cia, para possibilitar o solucionamento do problema dentro de um determinado limite de tempo de processamento. Ocasionalmente, problemas podem ser decompostos de forma natural, mas na maioria dos casos é difícil encontrar a decomposição ideal em partes independentes.

Existe diversas possibilidades de hibridização entre metaheurísticas e/ou com algoritmos de outras áreas de estudo. Um grande número de publicações mostra um grande sucesso e benefícios da hibridização. Tem-se técnicas promissoras de combinação de metaheurísticas com programação por restrições e técnicas de programação inteira. O trabalho de Puchinger e Raidl (2005) mostra a utilidade e o forte potencial da pesquisa de combinações entre algoritmos exatos e metaheurísticas.

Recentemente tem-se obtido diferentes esforços para a combinação de idéias e métodos dessas duas linhas científicas. Ambas as técnicas de otimização, exata e metaheurística, apresentam vantagens específicas que se complementam. A combinação dessas técnicas pode beneficiar pela sinergia, resultando em um melhor desempenho com respeito à qualidade da solução e o tempo de processamento. Assim, futuros estudos e aplicações de sistemas híbridos são altamente promissores (PUCHINGER; RAIDL, 2005).

Neste contexto se enquadra esta tese, que pretende combinar técnicas heurísticas e modelos de programação linear inteira mista na solução de problemas de *scheduling* operacional de uma rede de dutos.

3.2 METAHEURÍSTICAS

Uma heurística é um conjunto de regras usadas na geração de uma solução para um problema, baseada na experiência, entretanto não garante a geração de uma solução ótima. O termo metaheurística pode ser definido como um processo de geração iterativa de soluções que guia uma heurística pela combinação inteligente de diferentes conceitos de exploração do espaço de busca, em que estratégias de aprendizado são usadas para encontrar soluções ótimas, ou próximas, de forma eficiente (BLUM; ROLI, 2003).

Segundo Gendreau e Potvin (2005), o termo metaheurística foi inventado por Glover em 1986, para denominar heurísticas especializadas na solução de problemas complexos de otimização combinatorial, os quais geralmente envolvem um grande número de variáveis de decisão. Estes incluem os problemas de agrupamento, ordenação ou seleção de objetos discretos.

Falcone (2004) cita que a metaheurística é um guia para as heurísticas e tem por objetivo direcionar suas buscas, permitindo a exploração de diferentes regiões do espaço de busca e tentando obter diferentes ótimos locais, através de uma busca eficiente.

Silver (2002) apresenta algumas razões para a utilização de métodos heurísticos:

- Facilidade de implementação - A aceitação e uso de regras pelos tomadores de decisão pode ser facilitada pelo entendimento de como as regras operam, em particular como os parâmetros influenciam as ações.
- Melhorias sobre a prática tradicional - os gerentes podem ficar satisfeitos com uma solução heurística que pode produzir melhores resultados que os obtidos por métodos tradicionais.
- Resultados mais rápidos - muitas vezes faz-se necessário uma solução rápida. Técnicas heurísticas podem ser executadas mais rapidamente que as técnicas exatas de otimização.
- Robustez - A heurística pode ser menos sensível às variações das características do problema e da qualidade dos dados.
- Possibilidade de uso da heurística dentro de rotinas de otimização.

Blum e Roli (2003) resumiram as propriedades fundamentais que caracterizam as metaheurísticas:

- As metaheurísticas são estratégias que guiam o processo de busca;
- A finalidade é de explorar de forma eficiente o espaço de busca ao invés de procurar a solução ótima;
- As técnicas que constituem as metaheurísticas podem variar desde um simples procedimento de busca local até um processo de aprendizagem complexo;
- As metaheurísticas são técnicas aproximadas e geralmente não-determinísticas;
- Podem ser incorporados mecanismos que evitem que a busca permaneça em uma única região do espaço de busca;
- A metaheurística não é uma técnica desenvolvida para um problema em específico;
- O conceito básico da metaheurística permite uma descrição abstrata de seu algoritmo, que pode ser utilizado em muitos problemas;
- A metaheurística pode utilizar o conhecimento do problema como uma estratégia de alto nível para controlar uma heurística.

Existem várias formas de se classificar as metaheurísticas, que dependem da forma em que se quer diferenciar as características das técnicas. As classificações mais importantes são:

- Técnica inspirada ou não na natureza - é uma forma intuitiva de classificar; exemplificando, tem-se o algoritmo genético que é inspirado na natureza e a busca tabu que não é inspirada na natureza.
- Busca local ou busca populacional - uma classificação que se fundamenta no número de soluções usadas em um mesmo tempo. Os algoritmos que usam uma única solução são chamados de métodos de trajetória, como o *variable neighbourhood search* e o *simulated annealing*. As metaheurísticas baseadas na busca populacional, como o algoritmo genético e o *scatter search* descrevem a evolução de um conjunto de soluções.
- Função objetivo dinâmica ou estática - é uma forma de se classificar segundo a forma do uso da função objetivo. Existem algoritmos que permanecem com a mesma função objetivo e outros que a modificam durante a busca da solução.
- Estrutura de vizinhança única ou variada - a maioria das metaheurísticas possuem uma única estrutura de vizinhança, em que se procura a solução, outros métodos tais como o *variable neighbourhood search* usam um conjunto de estruturas diferentes de vizinhança que permite a diversificação da busca.
- Métodos com maior ou menor uso de memória - classificação baseada no uso ou não de um histórico, característica que descreve o uso da história de busca. Há muitas formas de uso da memória, geralmente, é dividida entre memória de curto e longo prazo. A memória de curto prazo apenas é capaz de guardar passos recentes, como últimas soluções visitadas ou últimas decisões tomadas. Por outro lado, a memória de longo prazo guarda alguns dados da busca, que podem ser parâmetros ou soluções.

Neste trabalho considerou-se a classificação em dois grupos: de busca local e de busca populacional, que segundo Blum e Roli (2003) permite uma descrição mais clara dos algoritmos. Os métodos estudados de busca local são o *greedy randomized adaptive search procedures* (GRASP), o *simulated annealing* (SA), a busca tabu (BT) e o *variable neighbourhood search* (VNS). E os métodos de busca populacional são o algoritmo de colônia de formigas (ACO), o algoritmo genético (AG) e o *scatter search* (SS).

As técnicas metaheurísticas são amplamente utilizadas na solução de problemas de otimização combinatória. A seguir tem-se uma descrição das técnicas utilizadas neste trabalho.

3.2.1 Métodos de Busca Local

Greedy Randomized Adaptive Search Procedures (GRASP)

O GRASP é um método aleatório e iterativo, que combina uma heurística de construção e uma busca local. É composto de duas fases, a construtiva, que determina a solução que

será submetida à busca local, e a segunda fase do algoritmo, cujo objetivo é tentar obter alguma melhoria na solução corrente (RANGEL et al., 2000).

A cada iteração, o método cria uma solução inicial aleatória, guiada por uma função, e então, é aplicado o algoritmo de busca local visando melhorar esta solução. Este procedimento é repetido para um grande número de soluções iniciais. Assim o GRASP difere das outras metaheurísticas, devido à formação de uma nova solução não depender da obtida na iteração anterior.

A fase de construção, na qual uma solução é gerada elemento a elemento, é a fase mais crítica que demanda um maior esforço computacional. A criação de uma solução inicial se dá da seguinte forma: para cada componente viável que poderá constituir a solução é calculado o valor de uma função, chamada função gulosa. Essa função gulosa seleciona sequencialmente o elemento que minimiza o custo de incremento da solução parcial. Então, a cada passo, a solução parcial adicionada de um novo elemento, é avaliada pela função gulosa e as melhores são mantidas em uma lista, denominada lista restrita de candidatos (RCL - *restricted candidate list*). Um elemento desta lista é aleatoriamente escolhido como a solução parcial, não sendo necessariamente o primeiro da lista. Esse passo é novamente executado até que seja gerada uma solução completa. Através dessa escolha aleatória, pode-se obter uma melhor diversificação de soluções.

Essa fase de construção pode ser dita dinâmica, variando o valor da função gulosa a cada adição de um novo elemento, ou ainda estática que fixa o valor de cada elemento antes do início desta fase.

Um fator importante do GRASP é a qualidade dos elementos da lista restrita de candidatos. Essa lista pode ser limitada por um número de elementos ou pela qualidade dos elementos que a compõem. Se a lista for limitada a um elemento, a solução encontrada será a única solução e não haverá uma diversificação da solução. Se, ao contrário, a lista for ampla, serão geradas várias soluções diferentes produzindo uma maior variação.

Formada a solução inicial, tem-se a segunda fase em que realiza-se uma melhoria da solução aplicando-se um procedimento de busca local. A vizinhança da solução é analisada, se não houver solução melhor então esta é localmente ótima, se houver uma melhor solução esta é adotada e realiza-se uma nova busca na vizinhança da nova solução até que não haja nenhuma solução melhor.

De acordo com Resende (2001), a chave do sucesso do algoritmo de busca local consiste na escolha da melhor estrutura de vizinhança, de uma técnica eficiente de busca na vizinhança e da solução inicial.

A busca na vizinhança pode ser implementada utilizando as estratégias *best-improving* ou *first-improving*. No caso da estratégia *best-improving*, todos os vizinhos são analisados e

é adotado o melhor entre eles. E no caso da estratégia *first-improving*, é adotado a primeira solução cujo valor da função é melhor que da solução atual. Na prática, observa-se que ambas as estratégias resultam na mesma solução final, entretanto a *first-improving* requer um menor tempo computacional, enquanto o *best-improving* converge prematuramente para um ótimo local (RESENDE; RIBEIRO, 2003).

A principal vantagem do GRASP é o fato de não depender de soluções de iterações anteriores e trabalhar de forma aleatória permitindo a procura da solução em um espaço de busca mais amplo e diversificado, fazendo pouco uso de memória, pois apenas guarda a melhor solução encontrada.

O GRASP é um método efetivo, se as duas condições a seguir forem satisfeitas (BLUM; ROLI, 2003):

- O mecanismo de construção da solução gerar soluções dentro da melhor região do espaço de busca;
- As soluções geradas pela heurística construtiva pertencerem às regiões que tendem a diferentes ótimos locais.

Na fase de busca local, é possível utilizar outras metaheurísticas, de forma híbrida, que podem resultar em uma busca mais eficiente.

Algoritmo propriamente dito

O algoritmo simplificado do GRASP, para problemas de minimização, é dado no quadro 3.1.

```

Início (g(), f())
  f* = ∞
  Enquanto (critério de parada)
    Criação da solução i (g(i)):
    Elemento a elemento através de uma função gulosa.
    Faz-se uma Busca Local : solução j (f(j))
    se (f(j) < (f*))
      então i* = j
      f* = f(j)
    senão rejeitar j
  fim {se}
fim {enquanto}
fim

```

Quadro 3.1 - Algoritmo GRASP (RESENDE, 2001)

Simulated Annealing (SA)

Simulated annealing (SA) é um procedimento de busca aleatória local, utilizado em otimização combinatória. SA é como um algoritmo de descida: inicia-se com uma solução inicial, gera-se outras soluções vizinhas, calcula-se o custo, e se o custo for menor, então esta nova solução é aceita, caso contrário a solução é descartada. O processo é repetido até que não haja melhora, entretanto, o ótimo obtido é local, não há garantias de que seja o ótimo global.

Para evitar o ótimo local, normalmente obtido pelo algoritmo de descida, o *simulated annealing* pode aceitar uma configuração que forneça um valor 'pior' para a função objetivo, evitando assim a convergência para um mínimo local. Essa escolha é determinada por um número aleatório, e controlada através de uma probabilidade, que diminui ao longo da busca (YAMAMOTO, 2004).

Analogia ao processo físico

O algoritmo SA é inspirado no processo natural de busca por um estado de menor energia interna que ocorre quando um metal torna-se sólido. Na solução de um problema combinatório, com SA, os diferentes estados do sólido correspondem às diferentes soluções do problema, e a energia do sistema corresponde à função objetivo a ser minimizada.

Em 1953, Metropolis et al. (EGLESE, 1990) apresentaram um algoritmo que simula o processo de busca do equilíbrio térmico de um grupo de átomos a uma dada temperatura. A cada iteração é calculada a diferença, δ , da energia do sistema, onde um átomo é aleatoriamente substituído. Se $\delta < 0$, a substituição é aceita, porém, se $\delta > 0$, a mudança é aceita com a probabilidade $e^{-\delta/k.T}$, onde T é a temperatura e k é a constante de Boltzmann. Aplicando-se o algoritmo de Metropolis várias vezes, a cada temperatura são executadas várias iterações, e o sistema encontra o equilíbrio térmico para cada temperatura.

Assim, a analogia dos diferentes estados do metal corresponde a uma diferente solução viável de um problema de otimização combinatória e a energia do sistema corresponde à função objetivo a ser minimizada. O primeiro a utilizar este algoritmo para otimização foi Kirkpatrick, em 1983, que o denominou algoritmo *simulated annealing* (NARA, 1999).

A tabela 3 mostra as relações entre a nomenclatura dos elementos usada no processo físico e na otimização.

A execução do algoritmo começa com um valor alto de temperatura T , refletindo a aceitação de um grande número de soluções, e à medida que o valor da temperatura decresce as soluções que possuem uma maior energia diminuem sua probabilidade de aceitação. O processo que, inicialmente, trabalha com uma enorme aceitação de soluções, tende a evitar caminhar apenas para um mínimo local. Ao iniciar com uma temperatura relativamente alta, e com

Tabela 3 – Relações entre o processo físico e o processo de otimização funcional

Processo Físico	Processo de Otimização
Estado	Solução
Energia	Custo ou Função Objetivo
Estado de Transição	Soluções Vizinhas
Temperatura	Parâmetro de Controle
Ponto de Congelamento	Solução Heurística

decréscimo gradual, espera-se que ao final o sistema estacione em um estado de energia globalmente mínima, por analogia com a física. Através dessa aplicação, pode-se ignorar a constante de Boltzmann, e reescrever a probabilidade de aceitação de uma ordenação de energia mais alta como $e^{-\delta/T}$ (BARBOSA, 1989).

A temperatura T diminui segundo a equação $T^{k+1} = \rho T^k$, onde ρ normalmente assume valores de 0,80 a 0,99, até alcançar o ponto de congelamento, quando o algoritmo pára. Este ponto de congelamento não necessariamente implica na temperatura zero, podendo ser considerada uma temperatura pré-determinada muito baixa.

Algoritmo propriamente dito

O algoritmo simplificado do *simulated annealing*, para uma problema de minimização, é dado no quadro 3.2.

```

Início (i,energia(i),T,  $\alpha$ )
enquanto (critério de parada)
  para k=1:L (nº de iterações por nível de temperatura)
    gerar vizinho(j de i)
    calcular energia(j)
     $p = e^{(energia(i)-energia(j))/T}$ 
    se (energia(j)<energia(i) |  $p > \text{rand}(0,1)$ )
      então i =j
      energia(i)= energia(j)
    senão rejeitar j
  fim {se}
fim {para}
 $T = \alpha.T$ 
fim {enquanto}
fim

```

Quadro 3.2 - Algoritmo SA (YAMAMOTO, 2004)

A função $\text{rand}(0,1)$ calcula um número aleatório uniformemente distribuído entre 0 e 1.

Busca Tabu (BT)

A técnica busca tabu foi desenvolvida após uma observação, nos anos sessenta, da forma do ser humano agir na solução de problemas, usando a memória e crenças (GLOVER, 1998). O BT é um algoritmo determinístico de busca local que usa um histórico de busca para fugir do ótimo local e para implementar a estratégia de busca.

O BT começa por uma solução inicial, e a cada passo é criado um conjunto de soluções chamado 'vizinhança' da solução atual, através de uma função chamada 'move' que transforma uma solução em outra solução. Assim, é realizada uma seleção na vizinhança da solução, onde a melhor solução encontrada é aceita. O vizinho escolhido é considerado como solução atual para o próximo passo.

A nomenclatura 'lista tabu' é dada à memória de curto prazo do algoritmo, que armazena um histórico das últimas soluções visitadas, dentro de um número fixo de passos, não permitindo uma busca cíclica, isto é, não permitindo que a busca retorne às soluções recentemente visitadas. A cada nova solução vizinha, a solução anterior é inserida como o último elemento da lista, e o primeiro elemento é excluído, isto é, a lista é atualizada a cada passo. Os elementos da lista são considerados tabus, isto é, não é permitido adotar as soluções pertencentes à lista. Devido à dinâmica do conjunto de possíveis soluções na vizinhança, o BT pode ser considerado como uma técnica de busca dinâmica da vizinhança (BLUM; ROLI, 2003).

O número de elementos da lista tabu controla a memória do processo de busca. Se pequeno, a busca ficará concentrada em pequenas áreas do espaço de busca. Do contrário, se for grande, a busca será forçada a explorar amplas regiões, devido às proibições de revisão de um número muito grande de soluções pertencentes à lista. Blum e Roli (2003) citam exemplos de trabalhos que permitem a variação do número de elementos da lista durante a busca.

Algumas vezes a lista tabu pode ser muito restritiva, pois ao mesmo tempo em que evita soluções previamente adotadas, também impede a movimentação em direção às novas soluções. Devido a uma possível proibição de um passo importante na busca, há uma função, denominada função de aspiração, que analisa o lucro obtido na aceitação de uma solução da lista, e pode permitir sua adoção mesmo pertencendo à lista tabu. Ou seja, a restrição tabu é ignorada e a solução é aceita. Um critério de aspiração bastante utilizado é o de ignorar a restrição tabu sempre que a solução proibida for melhor que a solução encontrada até o momento. A aplicação adequada desses procedimentos é fundamental para se atingir níveis de desempenho alto no BT (CORREA et al., 2004).

O critério de parada do algoritmo pode ser: achar uma solução próxima da esperada, ou a realização de um número máximo de iterações, ou ainda limitar o tempo máximo de processamento.

Algoritmo propriamente dito

O quadro 3.3 mostra o algoritmo simplificado da busca tabu, para uma problema de minimização.

```

Início (i,f(i),A(f(i)),nV,nT)
  enquanto (critério de parada)
    gerar vizinhança V de i
    calcular f(V)
    f(j)= min(f(V))
    se o movimento não for tabu ou j atenda a função de
    aspiração A(f(i))
      se (f(j) < f(i))
        então i =j
        f(i)= f(j)
      senão rejeitar j
    fim {se}
  Atualizar Lista Tabu T
  fim {se}
fim {enquanto}
fim

```

Quadro 3.3 - Algoritmo BT

Neste quadro o n_V é o número de soluções do conjunto vizinhança da solução atual, e n_T é o número de elementos da lista tabu.

Variable Neighborhood Search (VNS)

Mladenovic e Hansen (1997) propuseram um método simples, baseado na mudança sistemática das vizinhanças usadas na busca local, denotado *Variable Neighborhood Search* (VNS). Este método tem como principal característica variar as estruturas de vizinhança ao invés de utilizar apenas uma estrutura fixa como é normal nas buscas locais. A figura 4 mostra as várias estruturas de vizinhança (V_1, V_2, V_k) de i do VNS.

Contrário a outros métodos de busca local, o VNS não segue uma trajetória, mas explora vizinhanças gradativamente mais distantes da solução atual. E apenas focaliza a busca em torno de uma nova solução se realmente esta apresentar uma melhora. A sistemática de mudança das vizinhanças aplicada na busca permite fugir de um ótimo local.

Na figura 5(a) observa-se que a partir de i analisou-se uma vizinhança V_1 e ao encontrar uma solução que apresentou melhora ela é adotada como solução atual i e novamente é analisada uma vizinhança V_1 da nova solução atual i . A figura 5(b) mostra que ao iniciar a busca pela solução i , se não for encontrada uma solução melhor dentro da vizinhança V_1 , muda-se a estrutura de vizinhança para V_2 e efetua-se uma nova busca até encontrar uma solução melhor

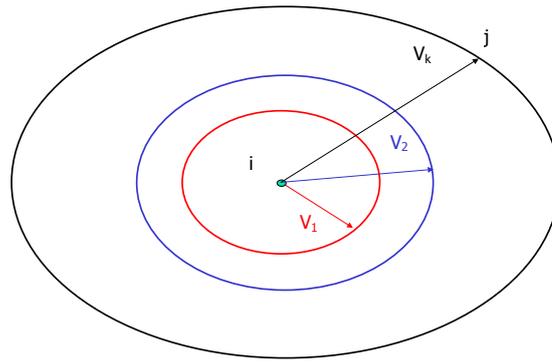


Figura 4 – As várias estruturas diferentes de vizinhança do VNS

que a atual, ao encontrá-la ela é adotada como nova solução, e a partir dela é realizada uma nova busca iniciando pela estrutura de vizinhança V_1 .

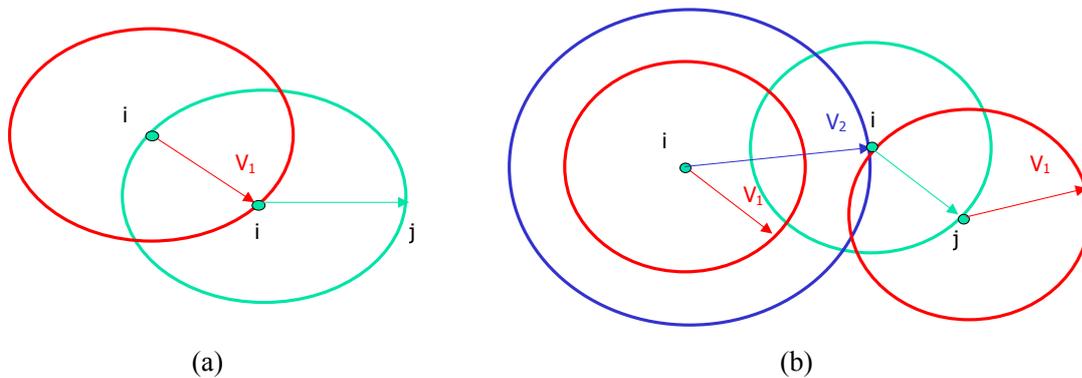


Figura 5 – Sistemática de mudança das vizinhanças

O VNS apresenta um comportamento estocástico na seleção aleatória de um vizinho, e um comportamento determinístico na aplicação da busca local em cada iteração (POLACEK et al., 2006).

Através da escolha aleatória de um elemento vizinho, evita-se a busca cíclica que poderia ocorrer quando se usa uma forma determinística de escolha (AVANTHAY et al., 2003).

Os passos básicos do VNS são a inicialização e o ciclo principal, que é composto por três fases: o *shaking*, a busca local e o *move*.

Na inicialização, as estruturas de vizinhança são definidas e são selecionadas as estruturas V_k ($k=1, \dots, K$) que serão usadas na busca. É fornecida uma solução inicial i e escolhe-se a condição de parada, faz-se $k=1$. Então, o ciclo é repetido até alcançar a condição de parada. Na fase *shaking* é gerada uma nova solução j aleatoriamente na vizinhança de i ($j \in V_k(i)$). Assim, o j torna-se o ponto inicial da busca local. Na próxima fase aplica-se um método de busca local, que pode usar qualquer estrutura de vizinhança não sendo limitado pelas estruturas previamente definidas, e denota-se como j^* , o ótimo local obtido. Em seguida, na fase *move*, se o ótimo local

j^* for melhor que i , então adotar $i=j^*$, $k=1$, e continua-se a busca. Senão, descarta-se j^* e faz-se $k=k+1$, retornando à fase *shaking* com uma nova vizinhança.

O objetivo da fase *shaking* é perturbar a solução visando encontrar uma melhor solução inicial para a busca local. Essas soluções iniciais, na busca, devem corresponder a diferentes ótimos locais, entretanto não deve representar um valor muito afastado da solução atual para que sejam mantidas algumas das características desta solução (BLUM; ROLI, 2003).

O uso de uma nova vizinhança, caso não haja melhora na solução, corresponde à diversificação na busca. A eficácia dessa estratégia de vizinhança dinâmica pode ser explicada, segundo Blum e Roli (2003), pelo fato de uma solução localmente ótima para uma vizinhança provavelmente não é localmente ótima para outra vizinhança. Assim, cada vizinhança define um espaço de busca diferente.

Algoritmo propriamente dito

O algoritmo simplificado do VNS, para uma problema de minimização, é dado no quadro 3.4.

```

Início ( $i$ ,  $f(i)$ ,  $V_k$ ,  $k=1$ )
  enquanto (critério de parada)
    gerar  $j$  aleatoriamente da vizinhança  $V_k$  de  $i$ 
    aplicar uma busca local a partir de  $j$ , e guardar o ótimo local  $j^*$ 
     $f(j^*) = \min(f(V_k))$ 
    se ( $f(j^*) < f(i)$ )
      então  $i = j^*$ 
       $f(i) = f(j^*)$ 
    senão rejeitar  $j$  e fazer  $k=k+1$ 
  fim {se}
fim {enquanto}
fim

```

Quadro 3.4 - Algoritmo VNS

3.2.2 Métodos de Busca Populacional

Colônia de Formigas (ACO - *Ant Colony Optimization*)

Este método, proposto por Dorigo et al. (1999), foi desenvolvido a partir de um experimento real em uma colônia de formigas, em que observou-se a capacidade de adaptação das formigas às mudanças em seu meio. Ao ser introduzido um obstáculo no caminho da busca de comida, havendo dois caminhos de comprimentos diferentes, as formigas, após uma fase inicial, percorrem sempre o caminho mais curto do ninho até a comida (DORIGO et al., 1999). A figura 6 mostra a reação das formigas quando é inserido um obstáculo em seu caminho.

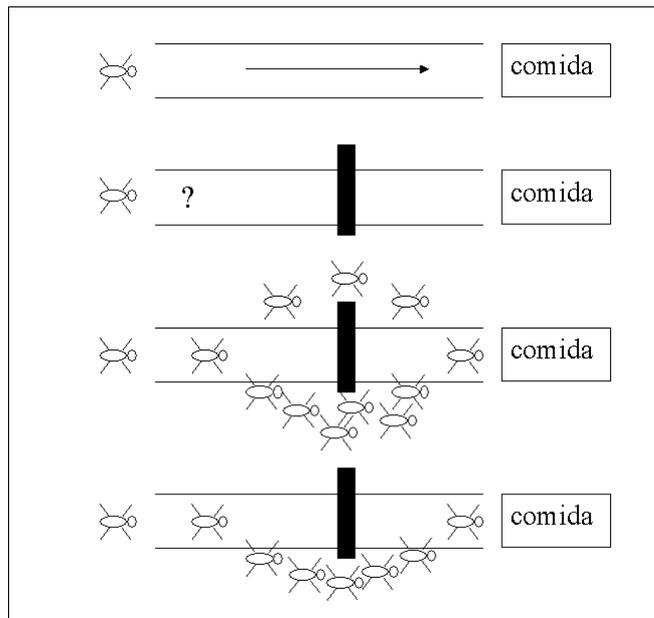


Figura 6 – A escolha do caminho mais curto pela colônia de formigas

Ao analisar uma colônia de formigas observou-se que as formigas não agem de forma independente na busca da comida e há um elemento que coordena a trajetória a ser percorrida. A forma de comunicação entre elas é uma substância chamada de feromônio que é depositada por cada formiga no caminho por ela percorrido. Uma formiga irá detectar o rastro de feromônio e tem a tendência de escolher o caminho que tenha uma quantidade maior de feromônio depositada.

O ACO é uma forma artificial de representação de uma colônia de formigas aplicada na solução de um problema. A solução é gerada de forma aleatória, permitindo criar uma diversidade de soluções. Essa geração é obtida elemento a elemento, e o próximo elemento que irá compor a solução é adotado por uma evolução heurística, que através de uma certa probabilidade, escolherá o caminho a ser seguido. A probabilidade da adoção do caminho aumenta com o aumento do número de formigas que o adotou, ou seja, com o aumento da quantidade de feromônio depositada no caminho. O processo também leva em consideração a evaporação de feromônio a cada iteração. Portanto, quando uma solução é composta, o procedimento é reinicializado atualizando a quantidade de feromônio. Esse procedimento é repetido até que haja uma estabilidade na busca.

O ACO é um algoritmo que define propriedades na iteração de vários agentes. Cada agente simples é chamado de formiga e o conjunto de formigas constitui o sistema, cooperando numa atividade comum para resolver um problema. A principal característica do método é que a interação desses agentes gera um efeito sinérgico, pois a qualidade da solução obtida aumenta quando tais agentes trabalham juntos, interagindo entre si, para a solução de um mesmo problema (LOIOLA et al., 2004).

Na geração de uma solução, a seleção do próximo elemento que constituirá uma solução é dada através da seguinte regra probabilística (GOLDBARG; GOLDBARG, 2006):

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} \quad \text{se } j \in N_i^k$$

Onde:

- $p_{ij}^k(t)$ é a probabilidade do elemento j ser escolhido pela formiga k , atualmente no elemento i , durante a t -ésima iteração;
- $\tau_{ij}(t)$ é a quantidade de feromônio presente na escolha do elemento j a partir de ij na t -ésima iteração do procedimento;
- α é o parâmetro que regula a influência de $\tau_{ij}(t)$;
- η_{ij} é a visibilidade do elemento j com relação ao elemento i ;
- β é o parâmetro que regula a influência de η_{ij} ;
- $N_i^k(t)$ é o conjunto de elementos não adotados pela formiga k , atualmente em i , durante a t -ésima iteração.

A quantidade de feromônio é sempre atualizada a cada geração de uma solução. A evaporação do feromônio é um processo que diminui a intensidade de feromônio depositada com o passar do tempo. Esse processo faz-se necessário para evitar uma rápida convergência do algoritmo, que pode direcionar a busca para uma região sub-ótima. O processo de evaporação do feromônio ocorre da seguinte maneira (GOLDBARG; GOLDBARG, 2006):

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

Sendo $\Delta\tau_{ij}(t) = \sum_{k=1}^M \Delta\tau_{ij}^k(t)$ e $\Delta\tau_{ij}^k(t) = Q$, se a formiga k adota o elemento j a partir de i .

Onde:

- t é a iteração atual;
- $\rho \in [0,1]$ é o parâmetro que regula a redução de $\tau_{ij}(t)$;
- Q é a quantidade de feromônio excretada pela formiga a cada iteração;
- M é o número de formigas;
- $\Delta\tau_{ij}^k(t)$ é o ganho de feromônio do elemento i para o j , causado pela formiga k na t -ésima iteração.

Pode-se incluir no algoritmo um processo chamado *daemon action* para implementar ações que não poderão ser realizadas pelas formigas. Como por exemplo a inserção de um procedimento de busca local, ou um armazenamento global de informações usados para mensurar um depósito adicional de feromônio. O processo de atualização do feromônio através do *daemon action* é denominado *offline pheromone updates* (DORIGO; CARO, 1999).

Algoritmo propriamente dito

O algoritmo simplificado da colônia de formigas, para uma problema de minimização, é dado no quadro 3.5 a seguir.

```

Início (i, f(i),M)
  enquanto (critério de parada)
    repita até que todas as M formigas encontrem uma solução completa j
      selecione o próximo elemento da solução, segundo uma regra
      probabilística
    fim {repita}
  calcular o valor da função f(j) para cada uma das M formigas
  para j=1 até M
    se (f(j) < f(i))
      então i =j
      f(i)= f(j)
    senão rejeitar j
  fim {se}
  fim {para}
  atualizar o processo de depósito e evaporação do feromônio
  fim {enquanto}
fim

```

Quadro 3.5 - Algoritmo ACO

Algoritmo Genético (AG)

O AG é um algoritmo de busca baseada na evolução natural e na genética. O segredo da adaptação e sobrevivência das reproduções é um exemplo das pesquisas do estudo biológico, onde os mais fortes sobrevivem e os mais fracos são eliminados. Assim, a população evolui de forma a melhorar em função de uma determinada característica dentro do meio em que vive (MITCHELL, 1997).

Na solução de um problema particular, o AG deve possuir cinco componentes (MICHALEWICZ, 1996):

- Uma representação genética das soluções potenciais do problema;
- Uma forma de gerar uma população inicial de soluções potenciais;

- Uma função de evolução, a qual classifica as soluções em função da adaptabilidade ou *fitness*;
- Operadores genéticos que formem os filhos;
- Valores para parâmetros (tamanho da população, probabilidades da aplicação de operadores genéticos, etc.).

Representação cromossômica e inicialização

Cada solução possível, dentro de um espaço de busca, é representada como uma seqüência de elementos, onde cada elemento é chamado de gene, e cada uma dessas seqüências formada pelos genes é um cromossomo, também chamados de indivíduo, assumindo, assim que cada indivíduo é formado por um único cromossomo. Cada indivíduo distinto é codificado por uma seqüência diferente de genes.

Na utilização do algoritmo genético, primeiramente, define-se um grupo de soluções factíveis, que denomina-se população inicial, formada por um grupo de indivíduos, ou cromossomos, distintos.

A população formada pelas configurações candidatas à solução do problema de otimização deve ser ordenada, do melhor ao pior indivíduo, de modo que, em um problema de minimização de uma função, o indivíduo que apresentar menor valor da função objetivo é considerado o melhor indivíduo, e tem a maior possibilidade de sobrevivência no ambiente, portanto localizando-se na primeira posição da ordenação.

O valor da função objetivo de cada indivíduo é denominado de *fitness*, que pode ser entendido como adequabilidade ao ambiente. Em um problema de minimização, a finalidade é obter a solução que apresente o menor *fitness*, portanto o indivíduo que apresenta menor *fitness* é o de maior adequação ao sistema.

Desta lista de indivíduos haverá reproduções as quais formarão novos indivíduos transmitindo parte de seu material genético às gerações futuras. Essas novas gerações serão analisadas de modo que a população seja sempre formada pelos indivíduos que apresentem melhor *fitness*, ou sejam mais adequados ao sistema.

Seleção

Para o processo de reprodução do algoritmo genético é realizada uma seleção dentre o grupo de indivíduos pertencentes à população, assim, através de dois indivíduos selecionados faz-se uma troca genética para geração de novos indivíduos.

Neste trabalho a seleção dos dois indivíduos que efetuarão o processo de reprodução

será executada através da fórmula proposta por Mayerle, de acordo com Correa et al. (2004), dada por:

$$\text{Indivíduo selecionado} = m + 1 - \left\lfloor \frac{-1 + \sqrt{1 + 4 * \text{rand} * (m^2 + m)}}{2} \right\rfloor,$$

Onde: m é o número de indivíduos da população, rand é um número aleatório uniformemente distribuído entre 0 e 1, e o símbolo $\lfloor b \rfloor$ é o maior número inteiro menor igual a b . Por exemplo, se $m=5$ e $\text{rand} = 0,5$, tem-se:

$$\begin{aligned} \text{Indivíduo selecionado} &= 5 + 1 - \left\lfloor \frac{-1 + \sqrt{1 + 4 * 0.5 * (5^2 + 5)}}{2} \right\rfloor \\ &= 5 + 1 - \lfloor 3.4051 \rfloor \\ &= 5 + 1 - 3 \\ &= 3 \text{ (terceiro indivíduo)} \end{aligned}$$

Essa fórmula favorece a seleção dos primeiros indivíduos da lista, isto é os indivíduos com melhores *fitness*.

Crossover ou Recombinação

O *crossover* é o processo de reprodução realizada pelos indivíduos selecionados pelo processo anterior. Neste processo há uma troca de genes entre os dois indivíduos selecionados, que são denominados os pais, representando um processo sexuado, formando novos indivíduos que são chamados de filhos.

Após a seleção dos pais, a troca dos seus genes ocorrem de forma aleatória, onde os elementos são escolhidos aleatoriamente para a formação dos filhos, como é ilustrado na figura 7 (TANOMARU, 1995), onde à esquerda tem-se os pais e à direita os filhos.

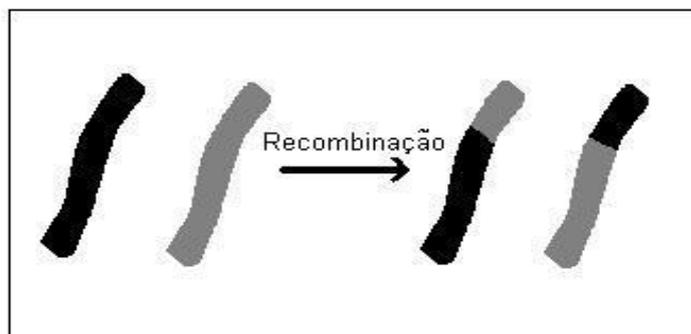


Figura 7 – Processo de *crossover* ou recombinação (Fonte: Tanomaru,1995)

Mutação

A mutação é um processo de busca aleatória. Esta operação visa manter e proporcionar a diversidade da população. Durante a mutação, escolhe-se aleatoriamente um indivíduo e realiza-se uma alteração de um ou mais genes, como demonstra a figura 8 (TANOMARU, 1995), assim, formando um novo indivíduo que será analisado quanto à adequabilidade ao ambiente. Os indivíduos assim formados introduzem novos elementos, que continuarão a fornecer o material genético às gerações futuras.

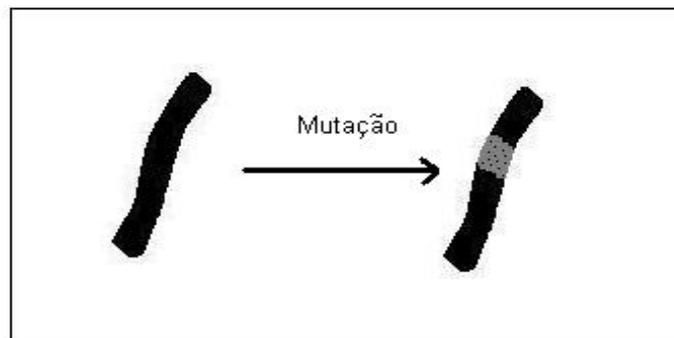


Figura 8 – Processo de mutação (Fonte: Tanomaru,1995)

Técnica Elitista

A técnica elitista visa garantir, na população, a permanência de um percentual de melhores indivíduos, podendo reproduzi-los na população seguinte. Esta estratégia evita que um bom indivíduo seja perdido em uma operação genética mal sucedida.

Em AG existem duas formas de inserção de novos indivíduos na população, os quais são conhecidos na literatura como: "geracional" e "em regime" (steady-state) (CASTRO, 2001b).

- Algoritmo Genético Geracional - toda a população é substituída por novos indivíduos gerados pelo processo de seleção e aplicação dos operadores genéticos, freqüentemente é aplicada uma técnica elitista;
- Algoritmo Genético em Regime - apenas um indivíduo é gerado, avaliado, e será inserido na população se for melhor que algum dos indivíduos da população, neste caso, o pior é descartado. Caso o novo indivíduo seja pior, segue-se para uma nova geração de indivíduos.

Algoritmo propriamente dito

O quadro 3.6 mostra o algoritmo simplificado do algoritmo genético, para uma problema de minimização.

```

Início
  gerar população inicial, composta por  $m$  indivíduos
  calcular fitness dos indivíduos da população
  ordenar os indivíduos, em ordem crescente do valor do
  fitness
  enquanto (critério de parada)
    guardar os  $e$  melhores indivíduos
    enquanto (gerar  $(m-e)$  novos indivíduos)
      seleção
      crossover ou mutação, para gerar o(s) filho(s)
      calcular fitness do(s) filho(s)
    fim {enquanto}
  inserir o(s) e indivíduos elite na nova população reordenar os
  indivíduos da população
  fim {enquanto}
fim
  
```

Quadro 3.6 - Algoritmo AG

Scatter Search (SS)

O *scatter search* é um método evolutivo, que combina regras de decisões e restrições com o objetivo de executar uma combinação de elementos para gerar uma solução melhor, ao invés de basear-se apenas na solução inicial. Essa estratégia de combinação tem a função de evoluir as funções de geração e modificação de soluções.

É um método diferente dos algoritmos evolucionários, em específico o algoritmo genético, pois a geração de novos candidatos à solução não é aleatória, segue uma estratégia de construção da solução no espaço Euclidiano. Ainda apresenta a vantagem de possuir mecanismos de memória adaptativa como a técnica de busca tabu (GLOVER et al., 2000).

O *scatter search* é uma estratégia de busca que gera novas soluções a partir da combinação de um conjunto de soluções chamado de conjunto referência, que são soluções viáveis do problema em questão. Diferente da população do algoritmo genético, o conjunto referência é relativamente pequeno. As soluções resultantes são chamadas soluções experimentais (*trial solutions*) (BLUM; ROLI, 2003). Estas soluções podem ser inviáveis, devendo, então, ser modificadas por procedimentos de reparos que as transformarão em soluções viáveis. Assim, é aplicado um mecanismo de melhora das soluções geradas, que pode ser um método de busca local. E um novo conjunto de soluções referência será selecionado à partir do conjunto referência anterior e do conjunto das novas soluções.

A metodologia *scatter search* visa:

- achar o ótimo através da diversificação de soluções elites;
- ampliar o espaço de busca, através da combinação de soluções, além do uso de processos heurísticos na criação de novas soluções permitindo maior diversidade e qualidade;
- operar com a possibilidade de exportação de dados de várias soluções elites na geração de novas soluções por combinações.

O SS possui várias estratégias de exploração, de acordo com Laguna (2002):

- Método de diversificação de geração;
- Método de melhoramento;
- Método de renovação do conjunto referência;
- Método de geração de subgrupos que operam dentro do conjunto referência;
- Método de combinação de soluções.

Método de diversificação na geração

O SS utiliza aleatoriedade controlada e memória de frequência para gerar um conjunto de soluções diversificadas. O conjunto de possíveis valores de cada variável é dividido em grupos de mesmo tamanho, e um destes grupos de valores será escolhido aleatoriamente. Assim, o valor da variável em questão é selecionado aleatoriamente dentro do grupo previamente selecionado.

Este método é usado na geração da população inicial e visa a diversificação e não a qualidade das soluções.

Método de melhora

Este método visa realizar uma otimização local para cada solução, obtendo uma melhora nos valores da função objetivo. De acordo com Glover et al. (2003), para exemplos pequenos, a aplicação dos métodos de diversificação e melhora são suficientes para encontrar o ótimo global.

Método de atualização do conjunto referência

O conjunto referência é formado por ambas as soluções de boa qualidade e soluções diversificadas, que serão utilizadas na geração de novas soluções pelo método da combinação de soluções.

Método de geração dos subconjuntos

Este método gera os subconjuntos que serão usados no método da combinação de soluções para a criação de novas soluções. Este método cria conjuntos da seguinte forma (tabela 4):

- 1) Entre todos os subconjuntos de dois elementos;
- 2) Entre subconjuntos de três elementos derivados dos subconjuntos gerado pelo item anterior (1) e a melhor das soluções restantes;
- 3) Entre subconjuntos de quatro elementos derivados dos subconjuntos do item anterior (2) e a melhor das soluções restantes;
- 4) Subconjuntos compostos pelos melhores i elementos, onde i é um valor compreendido entre 5 e o número total de soluções do conjunto referência.

Tabela 4 – Tabela de geração de subconjuntos, dado as 5 soluções a,b,c,d,e do conjunto referência

Tipos	Subconjuntos
1	{a,b}, {a,c}, {a,d}, {a,e} {b,c}, {b,d}, {b,e} {c,d}, {c,e} {d,e}
2	{a,b,e}, {a,c,e}, {a,d,e} {b,c,e}, {b,d,e} {c,d,e}
3	{a,b,c,d}, {a,b,d,e} {a,c,d,e}
4	{a,b,c,d,e}

Método de combinação de soluções

Este método utiliza os subconjuntos gerados pelo método anterior. Cada subconjunto pode gerar uma ou mais soluções através de uma combinação linear entre soluções pertencentes à lista de referência.

Por exemplo: $x = x' \text{ ou } x'' \pm d$

onde: x é uma nova solução, x' e x'' são soluções referência, $d = r \frac{x'' - x'}{2}$ e r é um número aleatório entre 0 e 1.

Pode-se usar as regras na geração:

- os dois elementos x' e x'' pertencem ao conjunto referência;
- apenas um dos elementos x' e x'' pertencem ao conjunto referência;
- nenhum dos elementos x' e x'' pertencem ao conjunto referência.

Após a criação de novas soluções, é aplicado o método de melhora.

Algoritmo propriamente dito

O algoritmo simplificado do *scatter search*, para um problema de minimização, pode ser visto no quadro 3.7.

<p>Início gerar um conjunto inicial de soluções (diversificado e de forma heurística), escolha para formar o conjunto referência enquanto (critério de parada) enquanto (houver mudanças) criar novas soluções baseadas na combinação das soluções referência aplicar um processo heurístico para melhorar as soluções do passo anterior se $f(j) < f()$ então adicionar ao conjunto referência a melhor solução do passo anterior fim {se} fim {enquanto} Renovar conjunto referência fim {enquanto} fim</p>
--

Quadro 3.7 - Algoritmo SS

Todas essas técnicas metaheurísticas serão aplicadas no problema de sequenciamento das bateladas em rede de dutos a fim de se escolher a que melhor pode solucionar o problema.

3.3 TÉCNICAS DE OTIMIZAÇÃO MULTIOBJETIVO

3.3.1 Problemas Multiobjetivos (PMO)

Muitos problemas reais geralmente envolvem mais de um objetivo, ou critério de desempenho. Esses problemas são chamados de problemas multiobjetivos (PMO).

Problemas com um único critério permitem encontrar apenas uma solução ótima que apresenta o melhor valor da função objetivo. Entretanto, problemas que envolvem múltiplos critérios, devido aos conflitos entre os objetivos, geralmente não possuem uma única solução ótima, porém muitas soluções ótimas (DEB, 1999).

Um problema de otimização multi-objetivo pode ser descrito por um vetor F de k objetivos que dependem de um vetor x de n variáveis de decisão e m restrições:

$$\text{Min}F(x) = (f_1(x), \dots, f_k(x))$$

$$\text{sujeito a } g_i(x) \leq 0, i = 1, \dots, m, x \in \Omega$$

A solução do PMO minimiza os componentes do vetor $F(x)$ em que x é um vetor de variáveis n -dimensionais ($x = x_1, \dots, x_n$) do universo Ω .

Os problemas multiobjetivos são caracterizados pelas formas de medição de desempenho (objetivos) que podem ser (in)dependentes e/ou não mensuráveis. Os múltiplos objetivos otimizados quase sempre conflitam, estabelecendo uma posição parcial, ao invés de total, no espaço de busca. Na realidade, a busca de um ótimo global para um PMO geral é um problema NP-Completo. A solução perfeita, onde todas as variáveis de decisão satisfazem todas as restrições e a função objetivo alcança o mínimo global, pode não existir (VELDHUIZEN, 1999).

Os PMOs requerem técnicas de otimização especializadas. Os modelos multiobjetivos permitem considerar simultaneamente todos os possíveis objetivos do problema (FERREIRA, 1999).

Um conceito importante na determinação de um conjunto de soluções do PMO é a Otimalidade de Pareto. O conjunto de soluções de um problema de otimização multiobjetivo consiste em todos os vetores de decisão para os quais não pode haver uma melhora com relação a um objetivo sem que haja uma degradação com relação a pelo menos algum outro objetivo. Estes vetores são os conhecidos como ótimos de Pareto (ZITZLER; THIELE, 1999).

3.3.2 Otimalidade de Pareto

Na figura 9, pode-se observar a ilustração de um problema de minimização com dois objetivos. O ponto A representa uma solução que possui um valor mínimo para a função objetivo 1, entretanto apresenta um valor alto para a função objetivo 2. A solução representada pelo ponto C, obteve um maior valor da função objetivo 2 e o menor valor da função objetivo 1. A solução B, é pior que a solução A com relação ao objetivo 1, porém é melhor com relação ao objetivo 2. Assim, podemos dizer que as soluções A, B e C fazem parte das soluções ótimas de Pareto. O ponto D representa uma solução que é pior que a solução B, com relação a ambos os objetivos, portanto não pertence às soluções de Pareto. Soluções como a D são consideradas soluções dominadas.

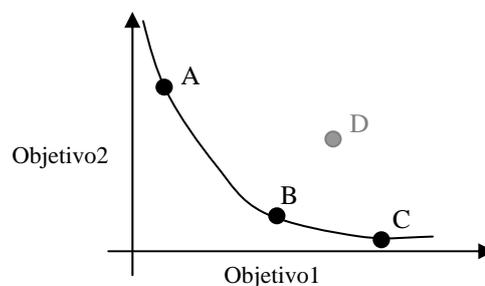


Figura 9 – Soluções Ótimas de Pareto para um problema com dois objetivos

Existem dois objetivos na otimização multiobjetivo:

- (i) buscar soluções próximas ao conjunto de soluções ótimas de Pareto;
- (ii) buscar soluções diferentes entre si.

Uma solução ótima deve apresentar, em um problema de minimização multiobjetivo, o menor valor com relação a todos os objetivos do problema. Uma solução dominada, como o ponto D do exemplo anterior, possui soluções melhores com relação a pelo menos um objetivo, não podendo, portanto, ser uma solução ótima.

Definição de soluções dominadas

Para um problema com mais de uma função objetivo (f_j , $j = 1, \dots, M$ e $M > 1$), quaisquer duas soluções $x(1)$ e $x(2)$ podem apresentar duas possibilidades: uma domina a outra ou nenhuma das soluções domina a outra (DEB, 1999).

Uma solução $x(1)$ é dita dominante em relação a outra solução $x(2)$, se são verdadeiras ambas as condições:

- 1) A solução $x(1)$ não é pior que $x(2)$ para todos $j = 1, \dots, M$ objetivo.
- 2) A solução $x(1)$ é estritamente melhor que $x(2)$ em pelo menos um objetivo, $j \in \{1, \dots, M\}$.

Se qualquer uma das condições for violada, a solução $x(1)$ não domina a solução $x(2)$.

Um conjunto de soluções não-dominadas é definido no contexto de uma amostra do espaço de busca, enquanto que o conjunto ótimo de Pareto é definido em relação a todo o espaço de busca.

Matematicamente, os vetores de decisão que não são dominados dentro do espaço de busca é denominado ótimo de Pareto e constituem o conjunto ou a fronteira ótima de Pareto (ZITZLER; THIELE, 1999).

Pela definição da otimalidade de Pareto, qualquer conjunto, não vazio, finito de soluções contém pelo menos uma solução ótima de Pareto, com respeito ao conjunto (VELDHUIZEN, 1999).

3.3.3 Algumas Técnicas Importantes de Aproximação

Muitos PMOs são de alta dimensão, descontínuos, multimodais, e/ou NP-Completo. Métodos determinísticos são geralmente ineficientes quando aplicados a problemas NP-Completo ou outros de grandes dimensões pois são limitados pelas suas necessidades de conhecimento do domínio do problema para direcionar ou limitar a busca deste amplo espaço de busca. Problemas que exibem uma ou mais características dessas são tidos como irregulares.

Devido a muitos PMOs reais serem irregulares, métodos baseados em buscas enumerativas e determinísticas não são adequadas para suas soluções. Algoritmos baseados em buscas estocásticas e otimizações aproximadas como as técnicas simulated annealing, busca tabu e computação evolucionária foram desenvolvidas como uma alternativa de aproximação para a solução de problemas irregulares. Os métodos estocásticos requerem uma função que determine um valor de aptidão para as possíveis (ou parciais) soluções, e um mecanismo de codificação entre o problema e o domínio do algoritmo. Entretanto não se pode garantir a solução ótima. Geralmente obtêm-se boas soluções para um grande número de problemas em que a busca determinística tradicional tem dificuldade (VELDHUIZEN, 1999).

Para a solução de problemas de otimização multiobjetivo têm-se, na literatura (COELLO, 2000; ZITZLER; THIELE, 1999; FONSECA; FLEMING, 1995), três tipos de abordagens:

- abordagem por agregação de objetivos, em que os objetivos são artificialmente combinados, ou agregados, por uma função escalar;
- abordagem não-agregada, que calcula um conjunto não Pareto, estas abordagens usam um conjunto de soluções tratando separadamente os múltiplos objetivos;
- abordagem baseada em Pareto, em que deseja-se gerar um conjunto de soluções não-dominadas, fazendo o uso da otimizabilidade de Pareto.

A seguir tem-se alguns exemplos dessas abordagens:

1. Abordagem por agregação de objetivos:

- Método das Ponderações
- Goal Programming
- Goal Attainment
- Método ϵ -Constraint

2. Abordagens Não-Agregadas:

- VEGA (Vector Evaluated Genetic Algorithm)
- Lexicographic Ordering
- Teoria dos Jogos
- Aproximação Min-Max com pesos e suas variações

3. Abordagens baseadas em Pareto:

- MOGA (multiple objective genetic algorithm)
- NSGA (Nondominated Sorting Genetic Algorithm)
- NPGA (Niched Pareto Genetic Algorithm)
- SPEA (Strength Pareto Evolutionary Algorithm)

As soluções finais de um problema multiobjetivo resultam de dois processos: de decisão e de otimização. Segundo Veldhuizen (1999), os problemas de otimização multiobjetivo podem ser tratados de três formas diferentes segundo o processo de decisão, em que as preferências do decisor são consideradas antes, durante ou depois do processo de otimização, como pode ser visto a seguir:

- Técnicas *a priori* : requerem a definição da importância dos objetivos anteriormente a busca, o que, na realidade, não é uma tarefa trivial.
- Técnicas Progressivas : técnicas iterativas de busca que incorporam as preferências do tomador de decisão.
- Técnicas *a posteriori* : oferecem um conjunto de soluções para futura escolha da solução pelo tomador de decisões.

Levando-se em consideração a falta de informações qualitativas e quantitativas dos problemas multiobjetivo do mundo real, observa-se que a abordagem mais apropriada do problema é pelo uso de técnicas *a posteriori* baseadas em Pareto.

Muitos métodos não são eficientes na otimização multiobjetivo, pois:

- i. muitos métodos não geram múltiplas soluções em uma única iteração, o que iria requerer um grande número de iterações na geração das soluções ótimas de Pareto;
- ii. a realização de inúmeras iterações desses métodos não garante a geração das soluções de Pareto diversificadas;
- iii. muitos métodos não são eficientes em problemas com variáveis discretas e em problemas com múltiplos objetivos.

A otimização de vários objetivos obtida através de uma única solução pode ocasionar a geração de uma solução ótima não aceitável, possivelmente devido a uma função objetivo utilizada não englobar alguma característica do problema ou não atribuir os pesos corretos aos objetivos analisados. Assim, uma solução somente seria gerada após muitas tentativas ou muitas execuções das rotinas de otimização.

O uso de algoritmos evolutivos, como o algoritmo genético, por se tratar de busca populacional, pode gerar um conjunto de soluções de Pareto em apenas uma única iteração. Com o uso de mecanismos de diversificação e preservação pode-se gerar inúmeras soluções diferentes. Konak et al. (2006) apresentam uma visão geral do uso do algoritmo genético como base de técnicas de otimização multiobjetivo.

Métodos Evolucionários

Existem vários métodos evolucionários na literatura, para a solução de problemas multiobjetivo, sendo os mais conhecidos segundo Coello (1999b): o VEGA (*Vector Evaluated Genetic Algorithms*); o MOGA (*Multiobjective Optimization Genetic Algorithm*); o NPGA (*Niched Pareto Genetic Algorithm*); o NSGA (*Nondominated Sorting Genetic Algorithm*); e o SPEA (*Strength Pareto Evolutionary Algorithm*). Esses serão brevemente descritos a seguir.

VEGA

O primeiro algoritmo evolucionário aplicado na solução de problemas multiobjetivos, denominado *Vector Evaluated Genetic Algorithms* (VEGA), foi proposto por Schaffer (apud Coello, 1999a). Este algoritmo é uma extensão do algoritmo genético simples diferenciando-se apenas na forma de execução da seleção (COELLO, 1999b). A cada iteração é criado um número de indivíduos agrupados em sub-populações, para cada objetivo separadamente, assim para um problema com k objetivos, tem-se k sub-populações de tamanho N/k (assumindo que N é o tamanho total da população). Os indivíduos são misturados e a partir destes são sorteados os indivíduos que farão as operações de recombinação e mutação, na forma usual. Com isto as soluções obtidas são não-dominadas com relação a apenas a população corrente (COELLO, 1999b).

Essa geração de soluções pode prejudicar a evolução das espécies pois pode descartar soluções que poderiam ser utilizadas na geração de soluções promissoras, porém que apresentem uma performance mediana, ou seja, que não represente ser ótimo para nenhum objetivo em específico. Para contornar este problema, o VEGA utiliza uma heurística para seleção de indivíduos não-dominados em cada iteração, ou ainda, uma seleção heurística com o cruzamento de espécies ao invés de uma seleção aleatória usada no algoritmo genético (COELLO, 1999b).

De acordo com Coello (1999b), a principal vantagem do VEGA é sua simplicidade, já que o agrupamento das sub-populações correspondem a uma média dos *fitness* associados a cada objetivo e, assim, o *fitness* resultante corresponde a uma combinação linear dos objetivos. A principal desvantagem deste método é a incapacidade de produzir soluções ótimas de Pareto para espaços de busca não-convexos.

MOGA

O *Multiobjective Optimization Genetic Algorithm* (MOGA) foi proposto por Fonseca e Fleming (1993). Esta técnica utiliza o conceito de não dominância para ordenar os indivíduos. Todos os indivíduos que são "não dominados" possuem ranque igual a 1, e os outros indivíduos receberão como valor de ranque 1 somado ao número de indivíduos que o dominam. Com isto todos os indivíduos são analisados quanto sua dominância dentre todos da população.

A designação do valor do *fitness* para cada indivíduo depende:

- da população atual através do ranqueamento;
- da interpolação do valor do *fitness*, do melhor ao pior indivíduo, de acordo com o ranque;
- da média do *fitness* dos indivíduos de mesmo ranque, sendo todos os indivíduos exemplificados através de uma mesma proporção, o que mantém constante o *fitness* total da população mantendo uma pressão apropriada da seleção.

Essa forma de designação do *fitness* pressiona a seleção que pode produzir uma convergência prematura. Para evitar esse problema, a distribuição da população sobre a região ótima de Pareto é feita pelo método de formação de nichos.

O MOGA é eficiente e relativamente fácil de implementar, embora seu desempenho dependa muito do tamanho do nicho σ_{share} . Entretanto, é importante citar que Fonseca e Fleming (1993) desenvolveram uma metodologia eficiente para a obtenção do valor σ_{share} para o MOGA.

NPGA

Horn e Nafpliotis (1993) propuseram um algoritmo genético em que a seleção é realizada por torneio e é baseada na dominância de Pareto para solução de problemas multiobjetivos denominada *Niched Pareto Genetic Algorithm* (NPGA) (HORN; NAFPLIOTIS, 1993). No início de cada processo de seleção, são escolhidos t_{dom} indivíduos da população, de forma aleatória, formando o conjunto para comparação. Duas soluções são aleatoriamente escolhidas da população e comparadas com os indivíduos desse conjunto com relação à função objetivo. Caso haja uma solução não dominada e outra dominada, a não dominada é selecionada. Caso contrário, é criado um contador de nichos para todos os indivíduos da população. O contador é calculado através do número de soluções na população dentro de uma distância σ_{share} do indivíduo. O indivíduo com o menor contador de nicho é selecionado dentre os dois iniciais.

A influência da adoção de múltiplos objetivos apresenta-se na análise de dominância. Devido à comparação ser realizada entre os t_{dom} indivíduos, o sucesso desse algoritmo depende

fortemente do valor de t_{dom} . Se esse valor não for corretamente considerado alguns ótimos de Pareto podem não ser achados. Assim, se o valor de t_{dom} for muito baixo pode resultar em poucas soluções não-dominadas dentro da população, se o valor for muito alto pode resultar em convergência prematura (DEB, 1999).

De acordo com (COELLO, 1999b), essa técnica é rápida e gera uma boa fronteira não dominante. Entretanto, requer um bom valor σ_{share} e um bom valor para o tamanho do conjunto comparação, dificultando seu uso na prática.

NSGA

Nondominated Sorting Genetic Algorithm (NSGA) foi proposto por Srinivas e Deb (DEB, 1999). Difere do algoritmo genético simplesmente pela forma de uso do operador de seleção. Os outros operadores de recombinação e mutação são mantidos da mesma forma que o usual.

Antes da seleção, a população é ranqueada pela classificação de não-dominância. Todos os indivíduos não-dominados são classificados em uma mesma categoria. São atribuídos valores de aptidão altos a todos os indivíduos não-dominados, de forma a garantir o mesmo potencial de reprodução a esses indivíduos.

Visando manter a diversidade da população, as soluções não-dominadas compartilham seus valores de aptidão segundo suas distâncias euclidianas. Finalmente, divide-se o valor da aptidão de cada indivíduo pelo contador de nichos que é proporcional ao número de vizinhos ao seu redor. Este procedimento proporciona a co-existência de pontos ótimos múltiplos na população (CASTRO, 2001b).

Então, esse grupo de indivíduos é ignorado e outro grupo de indivíduos não-dominados é considerado. Este processo continua até que todos os indivíduos da população tenham um valor de aptidão compartilhado.

O primeiro nível de soluções não dominadas possui as mais altas aptidões, assim sendo mais amplamente copiados encaminhando a busca mais próxima da fronteira de Pareto.

O NSGA permite lidar com qualquer número de objetivos, para problemas de minimização ou maximização. Entretanto o NSGA é muito sensível ao parâmetro de configuração σ_{share} , que tem grande influência no desempenho da busca (DEB, 1999).

SPEA

Zitzler e Thiele (1999) propuseram o *Strength Pareto Evolutionary Algorithm* (SPEA) que usa o conceito de não dominância. O SPEA possui um mecanismo inerente de elitismo em que mantém uma população externa que, a cada geração, guarda o conjunto de soluções

não dominadas. A cada solução do conjunto externo é atribuído um valor força, semelhante ao ranqueamento do MOGA, pois é proporcional ao número de soluções que a solução domina. Essa população externa participa das operações genéticas em que é combinada com a população atual. O *fitness* de cada indivíduo é obtido considerando o valor força de todos os indivíduos externos não-dominados pelos quais ele é dominado (COELLO, 2006).

Para manter a diversidade, um maior valor de *fitness* é atribuído às soluções não-dominadas que apresentem maior número de soluções dominadas na população. Por outro lado, é atribuído um alto valor de *fitness* para as soluções dominadas por um maior número de soluções, embora esse valor seja menor que os das soluções não-dominadas. Isso permite que a busca ocorra na direção das soluções não dominadas, simultaneamente mantém-se a diversidade dos indivíduos dominados e não dominados (DEB, 1999).

Segundo Castro (2001b), o SPEA destaca-se pela ausência de parâmetros de distância, como raio de nicho, compartilhamento e valores de *fitness* com relação às soluções do conjunto de Pareto. Entretanto, a eficiência depende do tamanho do conjunto externo de soluções não-dominadas Coello (2006).

Neste capítulo foi descrito como pode ser empregado um sistema híbrido. Todas as técnicas metaheurísticas empregadas neste trabalho foram apresentadas. Apresentou-se, também, um estudo das técnicas multiobjetivo a fim de selecionar uma delas a ser aplicada no problema de sequenciamento. No próximo capítulo tem-se a apresentação do problema de *scheduling* de operações de malha dutoviária real, bem como a abordagem que será empregada na solução deste problema.

4 REDE DE DUTOS

Este capítulo apresenta a rede de dutos que será modelada e o problema de *scheduling* tratado, como um todo. São apresentadas suas características operacionais e a abordagem de divisão proposta neste trabalho para a solução do problema de *scheduling*.

4.1 MODELO DA REDE

Uma rede de distribuição de produtos derivados do petróleo é composta por refinarias, portos e terminais, ou áreas, interligadas por oleodutos, que operam o transporte de produtos entre as áreas adjacentes.

Basicamente, a malha dutoviária brasileira é composta por duas redes principais, a rede de escuros e a rede de claros, diferenciadas pelo tráfego de diferentes tipos de produtos. A rede de escuros realiza o transporte de diversos tipos de petróleos e derivados pesados e a rede de claros transporta diversos tipos de derivados leves, de alto valor agregado, como por exemplo nafta, gasolina, diesel, querosene de aviação. A programação das operações envolvidas nestas duas redes é realizada de forma separada, visto que não há uma interdependência operacional significativa entre essas malhas.

Neste trabalho foi desenvolvida uma estrutura de otimização para o *scheduling* operacional de parte da rede de claros que compõe a malha dutoviária da indústria brasileira de petróleo. Esta malha realiza o transporte de claros das refinarias e portos aos centros de distribuição, localizados no estado de São Paulo, como mostrado na figura 10. O principal objetivo da estrutura de otimização proposta é dar suporte ao processo de decisão, que é particularmente complexo devido a existência de muitas áreas e dutos sujeitos a uma série de restrições operacionais.

A parte da rede estudada nesta tese é mostrada na figura 10, e é composta por: 9 nós, em que 3 deles são refinarias (nós N1, N3 e N7), 1 porto (nó N9), e 5 centros de distribuição (nós N2, N4, N5, N6 e N8). A rede inclui 15 dutos, em que cada duto possui um volume particular, por exemplo o duto 1 possui mais de 42000 m³. Os nós são conectados por vários dutos (por exemplo: os dutos 3, 4, e 5 conectam os nós 2 e 3). Entretanto, a lista de produtos que podem ser transportados pelos dutos é limitada (por exemplo: o duto 3 é tipicamente usado para

transportar gasolina e nafta). Um produto pode levar muitas horas para chegar ao seu destino. Uma batelada pode permanecer em um duto até que outra batelada a empurre. Os dutos 5,7 e 15 permitem a reversão de fluxo (mais detalhes serão vistos a seguir). Cada área possui tanques específicos para a armazenagem dos diversos produtos. A rede pode transportar mais de 10 diferentes produtos derivados do petróleo, este número varia de acordo com a demanda gerada pelo planejamento da companhia dentro do horizonte de tempo considerado.

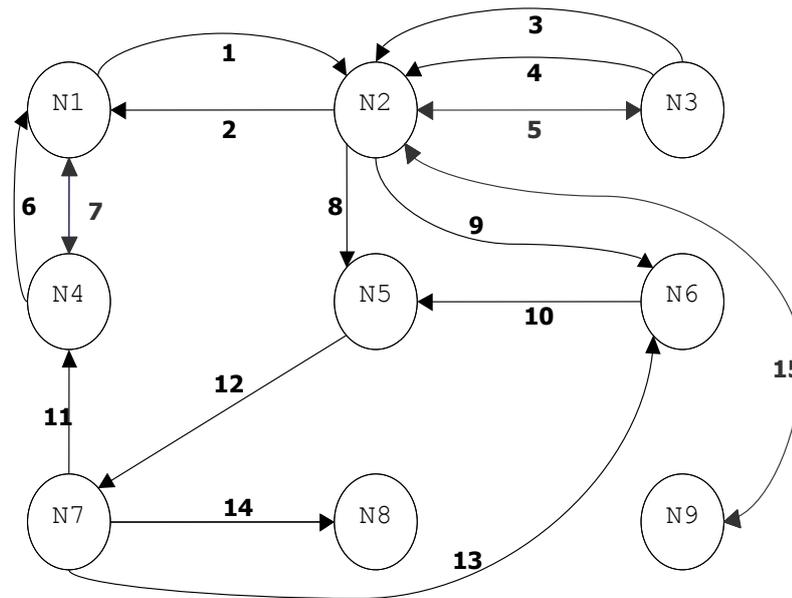


Figura 10 – Rede em estudo

As características operacionais das redes consideradas no modelo de otimização serão descritas a seguir:

1. As operações de transferência devem ocorrer num horizonte de tempo determinado (horizonte de *scheduling*).
2. Todo o planejamento da produção/consumo mensal para cada produto referente a cada área é previamente estabelecido pela companhia. Entretanto, durante o horizonte de *scheduling*, o volume total de cada produto deve ser dividido em volumes menores (bateladas), em função do bombeamento.
3. Os produtos podem ser bombeados a partir de diferentes origens. Em toda rede, para cada área é gerada uma lista de todas as bateladas a serem bombeadas e esta lista deve ser ordenada (sequência de bateladas) de acordo com uma prioridade determinada pela produção e demanda nos órgãos. A geração da sequência de bateladas é realizada de acordo com o horizonte de *scheduling*.

4. O volume da batelada de cada produto deve ser um valor múltiplo da capacidade de armazenagem da batelada na área de origem/destino ou igual ao volume de um duto específico dentro da sua rota de fluxo (veja o item 5). No caso de operações de reversão (veja o item 12), deve ser adicionada uma batelada com o mesmo volume do "segmento do duto" sob reversão.
5. Um rota de fluxo é um conjunto de dutos que conectam dois nós. Note que uma rota pode ter apenas um duto ou uma série de dutos conectando nós intermediários, porém sempre ligará a área de origem a área de destino. Devido às práticas operacionais, existem rotas preferenciais para cada produto. Essas rotas são determinadas como rotas típicas compostas por um produto, uma origem e um destino.
6. Uma rota pode ser usada para diferentes bateladas com diferentes produtos e vazões.
7. Para cada produto e/ou rota, existe um valor de vazão "típica" ou de bombeamento em cada área de origem, correspondendo ao número de bombas utilizadas para movimentar o produto no duto. Esta vazão pode ser afetada pelos produtos que serão bombeados no mesmo duto posteriormente a esta batelada.
8. Em algumas áreas, a batelada pode ser recebida em um tanque com uma vazão específica e simultaneamente ser bombeada deste tanque para um outro duto com uma vazão diferente. Este procedimento é caracterizado como operação pulmão.
9. Em cada área, uma batelada recebida de um duto pode ser bombeada para um tanque específico ou para outro duto (operação de passagem).
10. As áreas tem uma capacidade de armazenamento agregada por produto. Entretanto, existem limites superiores e inferiores para este armazenamento em cada área.
11. O nível de inventário do produto pode aumentar ou diminuir de acordo com o balanço de massa entre volume recebido, consumo local e a demanda de mercado. Este nível também é afetado pela produção local e o mercado de consumo de outras áreas.
12. Alguns dutos podem ter seu sentido de bombeamento revertido de acordo com as conveniências operacionais. São necessários alguns procedimentos operacionais específicos para a realização dessa operação. Após o envio das bateladas em um sentido, é necessário inserir uma batelada auxiliar de um produto com o mesmo volume do duto para que o bombeamento das bateladas desse sentido seja finalizado. A direção de bombeamento pode, então, ser revertida e o conjunto de bateladas podem ser enviadas no novo

sentido. A batelada auxiliar usada na operação de reversão deve ser especificada a priori no sequenciamento de bateladas.

13. Cada área possui características operacionais específicas. Por exemplo, em cada área há um número limitado de dutos e bombas, que determinam o número de bateladas que podem ser enviadas ou recebidas ao mesmo tempo em cada área. Este fato caracteriza as "restrições locais", que devem ser satisfeitas durante o horizonte de *scheduling*.
14. Devido às restrições de inventário em cada área, as bateladas devem ser enviadas e recebidas em um certo intervalo de tempo (ou janela de tempo). Para evitar ultrapassar os limites de inventário, o modelo determina janelas de tempo para as bateladas, fornecendo o intervalo de tempo em que as bateladas devem ser enviadas ou recebidas.
15. Todas as bateladas que estejam nos dutos antes do início do horizonte de tempo (bateladas em trânsito) devem ser consideradas para o atendimento da demanda e gerenciamento do inventário (inicialização do *scheduling*).
16. Os dutos sempre operam com seu volume completo e alguns deles apresentam um volume considerável. Portanto, eles são compartilhados por diferentes produtos durante seu bombeamento. Esses produtos podem ser enviados aos tanques ou a outro duto da rota de fluxo. Todo novo produto bombeado tem seu destino particular, assim como os produtos presentes nos dutos devem ser enviados aos seus destinos originais.
17. Alguns dutos podem se manter parados (sem bombeamento) durante todo o horizonte de tempo, entretanto há um custo operacional para manter a pressão nestes dutos.
18. A energia elétrica tem um custo diferenciado de acordo com a época e o período do dia. Normalmente, das 17:00 às 20:00 horas (horário de pico), o custo é maior que em qualquer outro período do dia. Portanto, o bombeamento deve ser periodicamente interrompido em algumas áreas durante o horizonte de *scheduling*.
19. Não há nada que separe os produtos bombeados sucessivamente. Consequentemente, há um volume de contaminação (interface) entre os produtos. Algumas interfaces não são operacionalmente recomendadas; então, um selo (uma pequena quantidade de produto) pode ser usada neste caso, mesmo que aumente o custo operacional. O tempo de bombeamento do selo é ignorado desde que o volume seja significativamente menor que qualquer batelada de produto demandada.

4.2 ABORDAGEM HÍBRIDA ADOTADA

A complexidade computacional para a determinação do *scheduling* operacional do cenário considerado torna-o uma tarefa difícil. Portanto, para a solução de um problema real considerou-se uma abordagem de decomposição baseada nos três elementos chave do *scheduling*: *assignment*, *sequencing* e *timing* (REKLAITIS, 1992). De acordo com essa abordagem, desenvolveu-se a estrutura ilustrada na figura 11. Nesta figura, os dados de entrada (cenários de consumo/produção, configuração de áreas/rotas, inventários, custo de energia elétrica e outros dados operacionais) são fornecidos inicialmente ao banco de dados. Estas informações descrevem o problema a ser resolvido. Os módulos de alocação dos recursos, sequenciamento e temporização, utilizam-se dessas informações para resolver o problema de *scheduling*.

O *scheduling* é solucionado através de um sistema híbrido, em que o problema é decomposto em três módulos, que são resolvidos separadamente. A alocação dos recursos é realizada de forma heurística; o sequenciamento utiliza otimização, segundo os estudos que serão apresentados nesta tese, modelos baseados na técnica metaheurística AG; e a temporização usa um modelo de programação linear inteira mista. Essa abordagem híbrida torna o problema *scheduling* real de grande complexidade computacional possível de ser resolvido em um curto tempo computacional.

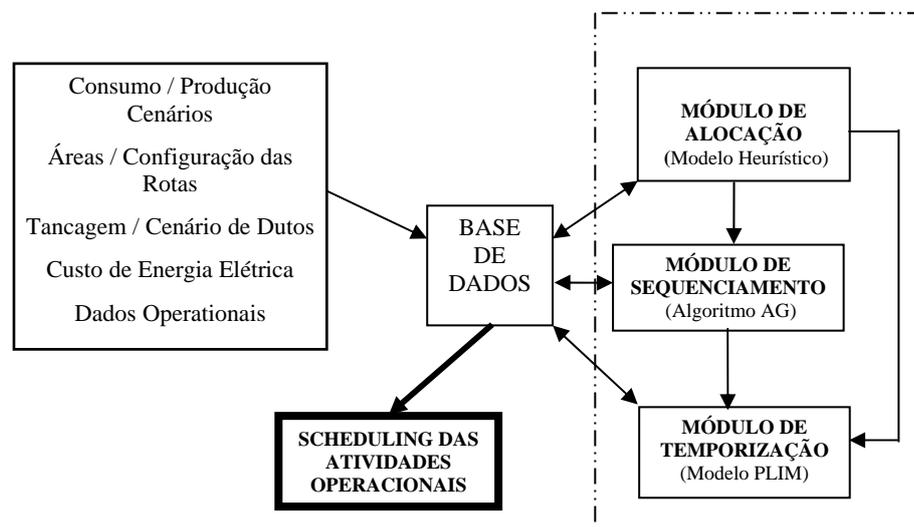


Figura 11 – Estrutura do *scheduling*

Alocação dos Recursos

Através dos dados do problema em estudo, este módulo gera os dados de entrada para os módulos seguintes (sequenciamento e temporização). São realizadas as análises de planos de produção, consumo, estoque, exportações e importações dos órgãos, e a análise das restrições

da rede, como características dos dutos, bombas e estoques nas áreas.

Para a formação das bateladas, primeiramente, são observadas as prioridades de envio e consumo de produtos, em todas as áreas. Então, são determinadas as áreas que irão fornecer e receber os produtos pela análise das curvas de estoque nos terminais de demanda e oferta.

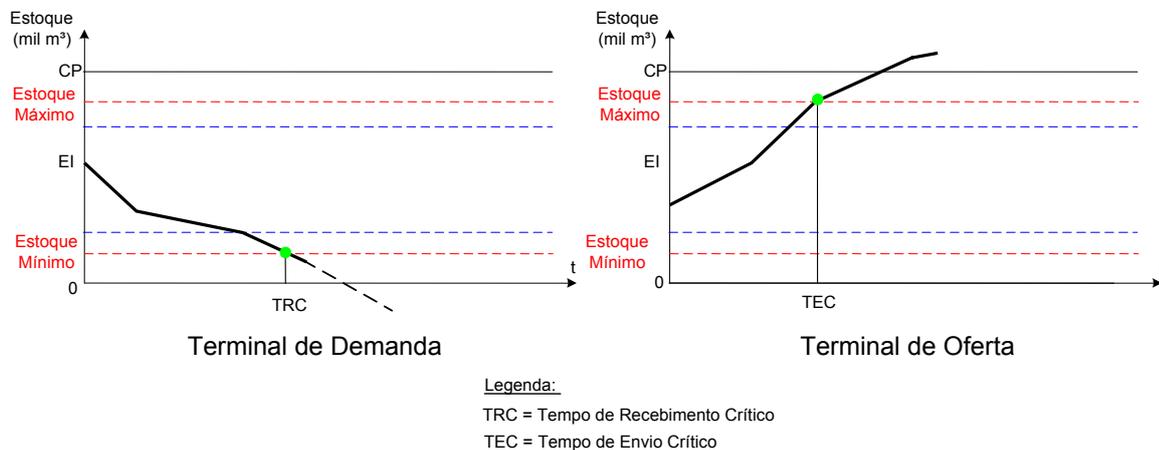


Figura 12 – Determinação dos tempos em que há a necessidade de envio e recebimento de produtos em uma área

Como pode ser observado na figura 12, dado o estoque inicial (EI), a taxa de consumo e o estoque mínimo, pode-se determinar o tempo de recebimento crítico (TRC) de cada produto em cada terminal de demanda. Da mesma forma, através do estoque inicial, da taxa de produção e do estoque máximo tem-se o tempo de envio crítico (TEC) de cada produto em cada terminal de oferta. Essa determinação dos tempos de recebimento e envio críticos permite gerar as bateladas, estabelecendo uma origem de envio e um destino de recebimento de cada produto, atribuindo a ela uma rota de fluxo. São adotadas vazões e volumes típicos das bateladas que são pré-determinados em função do produto e da rota de fluxo adotada.

Exemplificando: Ao considerar um produto 1, faz-se a análise do terminal de oferta (origem), que deve enviar primeiro esse produto (com menor TEC), assim observa-se o terminal de demanda (destino) com maior necessidade do produto (com menor TRC). Associa-se, então, uma rota de fluxo, pré-cadastrada, que realize a movimentação deste produto da origem ao destino. Assim tem-se que o produto 1, irá da origem ao destino, por uma determinada rota, e para esta, tem-se uma vazão e um volume pré-determinado.

Tendo gerado as bateladas, a construção das janelas de tempo de cada batelada deriva dos aspectos dos estoques da rede considerando os dados de produção e consumo, e são geradas da forma como mostra a figura 13.

Os terminais de demanda, através do estoque inicial, taxa de consumo e estoque máximo, determinam o tempo de recebimento disponível (TRD), quando possibilita receber um volume igual ao volume da batelada, bem como o tempo de recebimento crítico (TRC), quando

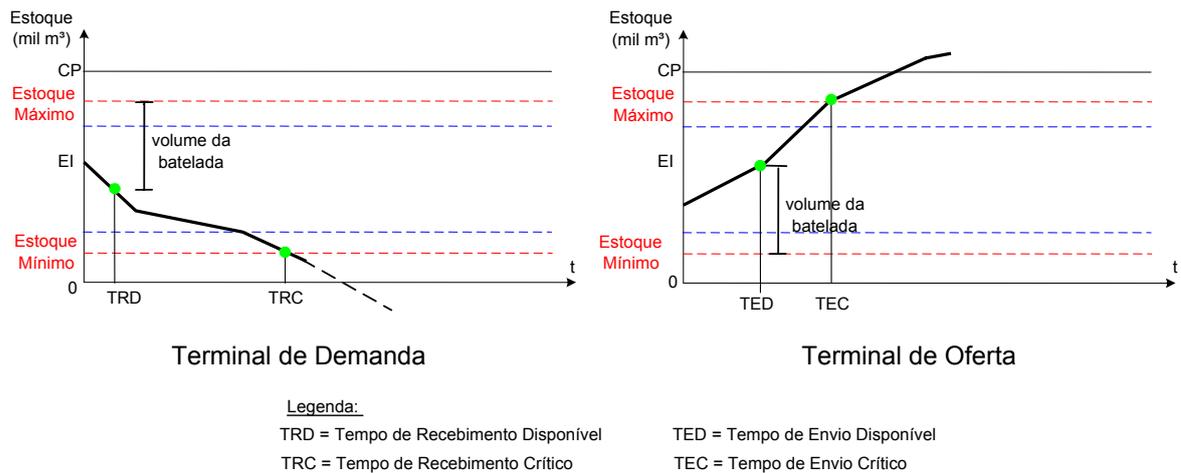


Figura 13 – Determinação das janelas de tempo para os terminais de demanda e oferta

o volume de estocagem chega no estoque mínimo e necessita do recebimento de produto para garantir o atendimento à demanda.

Os terminais de oferta, determinam o tempo de envio disponível (TED) através do estoque inicial, taxa de produção e estoque mínimo, quando houver no estoque a quantidade igual ao volume da batelada. O tempo de envio crítico (TEC) se dá quando o estoque do terminal alcança o volume máximo.

Dessa forma, todas as bateladas necessárias para o atendimento à demanda de produtos são geradas. Essas bateladas fornecem a área de origem e destino, o produto, o volume, a vazão e as janelas de tempo, como pode ser visto na tabela 5. Os planos de produção e as estimativas de demanda formam o principal critério de escolha entre os órgãos na programação de bateladas, e o monitoramento da evolução dos estoques é determinante para um bom seqüenciamento de atividades. Nesta tabela, por exemplo, estabelece que a batelada 4 irá partir do nó N1 ao nó de destino N9, transportando o produto 6 com o volume de 25.000 m^3 , através da rota 7, a uma vazão de $850 \text{ m}^3/\text{h}$. Esta batelada possui uma janela de tempo de 0 a 2000 horas na origem, ou seja, intervalo de tempo em que a batelada deve sair de N1, e de 0 a 104 horas no destino, sendo o intervalo de tempo no qual a batelada deve chegar em N9.

Tabela 5 – Exemplo de dados de saída do módulo de alocação

Bat.	Nó Orig.	Nó Dest.	Núm. Prod.	Vol. (m^3)	Núm. Rota	Vazão Típica (m^3/h)	TED (h)	TEC (h)	TRD (h)	TRC (h)
1	N7	N4	5	15000	23	450	0	92	0	91
2	N3	N8	2	27000	18	1000	0	164	0	104
3	N7	N5	5	10000	26	350	0	175	0	217
4	N1	N9	6	25000	7	850	0	2000	0	104
5	N1	N4	3	5000	8	190	0	2000	0	134
6	N7	N4	6	18000	23	450	0	189	0	762
7	N4	N1	6	18000	21	330	0	189	0	762
8	N1	N4	3	5000	8	190	0	2000	0	198
9	N7	N4	6	18000	23	450	0	338	0	427
10	N3	N5	1	13000	19	650	0	191	12	59

Seqüenciamento

O seqüenciamento tem a finalidade de fornecer a melhor ordenação de bateladas a serem inseridas na rede que implique em um melhor resultado final do problema de *scheduling*.

O número de interfaces e a otimização do tamanho das bateladas, a configuração e seqüenciamento são fatores importantes na redução do custo operacional do transporte em dutos. Os operadores de dutos, para a escolha da configuração e seqüência das bateladas, optam por algumas ações:

- Evitar a interface de produtos incompatíveis.
- Agrupar seqüencialmente produtos similares.
- Evitar a interface de dois produtos com significativa diferença de viscosidade.

Outros fatores também são importantes para o *scheduling* operacional e podem influenciar o seqüenciamento das bateladas: a rota da batelada e a localização com relação a toda rede, tempo de chegada apropriado e a vazão de envio esperada. Estes fatores são importantes para a troca de válvulas e bombas entre os produtos em cada nó de chegada. Neste trabalho, o problema do seqüenciamento de bateladas consiste na determinação da ordem das bateladas a serem bombeadas em cada terminal de envio a fim de minimizar o custo operacional total.

O seqüenciamento é um processo de geração de uma ou mais ordenações de bateladas de produtos derivados do petróleo visando uma diminuição no tempo total da programação, uma melhora da taxa de utilização dos dutos, priorizando o atendimento à demanda de produtos. Devido a esses vários fatores, o seqüenciamento, nesta tese, é um problema de otimização multiobjetivo.

A solução do problema de seqüenciamento é o foco principal da presente tese e um modelo baseado em metaheurísticas será proposto nos capítulos seguintes.

Temporização

Este módulo é objeto de estudo da tese de doutorado desenvolvida por Luiz Carlos Felizari (FELIZARI, 2009).

O módulo de temporização aloca as atividades, que neste caso são as operações de envio, recebimento das bateladas, no tempo. Os dados gerados são usados por um modelo de programação matemática, a programação linear inteira mista (PLIM) (ou MILP-*Mixed Integer Linear Programming*) com abordagem de tempo contínuo, o qual determina o *scheduling* de curto-prazo com base em restrições operacionais, como por exemplo, limites de tancagem, volume dos dutos, vazões de bombeio e custo horossazonal de demanda de energia elétrica.

Dada uma lista de bateladas, um conjunto de janelas de tempo, e o estado inicial da rede, o módulo de temporização deve determinar o *scheduling* operacional durante o horizonte de programação. A função objetivo do modelo de temporização é dada pela equação 1. O objetivo é minimizar o tempo de bombeamento e recebimento das operações (*makespan*), o tempo de parada das bateladas e o atraso ou adiantamento de entregas ou recebimentos em cada nó, de acordo com as janelas de tempo. Algumas variáveis foram criadas para a determinação exata do início ($ib_{b,n,n',d}$) e término ($fb_{b,n,n',d}$) do bombeamento das bateladas ($b \in B$) em cada nó n através da rota ($n \neq n' \in N$) por um duto em específico ($d \in D$, onde d conecta n e n'). De forma similar, outras variáveis contínuas determinam o tempo em que o nó n' começa a receber ($ir_{b,n,n',d}$) e termina o recebimento ($fr_{b,n,n',d}$) de um produto. Ainda, manter a batelada parada dentro dos dutos ($fstop_{b,n,n',d}$; $istop_{b,n,n',d}$) é uma condição válida porém influencia o custo operacional. A formulação foi amplamente estudada e as variáveis binárias foram usadas para assegurar a sazonalidade das condições impostas sobre a energia elétrica. Assim, o modelo evita o bombeamento durante os períodos de alto custo de energia.

Minimize :

$$\begin{aligned}
& \sum_{b \in B} \sum_{n \in N} \sum_{n' \in N, n \neq n'} \sum_{d \in D} (ib_{b,n,n',d} + fb_{b,n,n',d} + ir_{b,n,n',d} + fr_{b,n,n',d}) * ktb + \\
& \sum_{b \in B} \sum_{n \in N} \sum_{n' \in N, n \neq n'} \sum_{d \in D} (fstop_{b,n,n',d} + istop_{b,n,n',d} + fr_{b,n,n',d}) * ktstop + \\
& \sum_{b \in B} \sum_{n \in N} \sum_{n' \in N, n \neq n'} \sum_{d \in D} (ad_{b,n,n',d} + dd_{b,n,n',d} + fr_{b,n,n',d}) * ktd + \\
& \sum_{b \in B} \sum_{n \in N} \sum_{n' \in N, n \neq n'} \sum_{d \in D} (ao_{b,n,n',d} + do_{b,n,n',d} + fr_{b,n,n',d}) * kto
\end{aligned} \tag{1}$$

Foram criadas restrições específicas para atender aos limites de inventário e às janelas de tempo das bateladas. Embora, o modelo PLIM realize o *scheduling* operacional em cada nó ao invés de satisfazer os limites de tempo. Porém, algumas violações das janelas de tempo podem ser aceitáveis tanto na origem ($ao_{b,n,n',d}$; $do_{b,n,n',d}$) quanto no destino ($ad_{b,n,n',d}$; $dd_{b,n,n',d}$), mesmo que isso seja indesejado. À função objetivo são atribuídos pesos pelos custos operacionais (ktb , $ktstop$, ktd , kto), o que permite estabelecer prioridades dentro dos seguintes aspectos: inventário, movimento das bateladas, operações de bombeamento e recebimento, e tempo de parada. Cada nó possui características operacionais particulares, que deve ser descrito no modelo matemático. Por exemplo, bateladas podem ser bombeadas de N7 pelos dutos 11, 13, e 14 (ver 10). Há um número limitado de bombas neste nó, portanto só é permitido o envio de apenas uma batelada por vez. Então, em uma situação hipotética em que haja várias bateladas a serem enviadas por N7, o modelo deve gerenciar os tempos de início e término do bombeamento para

satisfazer essa característica local. Outro ítem, os dutos 5, 7, e 15 podem ter seu fluxo revertido, de acordo com as necessidades operacionais. Um conjunto de restrições específicas foi criado para gerenciar essa condição operacional. A figura 14 descreve esta situação para o duto 5: as bateladas 5 e 6 devem ser transportadas de N3 para N2 e as bateladas 4 e 3 de N2 para N3. A reversão é realizada pela inserção de uma batelada auxiliar (batelada 1) com o volume do duto 5 para que empurre totalmente a batelada 6 até N2. Assim, a batelada 4 empurra a batelada 1 retornando para N3. A literatura de *scheduling* de dutos (CAFARO; CERDÁ, 2004), mostra que esta operação gera complexidade nos modelos de *scheduling*.

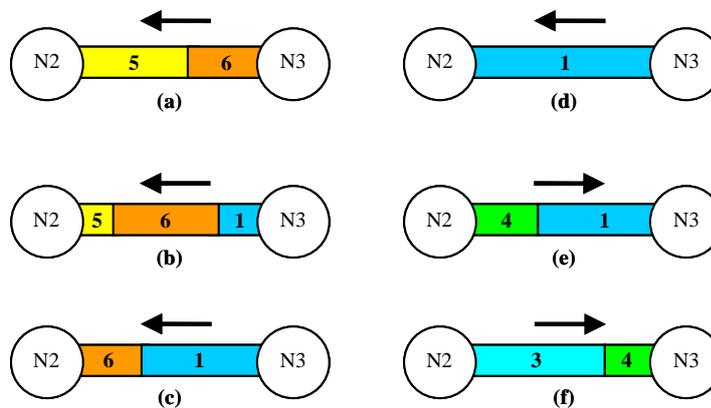


Figura 14 – Operação de reversão

A vazão típica pode variar de duto a duto, sendo necessário o uso de tancagem intermediária. Por exemplo, uma batelada de nafta deve ser bombeada de N7 ao seu destino final N1. Neste caso, o produto pode passar por um nó intermediário N4. A operação envolve os dutos 11 e 7, respectivamente. Para o duto 11, a vazão média é de $450 \text{ m}^3/\text{h}$ de N7 a N4. Para o duto 7, a vazão média é de $190 \text{ m}^3/\text{h}$ de N4 a N1. Outra alternativa seria o produto ser diretamente bombeado de N7 a N1 com vazão de $190 \text{ m}^3/\text{h}$. Neste caso, as bombas do nó N4 não são usadas. Entretanto, é recomendado por operações práticas que a nafta seja bombeada de N7 a N4 com $450 \text{ m}^3/\text{h}$, armazenado em um tanque em N4 e simultaneamente bombeado de N4 a N1 com $190 \text{ m}^3/\text{h}$. Esta situação, denominada operação pulmão, é mostrada na figura 15. Para modelar tal situação foi criado um conjunto de restrições específicas para estas condições operacionais, introduzindo mais complexidade no modelo.

A existência de operações de reversão e de pulmão numa carteira de bateladas implica em um aumento do número de bateladas. Pode-se observar, a seguir, a forma de inserção de novas bateladas, que é realizada após o seqüenciamento:

- A existência de reversão somente é possível de ser observada após o seqüenciamento das bateladas, em que os dutos que permitem reversão são analisados e verifica-se a existência de rotas consecutivas com o fluxo trocado. Caso haja a reversão, uma batelada, com o

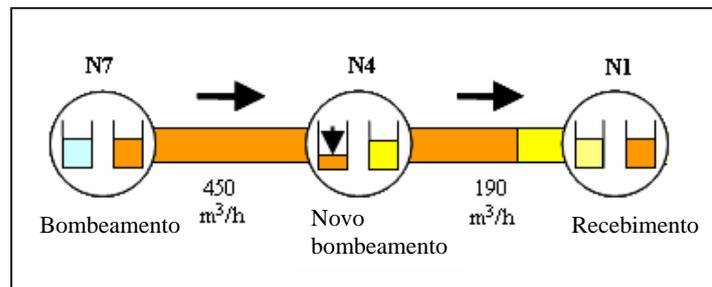


Figura 15 – Operação de pulmão

volume igual ao volume do duto, de um produto qualquer é inserida entre as bateladas com fluxo revertido.

- Na operação pulmão ocorre uma divisão da batelada existente em mais bateladas. Assim, após o sequenciamento ser realizado é possível verificar quais bateladas que passarão por determinadas rotas com determinados produtos realizarão a operação pulmão, e essa batelada é dividida. No exemplo mencionado anteriormente, dada uma batelada de nafta bombeada de N7 a N1 em que ocorrerá pulmão em N4, ela será dividida em duas bateladas: uma que irá de N7 a N4 com vazão de $450 \text{ m}^3/\text{h}$ e outra, posterior, que irá de N4 a N1 com vazão de $190 \text{ m}^3/\text{h}$.

Para uma descrição mais detalhada da formulação matemática do modelo PLIM usada no módulo de temporização consultar Felizari (2009).

Através da abordagem híbrida foi possível solucionar o *scheduling* operacional da rede, englobando as muitas características do cenário através de um baixo tempo computacional.

O modelo híbrido fornece as seguintes características do *scheduling* operacional:

1. Um conjunto de bateladas (volume e produtos) com a rota associada.
2. A seqüência com que essas bateladas serão inseridas na rede.
3. O tempo de início e término do bombeio das bateladas em cada nó.
4. O tempo inicial e final de recebimento das bateladas em cada nó.
5. O tempo de inicial e final das bateladas que permanecem paradas em cada duto.
6. Determinação dos movimentos das bateladas por todos os dutos pertencentes as rotas das bateladas.
7. As violações das janelas de tempo que ocorrem durante as operações de bombeamento e recebimento, permitindo um controle seguro dos níveis de inventário em cada nó.

Neste capítulo, pode-se observar as características de uma rede real de dutos. Assim como a abordagem de divisão do problema de *scheduling* que é adotada neste trabalho para que seja possível a resolução deste problema através de uma ferramenta computacional. No próximo capítulo tem-se a descrição do módulo de sequenciamento de bateladas, os estudos e testes para a modelagem do problema.

5 MODELOS PROPOSTOS

Este capítulo apresenta a abordagem do problema de sequenciamento das bateladas *scheduling* de operações, mostrando os estudos realizados para a solução deste problema. São mostrados os modelos baseados em metaheurísticas utilizados no problema de sequenciamento, assim como os resultados e análises das diferentes funções objetivos empregadas na otimização. Poderá ser observada, através de exemplos, a influência da otimização do sequenciamento de bateladas no *scheduling* das operações de uma malha dutoviária. Através das soluções alcançadas, adota-se uma função objetivo que corresponda às melhores soluções, além da escolha de uma metaheurística mais favorável ao problema do sequenciamento de bateladas.

5.1 SEQÜENCIAMENTO

Esta parte do trabalho traz uma análise comparativa de desempenho dos modelos baseados em metaheurística, anteriormente citados, para o sequenciamento de bateladas de produtos derivados do petróleo na rede de dutos descrita no capítulo 4.

Os modelos implementados na resolução do sequenciamento foram baseados nas seguintes metaheurísticas: Algoritmo Genético (em regime, geracional e híbrido), Busca Tabu, *Greedy Randomized Adaptive Search Procedures*, *Simulated Annealing*, Colônia de Formigas, *Scatter Search* e *Variable Neighbour Search*. Os modelos foram executados na linguagem C em um computador Pentium III com processador de 1GHz e 256Mbytes de memória RAM.

Os valores dos resultados apresentados neste trabalho estão representados por uma média aritmética simples dos valores obtidos através de uma série de processamentos. Estes valores foram analisados com relação à dispersão, a média adotada representa os dados analisados segundo a análise por Diagrama de Caixas. Um diagrama em caixas (ou *Boxplot*) é um gráfico de dados que consiste em uma reta que se prolonga do menor ao maior valor, e um retângulo com retas traçadas no primeiro quartil, na mediana e terceiro quartil (TRIOLA, 1999). Como as medianas revelam uma tendência central, ao passo que os quartis indicam a dispersão dos dados, os diagramas em caixa têm a vantagem de não serem tão sensíveis a valores extremos como outras medidas baseadas na média e no desvio padrão. Com a construção de um diagrama em caixas torna-se fácil identificar os *outliers* (ou valores extremos), que são valores raros, no

sentido que são muito afastados da maioria dos dados, valores estes que foram rejeitados neste trabalho.

5.1.1 Sequenciamento de Bateladas através de Modelos baseados em Metaheurísticas

Neste trabalho, um cenário representa os dados que caracterizam as operações a serem realizadas dentro do período de um mês (horizonte de planejamento), previamente gerada por um sistema de planejamento. Serão apresentadas as soluções geradas na utilização de diferentes cenários, cenário 1, 2 e 3, que apresentam dados reais. Nas tabelas 6, 7 e 8, a seguir, pode-se verificar as demandas requeridas pelos nós para os cenários 1, 2 e 3, respectivamente. Na tabela 6, por exemplo, a demanda do produto 3 é de 296 mil m^3 em N1, 27 mil m^3 em N3, 58 mil m^3 em N4, 60 mil m^3 em N5 e 97 mil m^3 em N7.

Tabela 6 – Tabela de demanda de produtos para o cenário 1, em mil m^3

	N1	N2	N3	N4	N5	N6	N7	N8	N9
Produto 1	20	154							
Produto 2								215	
Produto 3	296		27	58	60		97		
Produto 4									100
Produto 5	30		39	68	52		86		
Produto 6	694		80						200
Produto 7	221		26						9

Tabela 7 – Tabela de demanda de produtos para o cenário 2, em mil m^3

	N1	N2	N3	N4	N5	N6	N7	N8	N9
Produto 1	21				161				
Produto 2								243	
Produto 3	358		31	68	65		173		
Produto 4									57
Produto 5	56		38	77	49		104		
Produto 6	358		200						185
Produto 7	250		14						
Produto 8				68					
Produto 9				3					

Tabela 8 – Tabela de demanda de produtos para o cenário 3, em mil m^3

	N1	N2	N3	N4	N5	N6	N7	N8	N9
Produto 1	35	150							
Produto 2								227	
Produto 3	295		34	68	62		87		
Produto 4									90
Produto 5	58		36	77	46		88		
Produto 6	688		85						150
Produto 7				5					

Esses valores de demanda são utilizadas, no módulo de alocação, para gerar as carteiras de bateladas. Os dados de oferta de produtos são baseados nos planos de produção das refinarias, considerando uma produção linear. As tabelas 9, 10 e 11 mostram um resumo da produção para os cenários 1, 2 e 3, respectivamente. Por exemplo (tabela 9), o nó 7 produz 90 mil m^3 o que representa 3 mil m^3 por dia de produto 3. Então, todas as refinarias possuem os dados de produção com relação a todos os produtos, assim como os portos possuem os dados de importação de produtos.

Tabela 9 – Tabela de produção dos derivados para o cenário 1, em mil m^3

	N1	N2	N3	N4	N5	N6	N7	N8	N9
Produto 1	35		125						
Produto 2	80		122						
Produto 3	294		80				90		
Produto 4			121						
Produto 5	100		45				220		
Produto 6	623		219				115		
Produto 7	225								

Tabela 10 – Tabela de produção dos derivados para o cenário 2, em mil m^3

	N1	N2	N3	N4	N5	N6	N7	N8	N9
Produto 1	25		160						
Produto 2	72		112				47		50
Produto 3	330		120				219		
Produto 4			60						
Produto 5	120		50				230		
Produto 6	397		213				100		
Produto 7	241		39				18		
Produto 8							5		
Produto 9	3								

Tabela 11 – Tabela de produção dos derivados para o cenário 3, em mil m^3

	N1	N2	N3	N4	N5	N6	N7	N8	N9
Produto 1	30		145						25
Produto 2	60		116						
Produto 3	295		120				180		
Produto 4			95						
Produto 5	120		45				160		
Produto 6	653		227				87		
Produto 7									

Após a execução do módulo de temporização, é possível visualizar parte da carteira de bateladas geradas para o cenário 1 na tabela 12.

Tabela 12 – Algumas bateladas pertencentes à carteira de bateladas do cenário 1

Bat.	orig.	dest.	prod.	volume	rota	vazão	TED	TEC	TRD	TRC
1	N7	N4	5	15000	23	450	0	92	0	91
2	N3	N8	2	27000	18	1000	0	164	0	104
3	N7	N5	5	10000	26	350	0	175	0	217
4	N1	N9	6	25000	7	850	0	2000	0	104
5	N1	N4	3	5000	8	190	0	2000	0	134
6	N7	N4	6	18000	23	450	0	189	0	762
7	N4	N1	6	18000	21	330	0	189	0	762
8	N1	N4	3	5000	8	190	0	2000	0	198
9	N7	N4	6	18000	23	450	0	338	0	427
10	N3	N5	1	13000	19	650	0	191	12	59

Na tabela 12 a primeira coluna (Bat.) fornece o número da ordem da batelada, seguida pelos nós de origem e destino da batelada. A quarta coluna descreve o produto a ser enviado (ex.: a primeira batelada é formada pelo produto 5). Posteriormente, tem-se o volume da batelada, em m^3 , o número da rota pré-cadastrada, e a vazão de envio, em m^3/h . As quatro últimas colunas fornecem as janelas de tempo da origem (TED e TEC) e do destino (TRD e TRC) para cada batelada, em horas.

Portanto, por exemplo, a batelada 1, tem sua origem no nó 7 e destino no nó 4. É composta pelo produto 5 e seu volume será de $15000 m^3$. Deverá ser enviada, pela rota 23, com uma vazão de $450 m^3/h$. Seu início de envio deve respeitar o intervalo de 0 a 92 horas e o recebimento deverá ocorrer entre 0 e 91 horas.

A carteira de bateladas completa do cenário 1 é composta por 76 bateladas, do cenário 2 é composta por 80 bateladas e do cenário 3 por 48 bateladas.

Comparações dos Modelos baseados em Metaheurísticas

As metaheurísticas são amplamente empregadas em problemas de otimização combinatorial, resultando em boas soluções (GENDREAU; POTVIN, 2005). Uma das características dessas técnicas é a possibilidade de suas aplicações em problemas reais com complicadas restrições. Entretanto é necessário um amplo conhecimento do problema para uma implementação de sucesso.

Um primeiro passo é a representação de uma solução através de uma sequência de símbolos. Devido às características do problema de seqüenciamento, uma codificação simples representa a ordem em que as bateladas devem ser inseridas na rede. Assim, cada solução é formada pela permutação das bateladas, codificada por uma sequência de valores inteiros. Cada valor inteiro corresponde ao número da batelada originada por uma lista inicial (ex. a coluna Bat. da tabela 12), gerada pelo módulo de alocação. A posição corresponde à ordem da

seqüência. A figura 16 ilustra esta codificação em que a terceira batelada a ser inserida na rede será a batelada 5.

1	3	5	4	8	2	6	10	60	76	85	80
---	---	---	---	---	---	---	----	-------	----	----	----	----

Figura 16 – Representação de uma solução

As nomenclaturas para as técnicas estudadas são dadas na tabela 13 a seguir.

Tabela 13 – Nomenclatura das metaheurísticas implementadas

AG1	Algoritmo Genético "em regime"
AG2	Algoritmo Genético híbrido ("em regime" e "geracional")
AG3	Algoritmo Genético "geracional"
BT	Busca Tabu
GRASP	<i>Greedy Randomized Adaptive Search Procedures</i>
SA	<i>Simulated Annealing</i>
ACO	Colônia de Formiga (<i>Ant Colony Optimization</i>)
SS	<i>Scatter Search</i>
VNS	<i>Variable Neighbour Search</i>

Na tabela 14, a seguir, serão apresentados os parâmetros utilizados no algoritmo dos modelos baseados em metaheurísticas:

Tabela 14 – Parâmetros dos algoritmos

Metaheurística	Parâmetros utilizados
AG1	Tamanho da população → 50 indivíduos (gerados aleatoriamente) Número de iterações (gerações) → 100.000
AG2	Tamanho da população → 50 indivíduos (gerados aleatoriamente) Número de iterações (gerações) → 100.000 Número de gerações para a criação de uma nova população → 10.000 Número de indivíduos que permanecerão na nova população (elitismo) → 5 indivíduos
AG3	Tamanho da população → 50 indivíduos (gerados aleatoriamente) Número de iterações (gerações) → 100.000 Número de indivíduos que permanecerão na nova população (elitismo) → 5 indivíduos
Busca Tabu	Controle do número de iterações → 1.000 Número de tentativas de busca de um melhor vizinho → 1.000 Tamanho da lista tabu → 50
GRASP	Número de repetições do algoritmo → 10 Número de semi-soluções dentre as quais será sorteada apenas uma para compor a solução → 5 Número de elementos que sofrerão permutações para a melhora da solução → 2
<i>Simulated Annealing</i>	Temperatura inicial → 10.300 Temperatura final de encerramento do algoritmo → 10^{-300} Índice de decremento da temperatura(alpha) → 0.999
<i>Scatter Search</i>	Número de soluções iniciais geradas aleatoriamente pelo algoritmo → 100 Tamanho da lista de referência P → 10 Tamanho da lista de referência 1 → 3 Tamanho da lista de referência 2 → 2 Número de iterações que serão executadas → 5.000
VNS	Solução inicial gerada aleatoriamente Número máximo de vizinhos, critério de parada → 15 Número de evoluções que o vizinho sofrerá → 60
Colônia de Formigas	Número de formigas → 20 (geradas aleatoriamente) Fator de redução do feromônio, evaporação → 0.99 Número de iterações → 10.000

Os parâmetros da tabela 14 foram gerados após inúmeros processamentos e observações, para que fosse possível realizar uma análise dos resultados gerados pelos diferentes modelos.

Critério de Parada

Os critérios de parada utilizados foram comparativos entre valores da função objetivo (F.O.) e número de iterações sem melhoras da F.O. (após 1.500 iterações sem nenhuma melhora da F.O., o algoritmo pára). Não foi possível serem feitas comparações entre os modelos através de um mesmo número de iterações, devido a diferenças de implementações, já que algumas técnicas envolvem "loops"dentro de "loops". Logo, em modelos com "loops"internos a outros "loops"uma iteração pode demorar muito mais que uma iteração de outros modelos que não possuam "loops"internos. Assim, uma iteração deste primeiro modelo pode apresentar melhoras

mais significativas da solução do que uma iteração do outro modelo.

Número de iterações

Não foi possível realizar uma comparação por número de iterações devido à grande diferença das metaheurísticas. No AG, 100.000 iterações correspondem a 100.000 soluções (indivíduos) factíveis geradas e diferentes das soluções que compõem a população atual. No BT podem-se achar mais soluções dentro de uma iteração, pois dentro de cada uma das 1.000 iterações, executa-se 1.000 tentativas de geração de novas soluções. Essa geração de novas soluções implica em uma busca, através da solução atual, até que se obtenha uma nova solução que não faça parte da lista tabu e seja melhor que a melhor solução atual. No entanto, ao fim de uma iteração, sobrarão apenas uma nova solução, a melhor. No GRASP, as 15.000 iterações são o número de tentativas de melhoria efetuadas dentro da busca local. As 10 repetições representam o número de vezes que o algoritmo será executado, isto é, o número de gerações de novas soluções independentes. No final, tem-se a melhor das soluções analisadas.

Resultados e Análise

Os resultados obtidos para o problema de sequenciamento pelas metaheurísticas descritas anteriormente, estão apresentados nas figuras a seguir. Na formulação, considerou-se a maximização da função objetivo, cujo valor representa uma combinação linear dos valores da janela de tempo e dos tamanhos destas janelas, como mostra a equação a seguir:

$$\sum_{i=1}^{nbat} \alpha_1 * (ordTRC_i) + \alpha_2 * ord(TRC_i - TRD_i) + \alpha_3 * (ordTED_i) + \alpha_4 * ord(TEC_i - TED_i) \quad (2)$$

O $nbat$ é o número total de bateladas, o TRC, TRD, TEC e TED são os elementos da janela de tempo que representam o tempo de recebimento crítico, o tempo de recebimento disponível, o tempo de envio crítico e o tempo de envio disponível, respectivamente. O ord significa ordem, por exemplo, o $ordTRC_i$ é a colocação de TRC_i em uma lista ordenada do pior ao melhor valor encontrado de TRC. Os valores de α_i foram adotados após uma série de testes, em que foram observados melhores resultados para o *scheduling*, sendo $\alpha_1 = 15$ $\alpha_2 = 5$ $\alpha_3 = 10$ $\alpha_4 = 1$.

As figuras 17, a e b, mostram o tempo computacional necessário por todos os métodos para alcançar um valor alto e pré-fixado da função objetivo, neste caso 150.670.000 para o cenário 1 e 156.450.000 para o cenário 2. Estes valores, adotados após uma série de processamentos, permitem observar o comportamento dos algoritmos para alcançar um valor próximo do ótimo.

Observou-se que, em função do tempo de processamento para alcançar um valor esti-

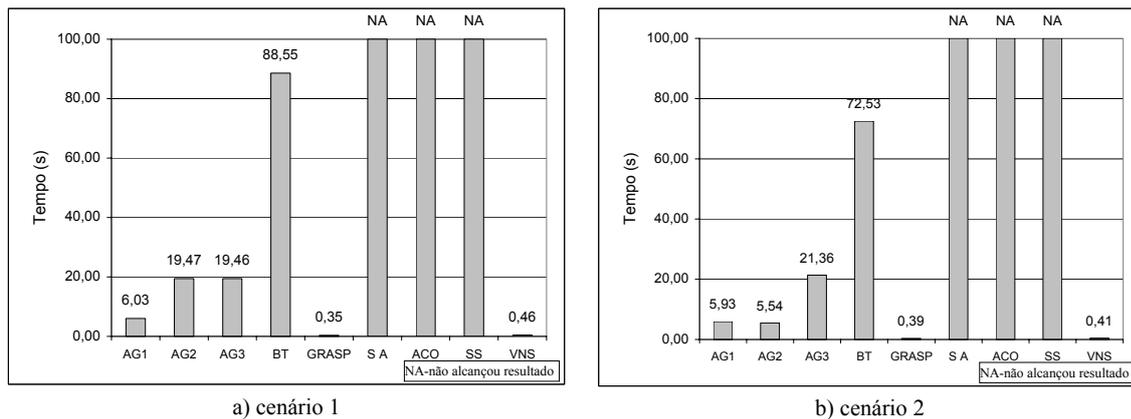


Figura 17 – Tempo computacional médio obtido pelo processamento das diferentes técnicas metaheurísticas

pulado da função objetivo, as modelos que obtiveram os melhores resultados foram baseados em AG, GRASP e VNS. Assim como, foi observado que as modelos baseados em BT, SA, ACO e SS apresentaram resultados muito inferiores aos outros métodos, isto é, um tempo muito elevado de processamento. Os modelos que apresentam NA (não alcançaram resultados) foram limitadas pelo tempo de processamento de 3 horas sem que o algoritmo alcançasse ou se aproximasse bastante do valor da função objetivo estipulada.

Observou-se que as três diferentes implementações do algoritmo genético obtiveram bons resultados. Assim, analisou-se o conjunto de soluções obtidos por diversos processamentos. Embora o AG1 tenha apresentado bons resultados segundo o tempo de processamento, para o cenário 1, 30% dos resultados obtidos foram descartados por serem muito dispersos do conjunto de soluções, da mesma forma 50% dos resultados obtidos pelo AG3, também foram descartados pelo mesmo motivo. O AG2, após muitos processamentos, não apresentou uma dispersão das soluções, portanto considerou-se o AG2 a melhor forma de implementação do algoritmo genético.

As figuras a seguir (18 e 19) mostram o comportamento quando considera-se o critério de parada a diferença do valor das funções objetivo ($f_{o_i} - f_{o_{i-1}}$) constante para 1500 iterações i . Na figura 18(a) têm-se os valores resultantes da função objetivo, e na figura 18(b) tem-se o tempo requerido pelos modelos baseados nas diferentes metaheurísticas, ambos para o cenário 1. As figuras 19(a) e 19(b) apresentam os resultados obtidos da função objetivo e tempo, respectivamente, para o cenário 2.

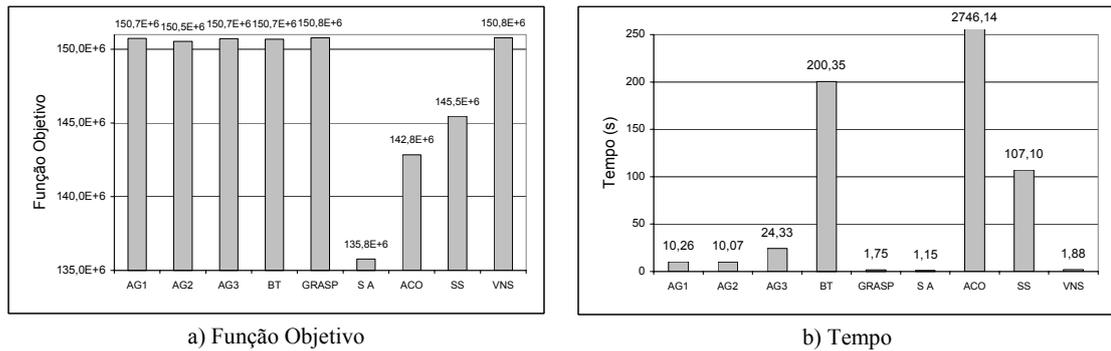


Figura 18 – Resultados obtidos pelo processamento das diferentes técnicas metaheurísticas (cenário 1)

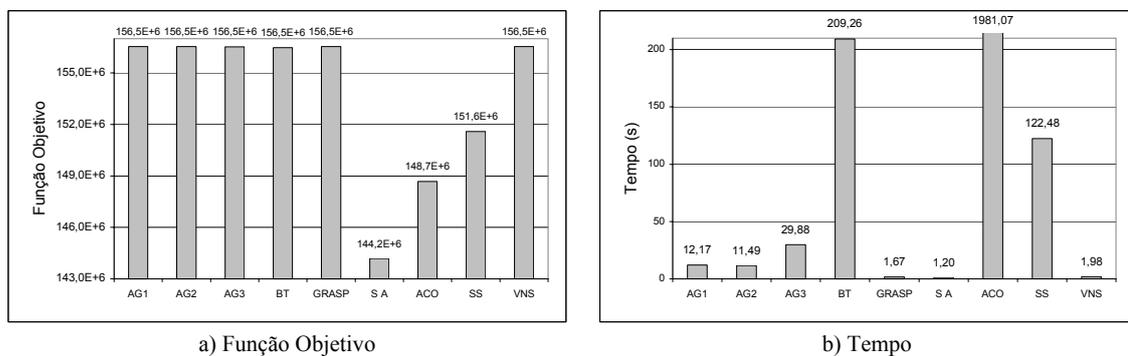


Figura 19 – Resultados obtidos pelo processamento das diferentes técnicas metaheurísticas (cenário 2)

A partir dos resultados apresentados, pode-se observar que as técnicas SA, ACO e SS resultam em soluções com valores da função objetivo muito inferiores aos outros métodos. O BT alcança valores da função objetivo muito bons embora o tempo computacional seja muito alto comparado com os outros métodos. Pode-se então dizer que as melhores soluções, segundo o tempo computacional versus a qualidade da solução, foram alcançadas pelo AG, GRASP e VNS.

Selecionou-se então o AG (AG2) e o VNS para um estudo mais detalhado, posteriormente.

As soluções serão obtidas utilizando o AG por dois motivos. O primeiro deles, como será visto posteriormente, é o fato de que o AG fornece soluções em um horizonte de *scheduling* menor que aqueles produzidos pelo VNS e GRASP para os problemas estudados. Em segundo lugar, a metaheurística AG pode ser facilmente estendida para o caso multiobjetivo, onde diferentes critérios (F.O.) serão tratados priorizando ou a demanda, ou a produção ou outro critério de acordo com a necessidade de programação.

Análise e Comparações de Diferentes Funções Objetivo

Com o objetivo de obter uma solução que atenda as várias exigências do modelo, foram realizados estudos de diferentes funções objetivo. Funções estas que têm por objetivo priorizar a entrega do produto no nó de demanda para que não apresente falta de produto para o consumidor. Assim, considerou-se os seguintes fatores na composição da função objetivo a ser otimizada:

- a janela de tempo no destino, que considera quando o nó de destino possibilita o recebimento de produto até o tempo em que o destino apresenta a necessidade de recebimento devido à demanda de produto;
- a janela de tempo na origem, que considera a curva de produção apresentando o tempo que o produto está disponível e o tempo limite para o envio do produto antes que exceda a capacidade de armazenamento do mesmo;
- uma ordenação de prioridade de envio entre os produtos derivados do petróleo.

Na seqüência serão apresentadas: uma fórmula heurística usada inicialmente pelo módulo de alocação para obter uma primeira ordenação de bateladas; e as diferentes funções objetivo, descritas a seguir, usadas no modelo de otimização do bloco de seqüenciamento, para análise e comparações dos resultados alcançados.

A tabela 15 traz a nomenclatura utilizada.

Tabela 15 – Nomenclatura dos objetivos usados

<i>TED</i>	tempo em que há disponibilidade do produto no nó de origem;
<i>TEC</i>	tempo limite, no qual deve ser retirado o produto da origem, para que não haja parada de produção na refinaria ou exceda a capacidade de tancagem;
<i>TRD</i>	tempo em que há capacidade de armazenamento do produto no nó de destino;
<i>TRC</i>	tempo limite, no qual deve ser recebido o produto no destino, para que não haja falta de produto ao consumidor;
<i>ordTRC_i</i>	ordenação da batelada <i>i</i> com relação ao <i>TRC</i> ;
<i>nprods</i>	número total de produtos;
<i>nprod_i</i>	ordenação do produto da batelada <i>i</i> em relação a todos os produtos;
α_j	pesos atribuídos a cada função <i>j</i> que compõe a função multiobjetivo.
<i>nbat</i>	número total de bateladas da carteira.

Fórmula da Alocação (Aloc.):

$$\begin{aligned}
 & TED + (TEC + 1 - TED) * \frac{TED}{TEC + 1} + \frac{TED + TEC}{TED + TEC + 1} \\
 & + TRD + (TRC + 1 - TRD) * \frac{TRD}{TRC + 1} + \frac{TRD + TRC}{TRD + TRC + 1}
 \end{aligned} \tag{3}$$

Esta primeira fórmula (Aloc.) foi desenvolvida a partir da forma de ordenação das bateladas realizada pelos especialistas. Esta heurística foi utilizada para a geração e ordenação das bateladas em Neves-Jr et al. (2007) e será utilizada como referência para as comparações entre as funções objetivo analisadas neste trabalho.

Função objetivo 1 (FO1):

$$\begin{aligned} & \sum_{i=1}^{nbat} \alpha_1 * (ordTRC_i) + \alpha_2 * (ordTED_i) \\ & + \alpha_3 * ord(TRC_i - TRD_i) + \alpha_4 * ord(TEC_i - TED_i) \\ & + \alpha_5 * (nprods - nprod_i), \end{aligned} \quad (4)$$

em que $\alpha_1 > \alpha_2 > \alpha_3 > \alpha_4 > \alpha_5$.

A função FO1 foi gerada visando obter uma ordenação das bateladas considerando primeiramente as bateladas com menor tempo de recebimento crítico no destino (TRC), seguido pelo menor tempo de disponibilidade do produto para o envio (TED), pelo menor tamanho da janela de tempo no destino e, em seguida, na origem, e por uma ordem de importância de envio de produtos.

Função objetivo 2 (FO2):

$$\begin{aligned} & \sum_{i=1}^{nbat} \left[\frac{\alpha_1 * TRC_i + \alpha_2 * (TRC_i - TRD_i) + \alpha_3 * TEC_i}{\sum_{j=1}^6 \alpha_j} \right. \\ & \left. + \frac{\alpha_4 * (TEC_i - TED_i) + \alpha_5 * TED_i + \alpha_6 * nprod_i}{\sum_{j=1}^6 \alpha_j} \right], \end{aligned} \quad (5)$$

em que $\alpha_1 > \alpha_2 > \alpha_3 > \alpha_4 > \alpha_5 > \alpha_6$.

Para a ordenação das bateladas, a função FO2 considera o menor tempo de recebimento do produto (TRC), o menor tamanho da janela de tempo no destino ($TRC - TRD$), o menor tempo de envio crítico na origem (TEC), o menor tamanho da janela na origem ($TEC - TED$) e o menor tempo de disponibilidade do produto para o envio (TED), nesta ordem de importância.

Função objetivo 3 (FO3):

Idem função objetivo 2, em que $\alpha_5 = \alpha_1 > \alpha_2 > \alpha_3 > \alpha_4 > \alpha_6$.

A função FO3, tem sua formulação semelhante à FO2, atribuindo igual importância para o tempo de recebimento crítico (TRC) e para o tempo de disponibilidade do produto na origem (TED).

Função objetivo 4 (FO4):

Idem função objetivo 2, em que $\alpha_5 > \alpha_1 > \alpha_2 > \alpha_4 > \alpha_6$ e $\alpha_3=0$.

A função FO4, tem sua formulação semelhante à FO2, descartando a consideração do tempo de envio crítico na origem, e atribuindo a maior importância para o menor tempo de disponibilidade do produto na origem.

Resultados

Na implementação para a análise das diferentes funções objetivo adotou-se, como explicado anteriormente, a técnica AG2, algoritmo genético híbrido. E a carteira de bateladas utilizada é referente ao cenário 1.

As figuras seguintes foram obtidas com o intuito de se avaliar a melhor função objetivo, dentre as funções apresentadas, a ser utilizada no seqüenciamento. Os resultados alcançados foram comparados também com a ordenação fornecida pela heurística dos operadores. A figura 20 mostra as médias dos tempos totais (*makespan*¹) requeridos pelo *scheduling* obtido com as diferentes funções objetivo. Pode ser observado que, neste caso, devido às diferentes considerações das quatro funções objetivo, apresentou-se um aumento do tempo total de até 24 horas, o que não representa muito (3%) levando em consideração que o horizonte de tempo foi de um mês. O menor tempo total foi obtido igualmente pela ordenação da alocação e da função FO1.

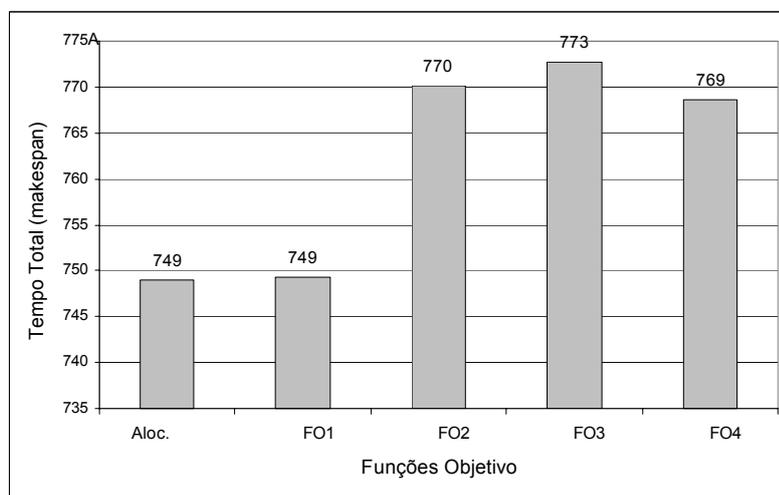


Figura 20 – Tempo total do *scheduling* obtido pelo seqüenciamento usando diferentes funções objetivo

Na figura 21 tem-se o somatório total das taxas de utilização dos dutos, quando ordenados pelas diferentes funções objetivo. Observando apenas a soma total da taxa de uso dos dutos não é possível concluir qual função fornece a melhor solução segundo o uso dos dutos. A soma total das taxas não mostra a dispersão destes valores. Assim, um baixo valor da soma

¹Makespan é o tempo total para que ocorram todas as operações de *scheduling* do horizonte de tempo estudado.

total da taxa de utilização, pode representar o somatório de alguns valores de baixa utilização dos dutos e alguns poucos valores de alta utilização, não permitindo visualizar a existência desses pontos críticos, ou seja, dutos críticos ou gargalos, na malha dutoviária. Portanto, também foram analisadas a média e desvio padrão das taxas de utilização, como pode ser visto na figura 22, com o objetivo de obter uma solução que represente um melhor valor do uso dos dutos, através de uma menor variação das taxas de utilização.

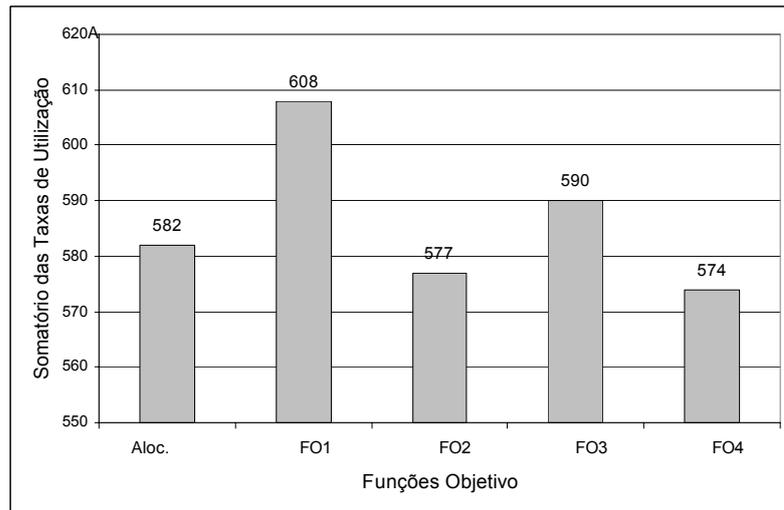


Figura 21 – Soma das taxas de utilização dos dutos obtidas pelo seqüenciamento usando diferentes funções objetivo

Pode-se observar que existe variação do somatório das taxas de utilização dos dutos. Essa variação não depende apenas do tempo total do *scheduling*, observado na figura 20, mas reflete que diferentes ordenações têm grande influência sobre o uso dos dutos.

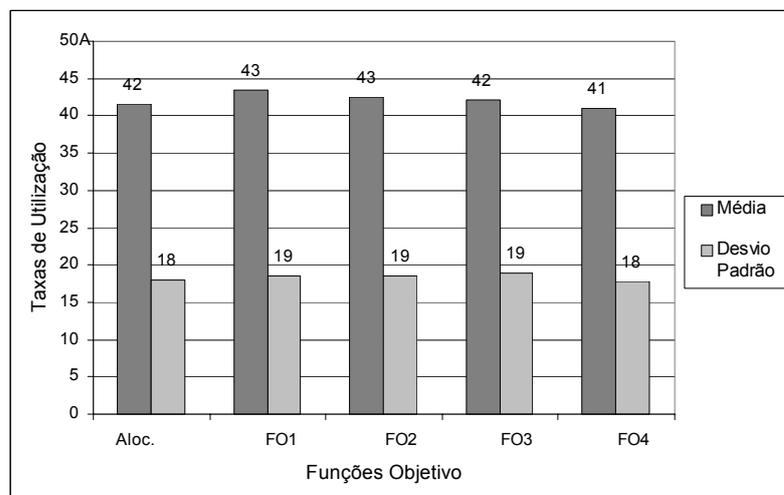


Figura 22 – Média e desvio padrão das taxas de utilização dos dutos obtidas pelas diferentes funções objetivo

A operação real da rede busca obter uma taxa de uso dos dutos sem muita discrepância de valores, ou seja, apresentando um menor desvio padrão. O uso uniforme dos dutos resulta em uma menor possibilidade de gargalos, ou na inexistência de dutos ociosos. Pode se observar que as soluções com maior desvio padrão da taxa de utilização dos dutos apresentaram maior média, da mesma forma as soluções com menor desvio padrão têm menor média da taxa de uso. Esses resultados fornecem um comportamento da solução que pode ser uma informação útil ao tomador de decisão ou operador da rede.

O envio sequencial de diferentes produtos em um mesmo duto provoca uma perda de qualidade na região de contato entre produtos devido a mistura na interface entre deles. Existem produtos em que, devido à grande perda em seu contato, há a necessidade do uso de um outro produto extra, denominado selo, para realizar a interface. Portanto, um menor número de transições entre diferentes produtos pode significar uma melhor solução, quanto a perda de produtos e uso de selos.

Na figura 23, a seguir, foi somado o número de transições entre diferentes produtos em um mesmo duto. Observou-se que as funções FO1 e FO4, resultaram em ordenações que favoreceram o envio seguido de bateladas de mesmo produto, diminuindo o número dessas interseções, pelo contrário, a heurística dos operadores obteve uma ordenação com maior número de intercalações de bateladas de diferentes produtos. Este comportamento reflete a prática operacional, que por não se ter uma ferramenta de auxílio na programação das operações, as bateladas enviadas são menores visando atender temporalmente pequenas demandas de produto e a seqüência de envio gera um número grande de transições entre diferentes produtos em todos os dutos.

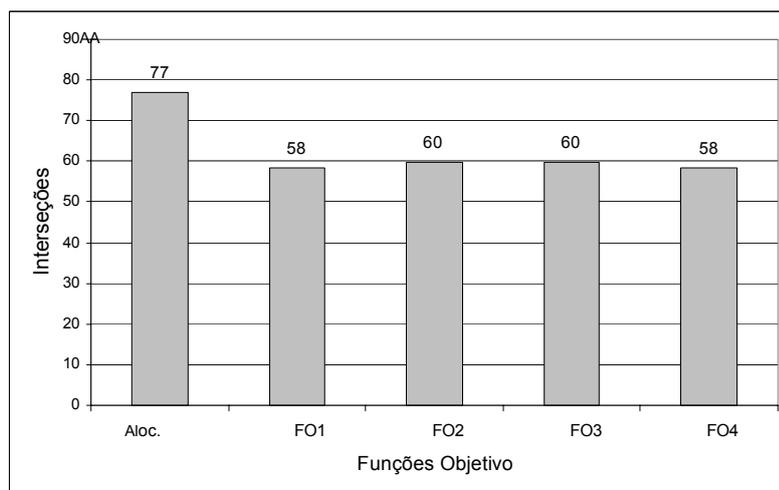


Figura 23 – Número total de transições entre diferentes produtos

O principal objetivo do presente trabalho é o melhor atendimento à demanda. Para suprir esse consumo é necessário uma produção que atenda a essa demanda. Portanto, para melhor

definir uma melhor solução para o problema de *scheduling* foram computadas as violações nos prazos de entrega de produtos apresentadas pelas soluções. Existem, também, violações no órgão produtor, quando um produto deve ser enviado antes que a batelada completa seja produzida ou depois quando ela deve desocupar espaço no tanque para novas campanhas. Essas violações são compostas pelo número de atrasos e adiantamentos do recebimento de produtos no destino e o número de atrasos e adiantamentos de envio de produtos na origem, que correspondem às violações das janelas de tempo das bateladas, descritas na seção 4.1.

A seguir, na figura 24, tem-se o número total de violações ocasionadas pelas soluções obtidas pelo seqüenciamento usando as diferentes funções objetivo.

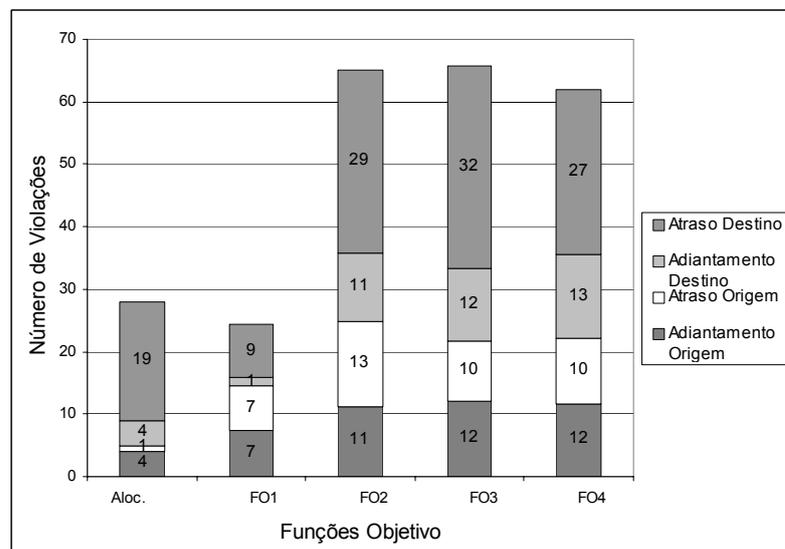


Figura 24 – Número total de violações ocorridas considerando as diferentes funções objetivo

Pode-se observar que, através do seqüenciamento usando a FO1, foi possível obter uma solução para o problema de *scheduling*, com um número de violações inferior. Foi significativa a diminuição do número de atrasos da chegada de produtos no destino, assim como o número de adiantamentos. Entretanto, os valores de violações na origem não apresentaram grandes mudanças. É esperado que se possa diminuir as violações no destino, visando o melhor atendimento aos clientes.

Algoritmo Genético

Número de Iterações

Para uma melhor análise do algoritmo genético foram efetuados testes do número de iterações do método para a obtenção de uma boa solução, através dos resultados dos valores da função objetivo. Os resultados são apresentados na figura 25 a seguir.

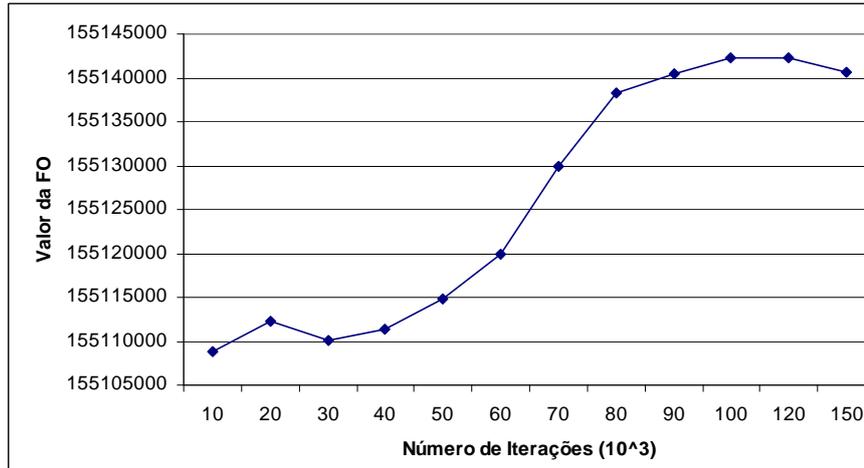


Figura 25 – Número de iterações para o Algoritmo Genético

Observou-se que as soluções do algoritmo genético apresentaram maiores valores para a função objetivo através da adoção de 100.000 iterações.

Fitness

Os estudos da melhor função objetivo a ser empregada, apresentado anteriormente, resultou na função apresentada a seguir.

Função Fitness:

$$\begin{aligned} &\alpha_1 * ord(TRC) + \alpha_2 * ord(TED) + \alpha_3 * ord(TRC - TRD) \\ &+ \alpha_4 * ord(TEC - TED) + \alpha_5 * ord(PROD) \end{aligned} \quad (6)$$

Com o intuito de se obter valores mais adequados aos coeficientes α_i empregados na função objetivo, foram realizados testes através de diferentes pesos, que podem ser vistos na tabela 16 a seguir.

Tabela 16 – Pesos da função objetivo

Caso	Pesos ($\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$)
1	(10.000,1.000,10,1,0)
2	(1.000,100,10,1,0)
3	(1.000,1.000,10,10,0)
4	(10.000,10,1.000,1,0)
5	(2.000,200,1.000,100,0)
6	(20.000,200,10.000,100,0)
7	(1.000,10,100,1,0)
8	(3.000,1.000,2.500,500,0)

A figura 26 apresenta a melhora percentual obtida, com relação ao resultado da heurística dos operadores, no resultado final de *scheduling* com o uso dos diferentes pesos na função de otimização da metaheurística no problema de seqüenciamento de bateladas. O maior valor para ambas as curvas representa melhores resultados quanto ao número de violações das janelas de tempo e transições entre diferentes produtos. Os resultados são mostrados para cada caso, com diferentes pesos, apresentado na tabela 16.

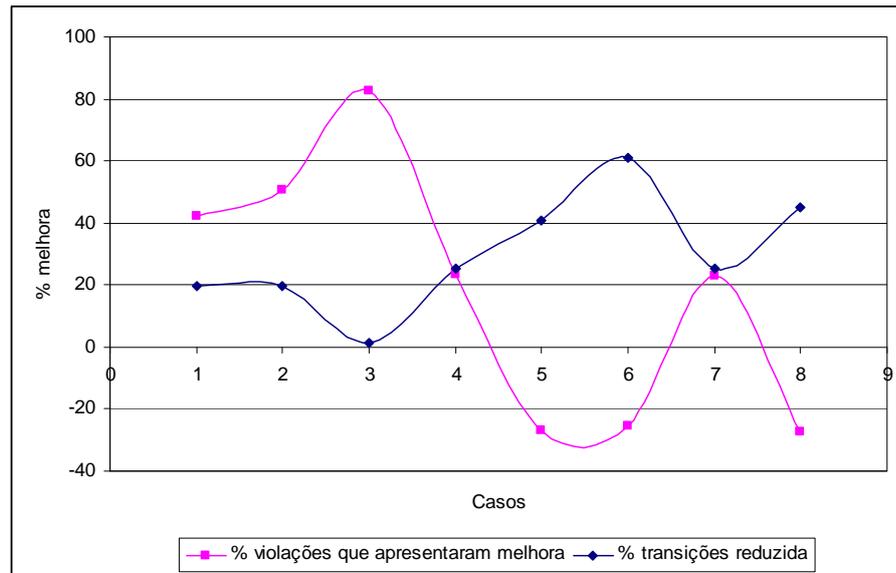


Figura 26 – Melhora do percentual de violações das janelas de tempo e transições entre produtos devido os diferentes pesos da função objetivo

Através dos resultados pode-se observar que as curvas são inversamente correlacionadas, isto é, quanto melhor é a solução com relação ao número de violações pior é com relação ao número de transições, e vice-versa. De acordo com os especialistas, a prioridade real das operações é alcançada minimizando as violações da janela de tempo, mesmo que isso não implique na redução das interfaces entre diferentes produtos. Portanto, observa-se que a melhor solução é dada pelo caso 3.

Elitismo

Para a obtenção de uma melhor solução final através da técnica algoritmo genético foi analisado número de indivíduos a serem mantidos na geração de uma nova população.

Utilizou-se as características mencionadas na tabela 17 a seguir.

A função objetivo usada foi:

$$1000 * ord(TRC) + 1 * ord(TED) + 10000 * ord(TRC - TRD) + 10 * ord(TEC - TED)$$

Tabela 17 – Parâmetros dos algoritmos

Tamanho da população	50 indivíduos
Número de iterações	10.001
Número de iterações para a criação de uma nova população	2.500
Número de indivíduos mantidos nas novas populações	variou-se de 0 a 50 indivíduos (0 a 100% da população)

O gráfico da figura 27, a seguir, mostra o comportamento da função objetivo em função do número de indivíduos mantidos na nova população e em função do tempo de processamento.

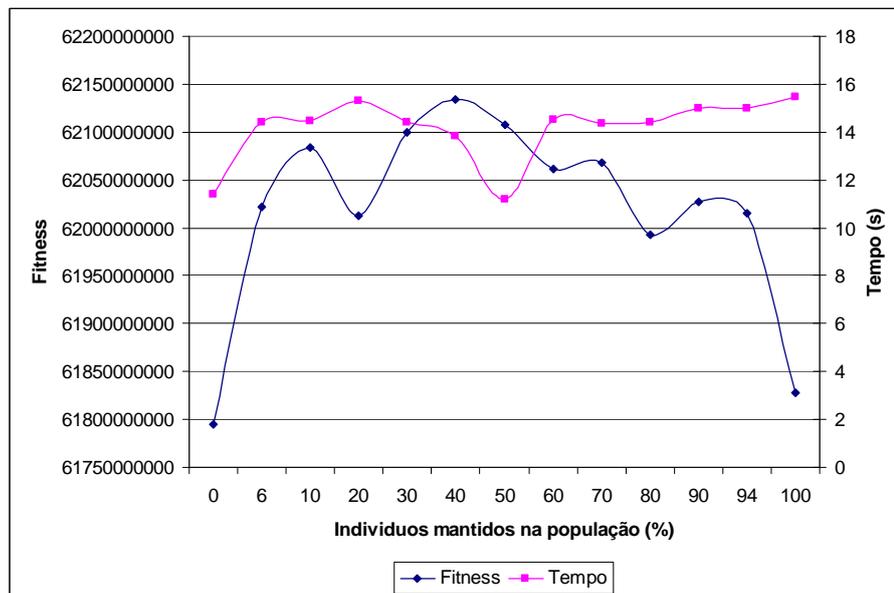


Figura 27 – Número de indivíduos mantidos em novas populações do Algoritmo Genético

Os resultados mostram uma pequena variação com relação ao tempo. Segundo o valor da função *fitness*, apresentaram melhores soluções quando foram mantidos 10% e 40% dos melhores indivíduos da população anterior na nova população gerada. Procurando manter a diversidade e buscando uma boa solução final, optou-se por manter o menor número de indivíduos, 5 melhores indivíduos ou 10%, em uma nova população gerada.

Mutação

Neste trabalho, a mutação é realizada apenas quando os filhos obtidos no processo de recombinação não são viáveis, ou são inactíveis.

Restrições

A geração de indivíduos viáveis de forma aleatória apresentou ser muito difícil devido às restrições referentes às bateladas de um mesmo produto que partem de um mesmo nó de ori-

gem. Estas possuem as janelas de tempo geradas de forma ordenada, por exemplo havendo três bateladas de mesmo produto que saem de um nó, elas terão suas janelas de tempo geradas nesta ordem não sendo possível invertê-las. Assim, visando retirar as soluções inviáveis do espaço de busca foram adotadas algumas formas de considerações iniciais para que as bateladas de mesma origem e produto, na geração das soluções, não possam apresentar a ordem invertida ao da geração, de forma a apenas ser possível a geração de bateladas factíveis, não sendo necessário o descarte, a penalização, ou o reparo de soluções.

Os procedimentos a seguir são adotados na geração de uma solução factível:

- Inicialmente as bateladas são ordenadas por ordem crescente de tempo de envio disponível (TED). Por exemplo, supondo 10 bateladas:

B1, B2, B3, B4, B5, B6, B7, B8, B9, B10.

- Posteriormente, respeitando a ordenação anterior, as bateladas são divididas em grupos de bateladas de mesmo produto e mesma rota. Do exemplo anterior, supondo duas rotas e dois produtos:

Rota 1/Produto 1: B1, B3

Rota 1/Produto 2: B2, B5, B6

Rota 2/Produto 1: B4, B8, B9

Rota 2/Produto 2: B7, B10

- A geração da solução é feita obedecendo a ordem dos itens anteriores. Do exemplo, temos:

Geração de uma solução: B1 B4 B2 B3 B8 B5 B7 B10 B6 B9.

Não é possível gerar uma solução como: B1 B7 B5 B4 B2 B10 B8 B3 B9 B6, pois não foi respeitada a ordem da Rota 1/Produto 2: B2, B5, B6.

Aplicações do AG

A seguir serão visualizados, através de gráficos de Gantt, dois exemplos de soluções de *scheduling* obtidos por diferentes sequenciamentos, para dois diferentes cenários (cenário 1 e cenário 3), e serão comparadas as características dos resultados.

Esses gráficos foram obtidos após o processo de temporização, através das diferentes ordenações das bateladas. Estes permitem obter as datas e horários do envio de todas as bateladas de uma carteira, apresentando a existência, ou não, de violações das janelas de tempo de envio e recebimento, o tempo total, o número de transições entre diferentes produtos e as taxas de utilização dos dutos.

As figuras 28a e 28b, foram geradas pela carteira de bateladas do cenário 1, através do ordenamento fornecido pela heurística construtiva e pelo algoritmo genético (AG2), respectivamente, através da função objetivo FO1.

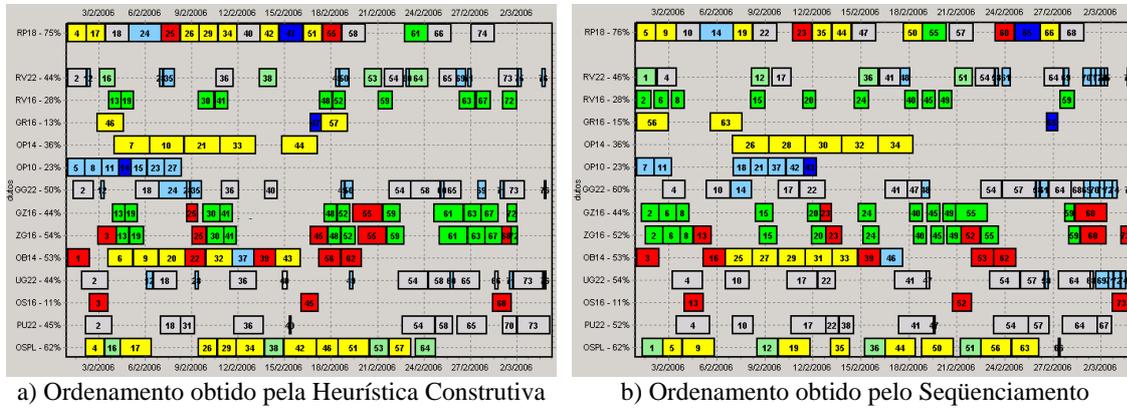


Figura 28 – Gráficos de Gantt para o cenário 1

As soluções do cenário 1 (figura 28) apresentaram seguintes características:

Características	Heurística Construtiva	Seqüenciamento
Violações	Origem:adiant.:4 atraso:1	Origem:adiant.:9 atraso:6
	Destino:adiant.:4 atraso:19	Destino:adiant.:2 atraso:7
	Total: 28	Total: 24
Tempo (horas)	749	742
Interseções	77	58
Taxa de	582	608
Utilização	(média:42, desvio padrão:18)	(média:43, desvio padrão:19)

Neste primeiro exemplo, pode-se observar que há uma diminuição no tempo total do *scheduling* em 7 horas, apresentada pela solução obtida através do ordenamento pelo módulo de seqüenciamento. Esta apresenta, também, um aumento da taxa de utilização dos dutos. O número de transições entre diferentes produtos diminuiu, ao comparar com a solução através da ordenação da heurística construtiva, representando menos perdas. Da mesma forma, através do seqüenciamento, houve uma diminuição do número de violações, considerado como um item de grande importância neste trabalho.

As figuras 29a e 29b, a seguir, mostram duas soluções obtidas para o cenário 3, por diferentes ordenações das bateladas. O gráfico à esquerda (29a) foi obtido pelo ordenamento fornecido pela heurística construtiva, e à direita (29b) pelo ordenamento obtido pelo algoritmo genético (AG2).

As soluções, apresentadas para o cenário 3 (figura 29), obtiveram as seguintes características:

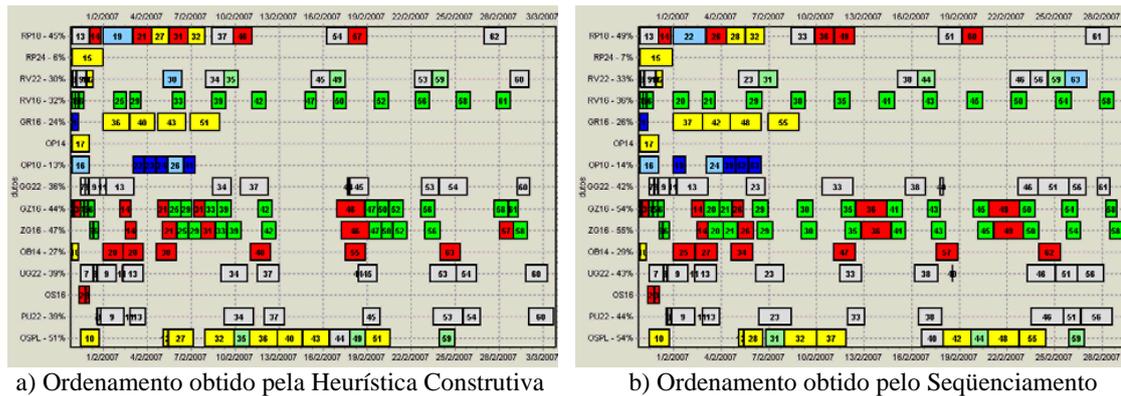


Figura 29 – Gráficos de Gantt para o cenário 3

Características	Heurística Construtiva	Seqüenciamento
Violações	Origem:adiant.:5 atraso:0 Destino:adiant.:0 atraso:5 Total: 10	Origem:adiant.:4 atraso:1 Destino:adiant.:0 atraso:2 Total: 7
Tempo (horas)	739	678
Interseções	46	46
Taxa de Utilização	433 (média:29, desvio padrão:17)	515 (média:34, desvio padrão:19)

Os resultados desse segundo exemplo mostram que o *scheduling* pode ser realizado em um tempo total inferior ao resultado fornecido através da ordenação fornecida pela heurística construtiva. Assim, pode-se atender à demanda mais rapidamente, neste caso, com uma diferença de 61 horas. Não houve alteração no número total de transições entre diferentes produtos nas duas soluções. Houve um aumento da taxa de utilização, total e média dos dutos através da ordenação fornecida pelo seqüenciamento. Assim como, apresentou um decréscimo do número total de violações das janelas de tempo.

Os resultados obtidos pela aplicação do algoritmo genético no seqüenciamento têm apresentado melhores soluções para o problema de *scheduling*. O seqüenciamento é um problema complexo de otimização combinatorial e esta ferramenta mostrou-se eficiente no auxílio da otimização da programação das operações de uma malha dutoviária.

VNS

O algoritmo VNS utilizou a mesma função objetivo usada no AG. A seguir serão mostrados os estudos efetuados para a adoção de melhores parâmetros, para o algoritmo VNS, do número de vizinhança a ser explorada e do número de aperfeiçoamentos realizados para cada nova solução adotada.

A tabela 18 mostra as características do VNS com a variação do número de vizinhanças a serem exploradas.

Tabela 18 – Número de vizinhança para o VNS

Número de Vizinhança	Violações Críticas (% Melhora)	Interfaces (% Redução)	Tempo (s)
10	22,17	25,4	9,25
15	23,85	25,4	10,22
20	22,73	25,4	13,08
25	22,17	25	15,87
30	23,85	25	17,55

A tabela 19 apresenta os resultados obtidos para a escolha do melhor número de aperfeiçoamentos a ser realizado para cada nova solução do algoritmo VNS.

Tabela 19 – Número de aperfeiçoamentos para cada nova solução do VNS

Número de Aperfeiçoamentos	Violações Críticas (% Melhora)	Interfaces (% Redução)	Tempo (s)
60	22,73	25,40	5,98
80	23,85	25,60	9,27
100	22,73	25,40	11,53
120	22,73	25,40	12,33
140	22,73	25,60	15,11
160	22,73	25,40	14,79

Para a adoção dos valores mais adequados para o número de vizinhanças (K) e para o número de buscas, foram observadas o percentual de melhora ou redução das violações críticas, isto é, violações de mais de seis horas, assim como o percentual de redução do número de interfaces entre diferentes produtos em um duto e o tempo computacional. Observou-se a obtenção dos melhores resultados através dos seguintes valores: número de estruturas de vizinhança K=15 e número de aperfeiçoamentos da solução igual à 80. Os resultados apresentados pelo uso do VNS no seqüenciamento mostraram uma melhora nas violações das janelas de tempo de em torno de 23% e de redução de interfaces de aproximadamente 25%. O tempo computacional utilizado é de aproximadamente 10 segundos.

Aplicações do VNS

A seguir tem-se os gráficos de Gantt referentes ao processamento através da ordenação das bateladas pela heurística construtiva (figura 30a) e pelo VNS (figura 30b) para o cenário 1, previamente descrito.

As soluções, da figura 30, apresentam as seguintes características:

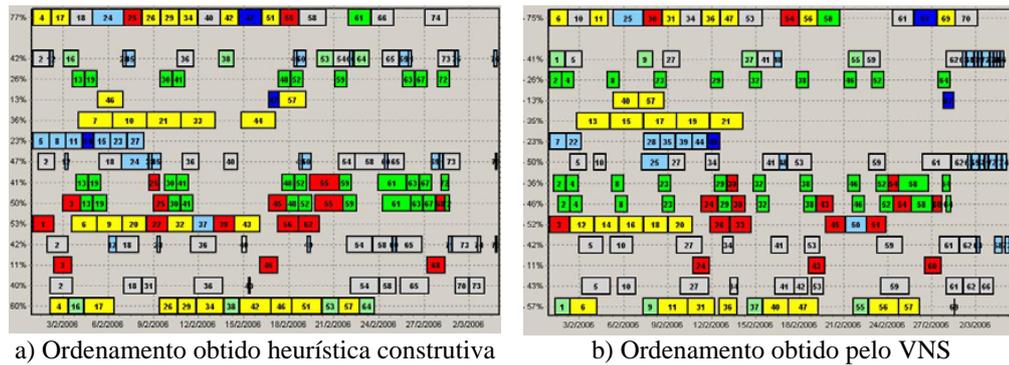


Figura 30 – Gráficos de Gantt para o cenário 1

Características	Heurística Construtiva	Seqüenciamento com VNS
Violações	Origem:adiant.:4 atraso:1 Destino:adiant.:4 atraso:18 Total: 27	Origem:adiant.:8 atraso:3 Destino:adiant.:1 atraso:8 Total: 20
Tempo (horas)	766	785
Interseções	79	57
Taxa de Utilização	561 (média:40, desvio padrão:17,9)	550 (média:39,3, desvio padrão:17,3)

Através das soluções apresentadas na figura 30 observa-se que o ordenamento das bateladas pela heurística construtiva resulta num total de 27 violações das janelas de tempo, com 18 de atrasos no destino e ao utilizar o VNS obteve-se uma redução para 20 violações, onde se tem 8 atrasos no destino. Apresentando uma grande melhora no atendimento à demanda de produtos. O tempo total para a realização de todas as operações programadas foi modificada de 766 horas, através da heurística construtiva, para 785 horas, com o uso do VNS. Houve um aumento total de 19 horas. Com referência ao número de transições entre diferentes produtos, houve uma grande redução com a otimização do seqüenciamento de bateladas, de 79 para 57 transições. Segundo as taxas de utilização dos dutos, pode-se observar que houve uma pequena redução, possivelmente devido ao pequeno aumento do tempo total, embora os valores da média e do desvio-padrão tenham permanecido muito próximos, mostrando não haver muita variação do uso dos dutos. Os dutos de maior uso permanecem sendo o primeiro e o último duto (figuras 3 e 4), com 77% e 60% de uso na solução segundo a heurística construtiva, e 75% e 57% de utilização na solução com o seqüenciamento pelo VNS.

Conclusões

Neste capítulo foram realizadas comparações entre diferentes modelos baseados em metaheurísticas e observou-se através dos resultados alcançados que as melhores técnicas foram o AG e o VNS. Analisou-se, também uma melhor função objetivo a ser empregada na otimização do problema de seqüenciamento das bateladas. Assim, o AG e o VNS foram utilizados na ordenação das bateladas para a verificação da otimização do problema de *scheduling* operacional. Os resultados mostraram que a otimização do problema de seqüenciamento das bateladas implica em melhoramentos do *scheduling* da rede. O modelo AG pode ser considerado um melhor modelo devido à apresentação de redução do tempo total de *scheduling* do número de violações das janelas de tempo e de interfaces entre diferentes produtos, além da apresentação de um conjunto de soluções opcionais ao tomador de decisão. O modelo VNS apresentou bons resultados com a redução do número de violações das janelas de tempo e de interfaces, entretanto houve um aumento do tempo total de *scheduling* e para a apresentação de diferentes soluções opcionais faz com que seja necessário novas simulações.

No capítulo a seguir será apresentado o estudo da técnica de otimização multiobjetivo MOGA, no sequenciamento das bateladas.

6 SEQÜENCIAMENTO DE BATELADAS ATRAVÉS DO MODELO MOGA

Este capítulo apresenta a abordagem do problema de sequenciamento das bateladas *scheduling* de operações aplicando um modelo baseado em uma técnica de otimização multiobjetivo, o MOGA (*Multiobjective Genetic Algorithm*), e serão mostrados os resultados obtidos através deste modelo.

6.1 MOGA

O MOGA (*Multiobjective Genetic Algorithms*) difere do algoritmo genético convencional pela operação de seleção. Dois aspectos importantes do MOGA são: a atribuição do *fitness* baseado no ranqueamento e a implementação do *sharing* no domínio do valor da função objetivo. O conceito da dominância de Pareto é usada juntamente com as preferências do tomador de decisão para estimar o desempenho individual, mantendo os objetivos separados durante o processo de otimização (FONSECA; FLEMING, 1994).

Fitness

Método de ranqueamento do *fitness* para o MOGA

Considere o indivíduo x , da geração t que é dominado por $p(t)$. Sua posição do ranking é dada por:

$$\text{rank}(x,t) = 1 + p(t)$$

Toda população não-dominada tem ranque 1. Segundo a figura 31, a seguir, para um problema de minimização, observa-se que as soluções de ranque 1 não possuem nenhuma outra solução que apresente menor valor de f_1 e f_2 . Assim como a solução de ranque 2 possui 1 solução com menor valor de f_1 e f_2 , o ranque 3 apresenta 2 e o ranque 5, 4 soluções.

O método tradicional de atribuição do *fitness* baseado no ranqueamento é dado da seguinte forma:

1. classifica a população de acordo com o ranque;
2. atribui o *fitness* ao indivíduo pela interpolação do melhor (ranque 1) ao pior, de acordo

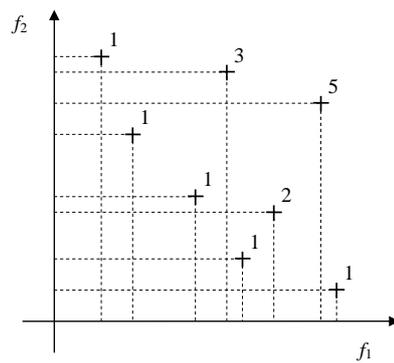


Figura 31 – Método do ranqueamento
(Fonte: Fonseca e Fleming (1993))

com uma função, geralmente linear;

3. calcula a média do *fitness* dos indivíduos de mesmo ranque, sendo assim exemplificados com uma mesma proporção. Este procedimento mantém o *fitness* global da população constante enquanto mantém uma pressão apropriada de seleção, definida pela função usada.

Diversidade

Manter a diversidade da população, em problemas multiobjetivos usando AG, é muito importante para a obtenção de soluções uniformemente distribuídas sobre a fronteira de Pareto. A população do MOGA tende a formar poucos agrupamentos (KONAK et al., 2006), fenômeno chamado de tendência genética. Faz-se necessário o uso de alguma técnica que previna a alta densidade populacional em poucas áreas e permita uma busca em novas regiões. Este fenômeno é tratado por nichos, que encorajam a busca em regiões inexploradas da fronteira de Pareto pela penalização das soluções localizadas em áreas densamente populadas.

Método de formação de nichos para o MOGA

O método de formação de nichos evita o fenômeno de tendência genética. Entretanto, é introduzido um novo parâmetro no algoritmo genético, o fator de tamanho do nicho (σ_{share}), que deve ser adotado cuidadosamente.

As teorias existentes que ajustam esse valor assumem que o conjunto solução tem um número finito de picos e tem a localização uniforme de nichos (FONSECA; FLEMING, 1993). Durante a convergência, os ótimos locais são ocupados por um grupo de indivíduos, proporcional aos valores de seus *fitness*. Entretanto, ao contrário, as soluções globais de problemas multiobjetivos são planas em termos do *fitness* individual, e não se tem como saber, de antemão, o tamanho do conjunto de soluções. Ainda, o uso do ranqueamento faz com que a busca se

concentre apenas no ótimo global. Pela implementação do *fitness sharing* no domínio do valor da função objetivo, assim como no domínio das variáveis de decisão em um par de indivíduos não-dominados permite representar uma distribuição uniforme da superfície de fronteira global.

No *fitness sharing*, o nicho é incorporado no método estendido de ranqueamento do *fitness* para escalar os *fitness* dentro de cada ranque. A proporção do *fitness* alocado no conjunto de indivíduos não-dominados será independente dos coeficientes *sharing*.

Elitismo

Para o algoritmo genético de um único objetivo, o elitismo significa que as melhores soluções encontradas durante a busca sempre permanecem na população para as próximas gerações. No caso de multiobjetivo, todas as soluções não-dominadas são consideradas soluções elites. Entretanto, a consideração do elitismo não é tão simples devido ao grande número de soluções elite. Anteriormente (até o final dos anos 90), não se usava o elitismo em problemas multiobjetivo, mas recentemente tem-se duas estratégias: manter as soluções elites na população ou guardá-las em uma lista externa e reintroduzi-las à população. No problema em estudo não foi usada técnica elitista no MOGA.

Restrições

Muitos problemas reais de otimização possuem restrições a serem satisfeitas. Para problemas de um único objetivo utiliza-se uma das quatro estratégias: (i) descarte de soluções infactíveis; (ii) redução do *fitness* das soluções infactíveis através de uma função penalidade; (iii) fazer com que as operações genéticas produzam apenas soluções factíveis; e (iv) transformar soluções infactíveis em factíveis (reparo). As estratégias (i), (iii) e (iv) são perfeitamente aplicáveis ao problema multiobjetivo, entretanto a implementação de uma função penalidade não é possível devido a atribuição do *fitness* se basear no ranque de não-dominância, e não em valores de função objetivo (KONAK et al., 2006).

A escolha do MOGA

Segundo alguns trabalhos (COELLO, 2000, 1999a, 1999b), o MOGA é uma boa técnica, eficiente e relativamente de fácil implementação, embora, assim como as outras técnicas baseadas no ranqueamento de Pareto, seu desempenho dependa muito de uma apropriada seleção do fator *sharing* (σ_{share}). Porém, Fonseca e Fleming (1993) desenvolveram uma boa metodologia para o cálculo desse fator, que será mostrada na próxima seção. Outras técnicas de otimização multiobjetivo necessitam de, além do σ_{share} , mais parâmetros.

O VEGA é uma técnica simples porém, nota-se que o agrupamento das sub-populações correspondem a uma média dos *fitness* associados a cada objetivos, assim o *fitness* resultante corresponde a uma combinação linear dos objetivos, tendo assim a incapacidade de produzir soluções ótimas de Pareto para espaços de busca não-convexos (COELLO, 1999b).

A técnica NSGA é mais ineficiente que o MOGA, tanto computacionalmente quanto na qualidade de produção da fronteira de Pareto, e ainda é mais sensível ao valor do fator *sharing* (σ_{share})(COELLO, 1999b). Segundo Coello (1996), o MOGA é mais rápido que o NSGA, pois o algoritmo de ranqueamento é mais eficiente e usa uma estratégia de seleção.

O NPGA não utiliza a seleção de Pareto para toda a população, portanto é rápido e produz uma boa fronteira de Pareto. Entretanto requer uma boa escolha do fator *sharing* e do tamanho do número de indivíduos do conjunto de comparação, que dificulta seu uso na prática (COELLO, 2000).

O SPEA destaca-se pela ausência de parâmetros de distância, como raio de nicho, compartilhamento e valores de *fitness* com relação às soluções do conjunto de Pareto (CASTRO, 2001b). Entretanto, a eficiência depende do tamanho do conjunto externo de soluções não-dominadas (COELLO, 2006).

Baseado nessas considerações e na facilidade de implementação à partir do AG existente, optou-se pelo MOGA.

6.1.1 Discussões

O modelo MOGA utilizado neste trabalho segue o seguinte algoritmo, apresentado por Konak et al. (2006):

Algoritmo MOGA

Passo 1: Comece com uma população inicial aleatória P_0 . Faça $t = 0$.

Passo 2: Se o critério de parada for satisfeito, retorne P_t .

Passo 3: Avaliar o *fitness* da população da seguinte forma:

Passo 3.1: Determinar o ranque $r(x, t)$ para cada solução $x \in P_t$ usando a equação abaixo, em que Fonseca e Fleming (1990) usam uma abordagem de ranqueamento baseado na fronteira não-dominante.

$$r(x, t) = 1 + nq(x, t),$$

onde $nq(x, t)$ é o número de soluções que dominam x na iteração t .

Passo 3.2: Avaliar o valor do *fitness* de cada solução, baseado no ranque, de acordo

com a equação a seguir:

$$f(x, t) = N - \sum_{k=1}^{r(x,t)-1} n_k - .5(n_{r(x,t)} - 1)$$

onde n_k é o número de soluções de ranque k , N é o tamanho da população.

Passo 3.3: Calcular o "niche count" $nc(x, t)$ de cada solução $x \in P_t$:

Passo 3.3.1: Calcular a distância Euclidiana entre todos os pares de soluções x e y no espaço objetivo normalizado entre 0 e 1 da forma:

$$d(x, y) = \sqrt{\sum_{k=1}^K \left(\frac{z_k(x) - z_k(y)}{z_k^{max} - z_k^{min}} \right)^2}$$

onde z_k^{max} e z_k^{min} são o valor máximo e mínimo da função objetivo $z_k(\cdot)$, respectivamente, observado durante a busca.

Passo 3.3.2: Baseado nas distâncias, calcular o "niche count" para cada solução $x \in P$:

$$nc(x, t) = \sum_{y \in P, r(y,t)=r(x,t)} \max\left\{ \frac{\sigma_{share} - dz(x, y)}{\sigma_{share}}, 0 \right\},$$

onde σ_{share} é o tamanho do nicho.

Passo 3.4: Calcular o valor do "shared fitness" de cada solução $x \in P_t$ da seguinte forma:

$$f'(x, t) = \frac{f(x, t)}{nc(x, t)}$$

Passo 3.5: Normalizar o valor do "fitness" pelo uso do valor do "shared fitness".

$$f''(x, t) = \frac{f'(x, t) n_{r(x,t)}}{\sum_{y \in P_t, r(y,t)=r(x,t)} f'(x, t)} f(x, t)$$

Passo 4: Usar o método de seleção estocástica baseado em f'' para a seleção dos pais para aplicar a recombinação e a mutação, até que população gerada Q_t de tamanho N esteja completa. Faça $P_{t+1} = Q_t$.

Passo 5: Faça $t = t + 1$, vá para o passo 2.

Objetivos

Para o problema de seqüenciamento das bateladas foram considerados dois objetivos. As informações deste módulo são os dados de saída do módulo de alocação, mostrados na seção 4.2. Considerando os estudos realizados através do algoritmo genético, os dois critérios adotados são: as janelas de tempo das bateladas na origem e no destino como mostrados a seguir.

$$criterio1 = \alpha_1 * ord(TRC) + \alpha_2 * ord(TRC - TRD)$$

$$criterio2 = \alpha_3 * ord(TED) + \alpha_4 * ord(TEC - TED)$$

em que os pesos são os mesmos utilizados no algoritmo genético $\alpha_1 = 1000, \alpha_2 = 1000, \alpha_3 = 10e\alpha_4 = 10$ (seção 5.1.1.3).

Parâmetro σ_{share}

Um dos principais parâmetros usado no MOGA é o σ_{share} . Este parâmetro estabelece a distância entre dois indivíduos com a qual gera um decremento de um dos critérios do *fitness* (FONSECA; FLEMING, 1993).

A forma de cálculo do σ_{share} adotada por Fonseca e Fleming (1993) foi usada neste trabalho. Este considera o máximo número de pontos que exemplificam uma área "A" sem interferir em outro. Seu valor pode ser computado como a diferença em volume entre dois hiperparalelogramos, como será descrito a seguir.

Seja S o conjunto de soluções do domínio de variáveis de decisão, $f(S)$ o conjunto de soluções do domínio dos objetivos e $y = (y_1, y_2, \dots, y_q)$ um vetor objetivo em $f(S)$. E $m = (miny_1, \dots, miny_q) = (m_1, \dots, m_q)$ e $M = (maxy_1, \dots, maxy_q) = (M_1, \dots, M_q)$.

A figura 32 exemplifica m e M para um caso tridimensional, mostra uma superfície de um conjunto solução limitada externamente por um paralelepípedo formado por m e M .

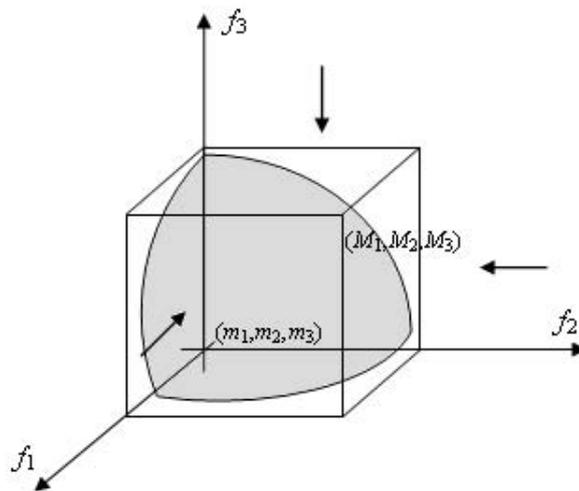


Figura 32 – Exemplo de superfície tridimensional
(Fonte: Fonseca e Fleming (1993))

A maior área corresponderá ao hiperparalelepípedo definido por m e M , como pode ser visto, para o exemplo tridimensional, o paralelepípedo da figura 33. E a hiperárea será menor que:

$$A = \sum_{i=1}^q \prod_{j=1, j \neq i}^q (M_j - m_j)$$

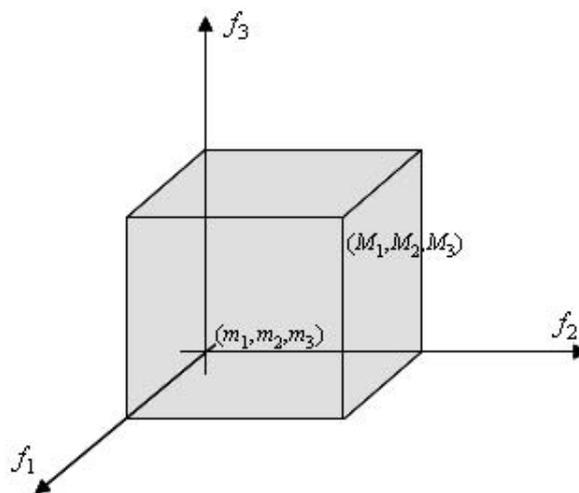


Figura 33 – Paralelepípedo definido por (m_1, m_2, m_3) e (M_1, M_2, M_3) , que limita a área da superfície

(Fonte: Fonseca e Fleming (1993))

Devido a possibilidade dos objetivos serem não mensuráveis, o uso da norma de Tchebycheff (PRENTER, 1975) para medir a distância entre os indivíduos parece a mais natural e simples de calcular. Neste caso é necessário especificar o σ_{share} para cada objetivo.

Assumindo os objetivos normalizados, então todos os parâmetros σ_{share} são os mesmos. O número máximo de pontos que podem exemplificar a área "A" sem interferência é o número de hipercubos do volume σ_{share}^q que cabem no hiperparalelogramo A, como pode ser visto na figura 34.

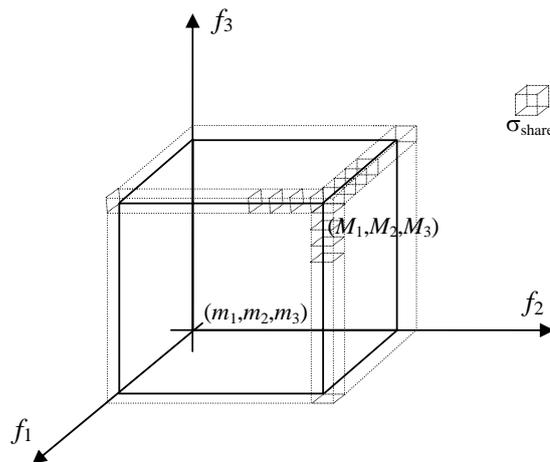


Figura 34 – Exemplificação da Área "A". Cada ponto está a uma distância σ_{share} dos vizinhos (Fonte: Fonseca e Fleming (1993))

A equação a seguir calcula a diferença em volume de dois hiperparalelogramos, um com a margem $(M_i - m_i + \sigma_{share})$ e outro com $(M_i - m_i)$, dividido pelo volume do hipercubo com margem σ_{share} .

$$N = \frac{\prod_{i=1}^q (M_i - m_i + \sigma_{share}) - \prod_{i=1}^q (M_i - m_i)}{\sigma_{share}^q}$$

Dado um número de indivíduos (pontos) N é possível calcular o σ_{share} resolvendo a equação polinomial de ordem $q - 1$.

Número de indivíduos da população e número de iterações

Foram realizados vários testes com a finalidade de se obter o número de indivíduos da população e o número de iterações adequados para o programa. Os gráficos das figuras 35, 36, 37, 38 e 39 a seguir mostram os resultados obtidos neste estudo.

A figura 35 mostra os valores obtidos da função objetivo em relação ao número de iterações, assim como o tempo computacional em relação ao número de iterações, para uma população de 20 indivíduos. As figuras 36, 37, 38 e 39 mostram os valores obtidos para uma

população de 50, 100, 150 e 200 indivíduos, respectivamente. Os resultados apresentados foram obtidos por 10 experimentos para cada caso.

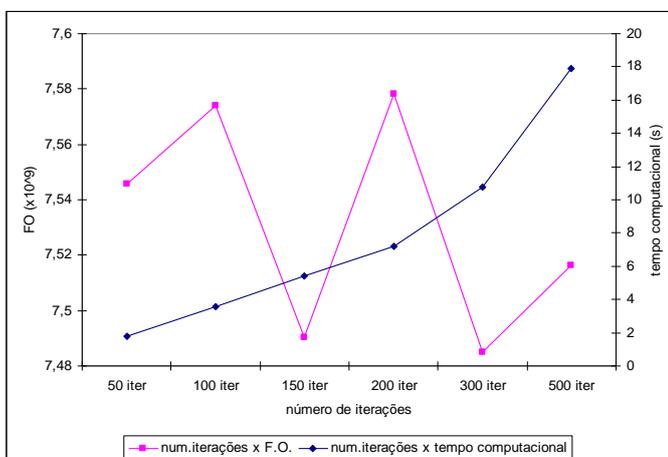


Figura 35 – População=20 indivíduos

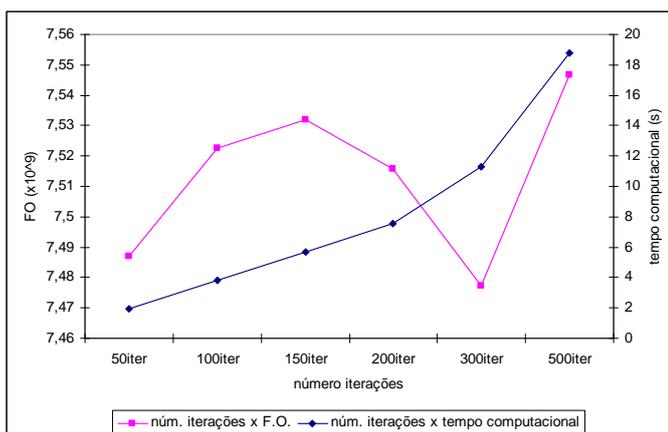


Figura 36 – População=50 indivíduos

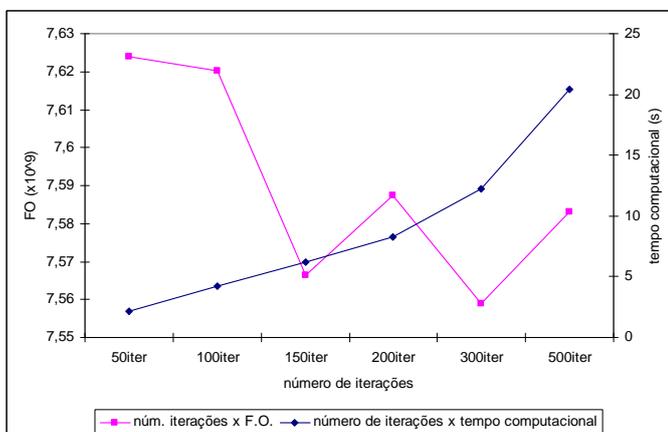


Figura 37 – População=100 indivíduos

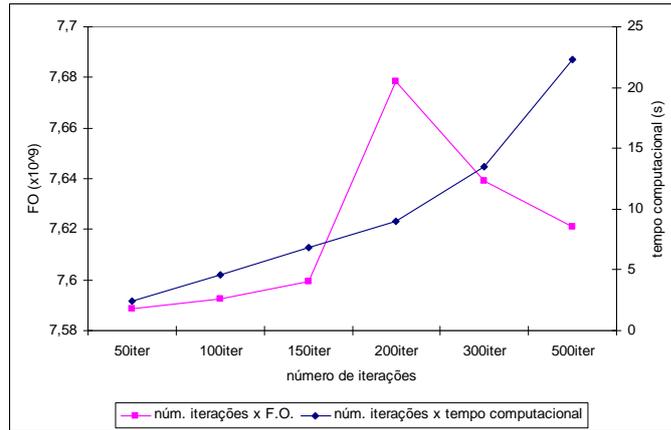


Figura 38 – População=150 indivíduos

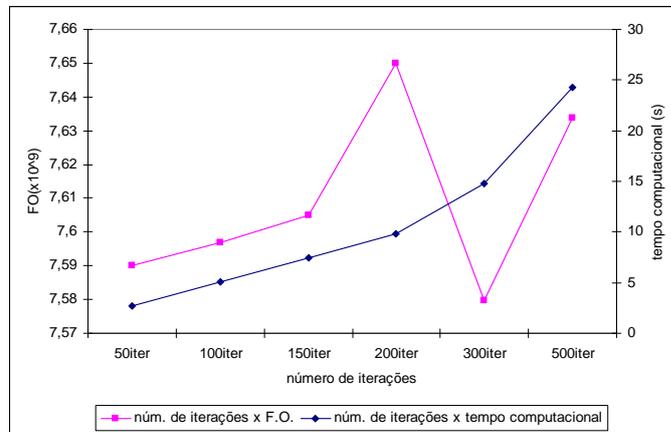


Figura 39 – População=200 indivíduos

Observou-se que a população com 150 e 200 indivíduos apresentaram melhores soluções quanto ao valor da função objetivo, tendo assim sido adotado 150 indivíduos na população.

Ao observar os resultados da análise da função objetivo final alcançada e o tempo computacional obtido, em função do melhor número de iterações, optou-se pela adoção de 200 iterações.

6.1.2 Aplicações

Abaixo tem-se alguns exemplos dos resultados obtidos pelo modelo de otimização multiobjetivo através do MOGA. Observou-se, após uma série de testes, que o conjunto de solução encontrado não necessariamente representa uma solução com as melhores características com relação ao número de violações das janelas de tempo nem pelo tempo total (*makespan*). A maioria das soluções apresentaram uma redução do número de transições entre diferentes produtos, mesmo não tendo esta característica como um dos objetivos de otimização.

Exemplificando, a tabela a seguir mostra as características de algumas soluções de

scheduling obtidas para o cenário 3 (apresentado no capítulo 5), sendo apresentadas as cinco melhores soluções de ordenamento da população final em três diferentes processamentos.

Tabela 20 – *Características das Soluções*

Solução (solução_ordenada)	Tempo Total (horas)	Número de Transições	Número de Violações
Heurística Construtiva	734	37	Origem: adiant:7 atraso:0 Destino: adiant:2 atraso:11 Total:20
1_1	717	35	Origem: adiant:8 atraso:1 Destino: adiant:3 atraso:9 Total:21
1_2	754	35	Origem: adiant:11 atraso:3 Destino: adiant:1 atraso:7 Total:22
1_3	734	27	Origem: adiant:9 atraso:3 Destino: adiant:3 atraso:10 Total:25
1_4	751	28	Origem: adiant:11 atraso:3 Destino: adiant:5 atraso:14 Total:33
1_5	734	34	Origem: adiant:10 atraso:3 Destino: adiant:4 atraso:7 Total:24
2_1	734	35	Origem: adiant:8 atraso:2 Destino: adiant:4 atraso:8 Total:22
2_2	734	30	Origem: adiant:9 atraso:1 Destino: adiant:3 atraso:11 Total:24
2_3	734	35	Origem: adiant:8 atraso:2 Destino: adiant:1 atraso:7 Total:18
2_4	734	38	Origem: adiant:11 atraso:2 Destino: adiant:1 atraso:8 Total:22
2_5	734	36	Origem: adiant:8 atraso:2 Destino: adiant:3 atraso:10 Total:23
3_1	734	36	Origem: adiant:6 atraso:2 Destino: adiant:4 atraso:9 Total:21
3_2	774	30	Origem: adiant:8 atraso:1 Destino: adiant:3 atraso:8 Total:20
3_3	765	31	Origem: adiant:10 atraso:3 Destino: adiant:4 atraso:6 Total:23
3_4	734	29	Origem: adiant:8 atraso:1 Destino: adiant:2 atraso:7 Total:18
3_5	763	31	Origem: adiant:8 atraso:1 Destino: adiant:3 atraso:6 Total:18

Através dos resultados mostrados na tabela 20 pode-se verificar que a solução 1_1 apresentou uma diminuição do tempo total de 17 horas com relação à solução da heurística construtiva, reduzindo ainda duas transições entre diferentes produtos, embora tenha gerado uma violação a mais. A solução 3_4 também apresentou melhoras comparadas à solução da heurística construtiva, através de um mesmo tempo total de *scheduling* diminuiu 8 transições entre diferentes produtos e um total de 2 violações, em que houve uma redução de 11 para 6 atrasos no destino melhorando efetivamente o atendimento à demanda.

Uma grande característica dos resultados obtidos através desse sequenciamento por multiobjetivo foi a diversidade do conjunto solução. As soluções finais mostraram ser bem diferentes, apresentando uma ordenação bastante diferenciada das bateladas entre elas, como pode ser visto através de alguns exemplos reais nas figuras a seguir. As figuras 40 e 41 mostrarão as três melhores soluções do *scheduling* de operações da rede obtidas em dois processamentos

distintos.

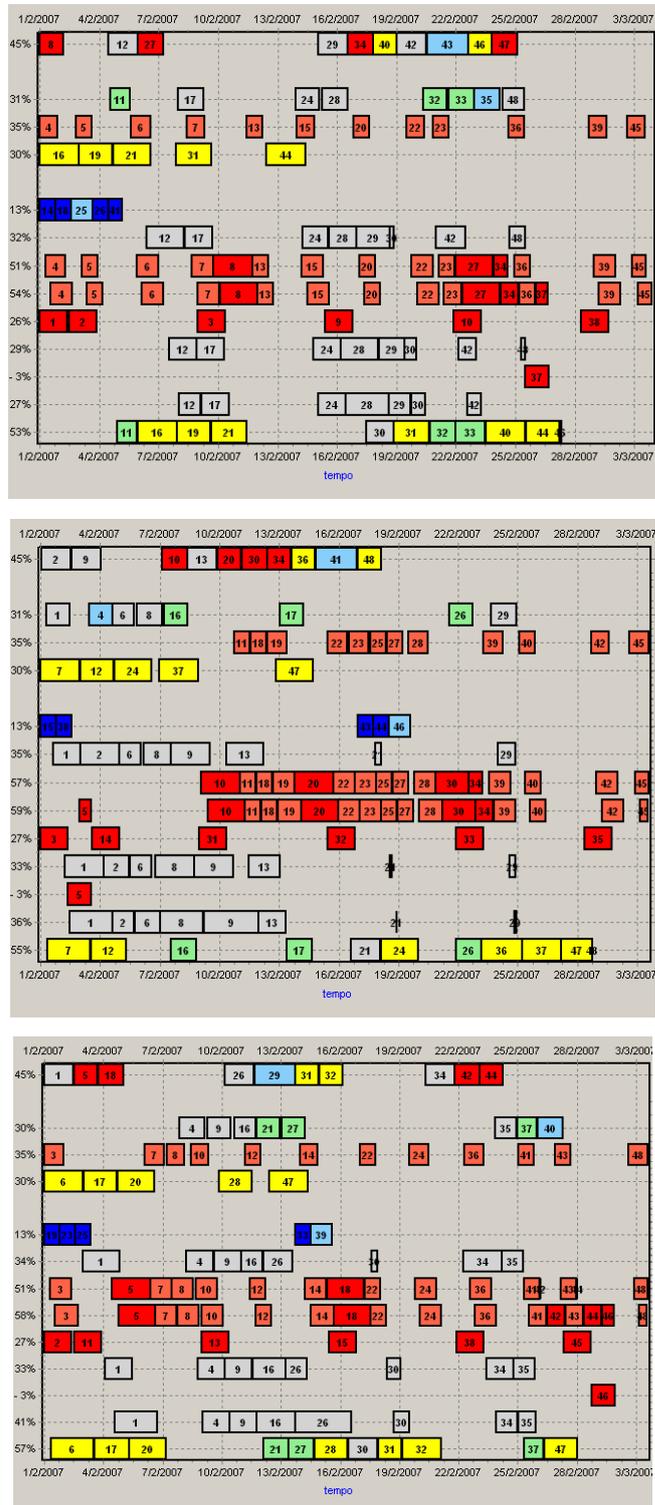


Figura 40 – Solução 1, 2 e 3 do cenário 1

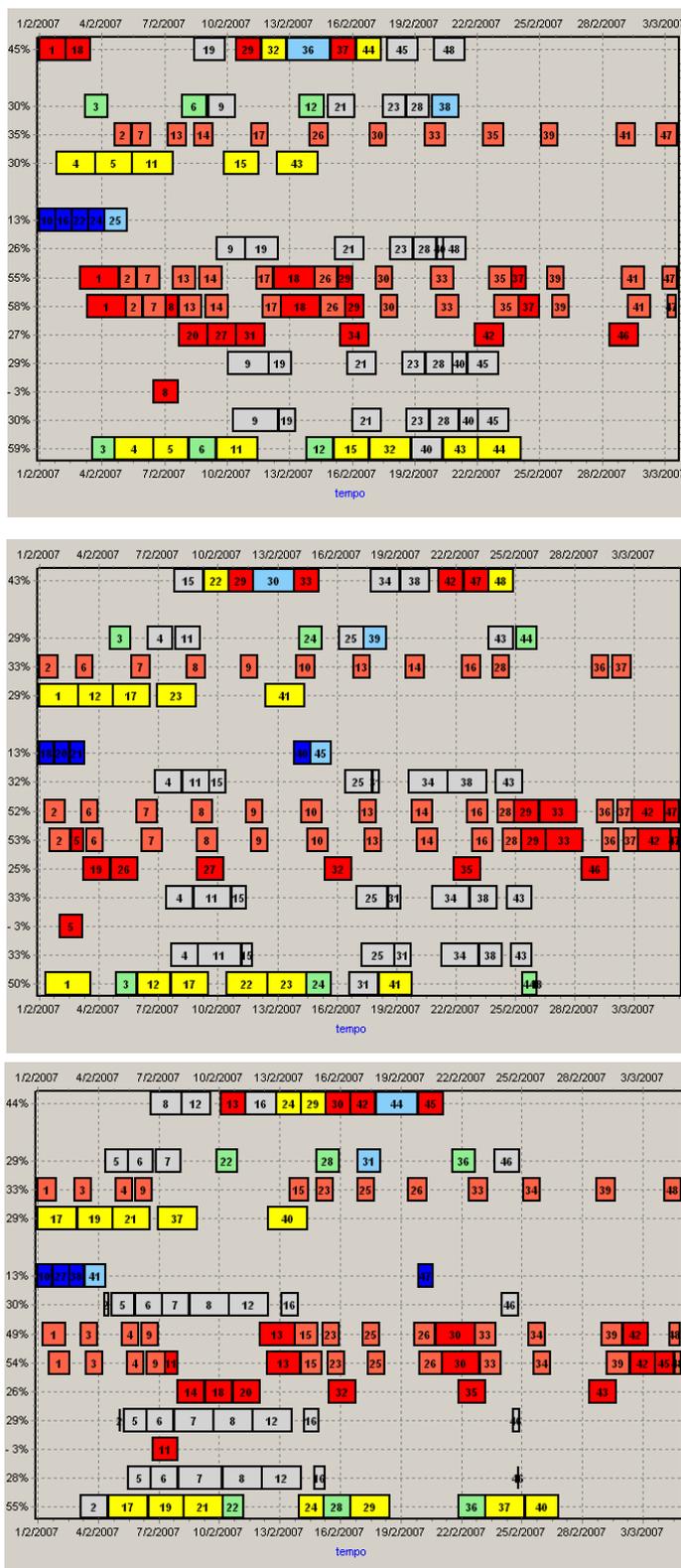


Figura 41 – Solução 1, 2 e 3 do cenário 2

A apresentação de diferentes soluções representa ao operador uma maior possibilidade de operação da malha.

O sequenciamento das bateladas, como previamente mencionado, é um problema mul-

tiobjetivo e observou-se ser possível a otimização através de modelos baseados em técnicas metaheurísticas, como o AG ou VNS, utilizando uma função objetivo que realize uma combinação dos muitos critérios a serem otimizados. Entretanto através do uso de uma técnica de otimização multiobjetivo, o MOGA, pode-se observar que as melhores soluções obtidas após um único processamento permitem apresentar, além da minimização das violações das janelas de tempo, um conjunto diferenciado de opções de operações da rede, sendo uma boa ferramenta de apoio ao tomador de decisão. Neste estudo considerou-se na otimização multiobjetivo dois critérios, a otimização dos tempos na origem e no destino; entretanto, novos critérios poderão ser considerados em estudos futuros.

A conclusão deste trabalho e a indicação de pesquisa para futuros trabalhos são apresentados no próximo capítulo.

7 CONCLUSÕES E TRABALHOS FUTUROS

7.1 CONCLUSÕES

Devido à complexidade do problema de *scheduling* da rede de dutos, faz-se necessária uma abordagem de subdivisão do problema. O sucesso de cada etapa tem grande influência no resultado final do *scheduling*. Portanto há a necessidade de análises e avaliações de métodos de otimização que possam fornecer melhores soluções ao problema como um todo e a cada subproblema em particular.

Para a resolução do problema de *scheduling* operacional adotou-se uma abordagem híbrida. Foram adotados uma modelagem para cada subproblema do *scheduling*, onde um algoritmo é executado após o outro, de forma colaborativa, em que os algoritmos trocam informações entre eles, mas um não faz parte de outro. Os algoritmos são executados seqüencialmente. Inicialmente, tem-se uma alocação de dados e características de um cenário, em que são geradas as bateladas de produtos derivados do petróleo. Através dos dados gerados, é possível realizar um sequenciamento das bateladas através da otimização por um método metaheurístico, objeto de estudo deste trabalho. Assim, executa-se uma temporização através de uma otimização por um método exato, programação linear inteira mista.

Este trabalho apresenta um estudo comparativo dos resultados alcançados na implementação de modelos baseados em algumas técnicas metaheurísticas no problema de sequenciamento do *scheduling* das operações de transporte de derivados de petróleo. Pode-se verificar, através do processamento de casos reais, um bom desempenho dos modelos metaheurísticos. Foram obtidas soluções em curto tempo de processamento, e os ordenamentos resultantes, das bateladas, implicaram em reduções das violações, ou seja, melhoraram o atendimento ao cliente, que segundo especialistas é a meta principal de uma solução.

A apresentação de diferentes ordenações otimizadas das bateladas, permite a geração de soluções para o *scheduling* com diferentes características, como operar em um tempo reduzido, assim como obter o menor número de transições entre diferentes produtos, diminuindo a perda de interface, ou mesmo, optar por uma solução que apresente maiores taxas de utilização dos dutos. Isso representa diferentes opções de operações da malha ao especialista.

Dentre os modelos metaheurísticos estudados (GRASP, *Simulated Annealing*, Busca

Tabu, VNS, Colônia de Formigas, Algoritmo Genético e *Scatter Search*) observou-se um bom desempenho do GRASP, VNS e AG, quanto ao tempo de processamento e o melhor valor da função objetivo.

Com o intuito de uma obtenção de um melhor resultado de *scheduling* foram estudadas diferentes funções objetivos a serem otimizadas no sequenciamento de bateladas. Assim, adotou-se uma combinação de alguns objetivos. Isso resultou no interesse do estudo de técnicas de otimização multiobjetivo.

A aplicação do método de otimização multiobjetiva, MOGA, no sequenciamento de bateladas possibilitou a comparação com a metaheurística anteriormente analisada (AG). O modelo MOGA apresentou ser uma boa técnica pois permite o fornecimento de mais de uma solução otimizada, com características diversificadas, o que é muito importante devido à dinâmica do problema de *scheduling*. Embora a melhor solução final dos resultados analisados não tenha apresentado a mesma melhoria das soluções obtidas pelo modelo AG, o conjunto das melhores soluções apresentou características bastante diferenciadas. Por outro lado, as melhores soluções obtidas pelo modelo AG mostraram ser muito semelhantes.

Este trabalho contribuiu com o desenvolvimento de uma ferramenta de auxílio ao especialista que apresentou boas soluções para o problema do sequenciamento de bateladas, refletindo em um uso mais eficiente da malha dutoviária. A associação aos outros dois módulos, pré-processamento e temporização, tornou o processo de *scheduling* uma tarefa possível de se efetuar de forma sistêmica permitindo realizar uma análise global, do problema e de suas restrições, para futuros planejamentos.

7.2 CONTRIBUIÇÕES

Como principais contribuições deste trabalho tem-se:

- Caracterização do *scheduling* da rede de dutos.
- O estudo e apresentação de sistemas híbridos, técnicas metaheurísticas e técnicas multiobjetivo, assim como a descrição das seguintes metaheurísticas: GRASP, Simulated Annealing, Busca Tabu, VNS, Colônia de Formigas, Algoritmo Genético e Scatter Search, e a técnica multiobjetivo MOGA.
- A descrição de um problema real de *scheduling* operacional de distribuição de produtos derivados do petróleo, bem como sua estratégia de decomposição em subproblemas (alocação dos recursos, sequenciamento e temporização).
- A abordagem híbrida para o problema de *scheduling* operacional de uma rede de dutos.

- A formulação do modelo para o problema do sequenciamento de bateladas do problema de *scheduling* através da implementação das técnicas mencionadas no segundo item, e a comparação dos resultados alcançados.
- Análise dos resultados apresentados neste trabalho.
- Desenvolvimento do software de sequenciamento de bateladas para o *scheduling* da rede de dutos apresentada neste trabalho.

7.3 TRABALHOS FUTUROS

Como pontos de investigação futura, pode-se propor:

- A aplicação da técnica de otimização multiobjetivo, MOGA, neste trabalho apresentou ser muito importante na geração e apresentação de diferentes soluções ao especialista. Novos estudos e implementações de outras técnicas de otimização multiobjetivo poderiam ser realizadas na resolução do problema de sequenciamento das bateladas;
- Neste trabalho utilizou-se como objetivos a serem otimizados as janelas de tempo das bateladas, na origem e destino. Entretanto devido a as técnicas multiobjetivo possibilitarem o trabalho de inúmeras e diferentes funções de otimização, seria muito interessante a realização de um estudo que considerasse um maior número de objetivos;
- Analisar um maior número de testes com instâncias reais fornecidas pelas empresas responsáveis pelo transporte e distribuição dos produtos;
- Além disso seria interessante um estudo de outros possíveis dados que pudessem ser retirados da base de dados ou realizados pelo módulo de alocação dos recursos para que pudessem ser considerados pelo sequenciamento afim de melhorar os resultados.
- Possibilitar um "pós-processamento" das soluções do *scheduling* com a finalidade de introduzir algumas questões importantes pelos tomadores de decisão para uma melhora da solução.

REFERÊNCIAS

- ALLE, A. **Técnicas de programação mista-inteira aplicadas ao scheduling de plantas químicas contínuas**. São Paulo, Brasil: Tese de doutorado, Escola Politécnica da Universidade de São Paulo, 2003.
- ANP. **Agência Nacional do Petróleo**. 2006. Disponível em: <<http://www.anp.gov.br>>. Acesso em: 2006.
- AVANTHAY, C.; HERTZ, A.; ZUFFEREY, N. A variable neighborhood search for graph coloring. **European Journal of Operational Research**, v. 151, p. 379–388, 2003.
- BARBOSA, V. Redes neurais e simulated annealing como ferramentas para otimização combinatoria. **Investigación Operativa**, v. 1, p. 125–141, 1989.
- BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. **ACM Computing Surveys**, v. 35, n. 3, p. 268–308, 2003.
- BUZZO, W. R.; MOCCELLIN, J. V. Programação da produção em sistemas flow shop utilizando um método heurístico híbrido algoritmo genético simulated annealing. **Gestão e Produção**, v. 7, n. 3, p. 364–377, 2000.
- CAFARO, D. C.; CERDÁ, J. Optimal scheduling of multiproduct pipeline system using a non-discrete milp formulation. **Computers and Chemical Engineering**, v. 28, p. 2053–2068, 2004.
- CASTRO, H. P. **Utilização de Algoritmos Genéticos para Solução de Problema de Programação de Produção de uma Refinaria de Petróleo**. Florianópolis: Dissertação de mestrado, Universidade Federal de Santa Catarina, Departamento de Engenharia de Produção e Sistemas, 2001.
- CASTRO, R. E. **Otimização de Estruturas com Multi-objetivos Via Algoritmos Genéticos de Pareto**. Rio de Janeiro: Tese de Doutorado, Programa de Engenharia Civil, COPPE / UFRJ, 2001.
- COELLO, C. A. C. **An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design**. Estados Unidos: Tese de doutorado, Universidade de Tulane, Departamento de Ciência Computacional, 1996.
- COELLO, C. A. C. A comprehensive survey of evolutionary-based multiobjective optimization techniques. **Knowledge and Information Systems**, v. 1, p. 269–308, 1999.
- COELLO, C. A. C. An update survey of evolutionary multiobjective optimization techniques: State of the art and future trends. **Proceedings of the Congress on Evolutionary Computation**, Washington, v. 1, p. 3–13, 1999.
- COELLO, C. A. C. An updated survey of ga-based multiobjective optimization. **ACM Computing Surveys**, v. 32, n. 2, p. 109–143, 2000.

COELLO, C. A. C. 20 years of evolutionary multiobjective optimization: What has been done and what remains to be done. **Yen, G.Y., Fogel, D.B. , Computational Intelligence: Principles and Practice, IEEE Computational Intelligence Society**, p. 73–88, 2006.

CORREA, E. S. et al. A genetic algorithm for solving a capacitated p-median problem. **Numerical Algorithms**, v. 35, p. 373–388, 2004.

COWLING, P.; KENDALL, G.; HAN, L. An investigation of a hyperheuristic genetic algorithm applied to a trainer scheduling problem. **Proceedings of Congress on Evolutionary Computation**, p. 1185–1190, 2002.

CRANE, D. S.; WAINWRIGHT, R. L.; SCHOENEFELD, D. A. Scheduling of multi-product fungible liquid pipelines using genetic algorithms. **Symposium on Applied Computing Archive, Proceedings of the 1999 ACM symposium on Applied Computing**, San Antonio, Texas, United States, p. 280 – 285, 1999.

CRUZ, J. M. et al. Multiobjective optimization of the transport in oil pipelines networks. **IEEE International Conference on Emerging Technologies and Factory Automation**, v. 1, p. 566–573, 2003.

DEB, K. Evolutionary algorithms for multi-criterion optimization in engineering design. In: MIETTINEN, K. et al. (Ed.). **Evolutionary Algorithms in Engineering and Computer Science**. Chichester, UK: John Wiley & Sons, Ltd, 1999. p. 135–161.

DORIGO, M.; CARO, G. **The ant colony optimization meta-heuristic**. UK, England: McGraw-Hill's Advanced Topics In Computer Science Series archive, New ideas in optimization, McGraw-Hill Ltd., 1999. 11-32 p.

DORIGO, M.; CARO, G.; GAMBARDILLA, L. M. Ant algorithms for discrete optimization. **Artificial Life**, MIT Press, v. 5, p. 137–1721, 1999.

EGLESE, R. W. Simulated annealing: A tool for operational research. **European Journal of Operational Research**, v. 46, n. 3, p. 271–281, 1990.

FALCONE, M. A. G. **Estudo Comparativo entre Algoritmos Genéticos e Evolução Diferencial para Otimização de um Modelo de Cadeia de Suprimentos Simplificada**. Curitiba: Dissertação de mestrado, Pontifícia Universidade Católica do Paraná, Programa de Pós-Graduação de Engenharia de Produção e Sistemas, 2004.

FELIZARI, L. C. **Programação das Operações de Transporte de Derivados de Petróleo em Redes de Dutos**. Curitiba: Tese de Doutorado, apresentado ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná, 2009.

FERREIRA, P. A. V. **Otimização Multiobjetivo Teoria e Aplicações**. Campinas: Tese de Livre-Docência, Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Departamento de Telemática na Área de Pesquisa Operacional, 1999.

FONSECA, C. M.; FLEMING, P. J. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. **Genetic Algorithms: Proceedings of the Fifth International Conference**, 1990.

- FONSECA, C. M.; FLEMING, P. J. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In: **Genetic Algorithms: Proceedings (S. Forrest, ed.)**, CA: Morgan Kaufmann, 1993.
- FONSECA, C. M.; FLEMING, P. J. Multiobjective optimal controller design with genetic algorithms. In: **International Conference on Control**, v. 1, p. 745–749, 1994.
- FONSECA, C. M.; FLEMING, P. J. An overview of evolutionary algorithms in multiobjective optimization. **Evolutionary Computation**, v. 3, p. 1–16, 1995.
- GARCIA, J. M. C. et al. Hybrid heuristic and mathematical programming in oil pipelines networks. **IEEE Congress on Evolutionary Computation**, v. 2, p. 1479–1486, 2004.
- GENDREAU, M.; POTVIN, J. Y. Metaheuristics in combinatorial optimization. **Annals of Operational Research**, v. 140, p. 189–213, 2005.
- GLOVER, F. Tabu search - wellsprings and challenges. **European Journal of Operational Research**, v. 106, p. 221–225, 1998.
- GLOVER, F.; LAGUNA, M.; MARTÍ, R. Fundamentals of scatter search and path relinking. **Control and Cybernetics**, v. 39, n. 3, p. 653–684, 2000.
- GLOVER, F.; LAGUNA, M.; MARTÍ, R. **Scatter Search, Advances in Evolutionary Computation: Theory and Applications**. Nova York: Springer-Verlag, 2003. 519–537 p.
- GOLDBARG, M. C.; GOLDBARG, E. F. G. **Computação Evolucionária**. Brasil, setembro 2006.
- HAN, L.; KENDALL, G.; COWLING, P. An adaptive length chromosome hyperheuristic genetic algorithm for a trainer scheduling problem. **Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning**, 2002.
- HORN, J.; NAFPLIOTIS, N. **Multiobjective Optimization Using the Niche Pareto Genetic Algorithm**. Illinois, 1993.
- KONAK, A.; COIT, D. W.; SMITH, A. E. Multi-objective optimization using genetic algorithms: A tutorial. **Reliability Engineering and System Safety**, v. 91, p. 992–1007, 2006.
- KU, H.; KARIMI, I. An evaluation of simulated annealing for batch process scheduling. **Ind. Eng. Chem. Res.**, v. 30, 1991.
- LAGUNA, M. Scatter search. In: **Handbook of Applied Optimization**. Nova York: Oxford University Press, 2002. p. 183–193.
- LEE, H. C.; DAGLI, C. H. A. A parallel genetic-neuro scheduler for job-shop scheduling problems. **International Journal of Production Economics**, v. 51, p. 115–122, 1997.
- LÖHL, T.; SCHULZ, C.; ENGELL, S. Sequencing of batch operations for a highly coupled production process: Genetic algorithms versus mathematical programming. **Computers Chemical Engineering**, v. 22, p. 579–585, 1998.
- LOIOLA, E. M.; ABREU, N. M. M.; NETTO, P. O. B. Uma revisão comentada das abordagens do problema quadrático de alocação. **Pesquisa Operacional**, v. 24, n. 1, p. 73–109, 2004.

- MAGALHÃES, M. V. O. **Refinery Scheduling**. Londres, Inglaterra: PhD thesis, Imperial College London, 2004.
- MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs**. Nova York, Estados Unidos: Springer-Verlag Berlin Heidelberg, 1996.
- MITCHELL, M. **An introduction to genetic algorithms**. Estados Unidos: Massachusetts Institute of Technology, MIT Press, 1997.
- MLADENOVIC, N.; HANSEN, E. Variable neighbourhood search. **Computers and Operations Research**, v. 24, n. 1, p. 1097–1100, 1997.
- MOCKUS, L.; REKLAITIS, G. V. Continuous time representation in batch/ semicontinuous process scheduling: Randomized heuristics approach. **Computers Chemical Engineering**, v. 20, p. 1173–1177, 1996.
- MONTANA, D. et al. Genetic algorithms for complex, real-time scheduling. **IEEE International Conference on Systems, Man, and Cybernetics**, v. 3, p. 2213–2218, 1998.
- MORTON, T. E.; PENTICO, D. W. **Heuristic Scheduling Systems: with Applications to Production Systems and Project Management**. Estados Unidos: Wiley Interscience, 1993.
- NARA, K. Simulated annealing applications. In: SONG, Y. H. (Ed.). **Modern Optimisation Techniques in Power Systems**. Estados Unidos: Kluwer Academic Publishers, 1999. p. 15–38.
- NEVES-JR, F. et al. An efficient approach to the operational scheduling of a real-world pipeline network. **Proceedings of 17th European Symposium on Computer Aided Process Engineering - ESCAPE17**, Bucarest, Romênia, 2007.
- POLACEK, M. et al. **A Variable Neighbourhood Search for the Capacitated Arc Routing Problem with Intermediate Facilities**. 2006. Disponível em: <<http://amsacta.cib.unibo.it/archive/00002132/01>>. Acesso em: 20 de outubro de 2006.
- PRENTER, P. M. **Splines and Variational Methods**. Estados Unidos: John Wiley and Sons, 1975.
- PUCHINGER, J.; RAIDL, G. R. Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. **Proceedings of the First International Workshop on the Interplay Between Natural and Artificial Computation, Part II**, v. 3562, p. 41–53, 2005.
- RAIDL, G. R. A unified view on hybrid metaheuristics. In **Francisco Almeida et al., editors, Proceedings of the Hybrid Metaheuristics Workshop**, v. 4030, p. 1–12, 2006.
- RANGEL, M. C.; ABREU, N. M. M.; NETTO, P. O. B. Grasp para o pqa: Um limite de aceitação para soluções iniciais. **Pesquisa Operacional**, v. 20, n. 1, p. 45–58, 2000.
- REJOWSKI, R.; PINTO, J. M. Scheduling of a multiproduct pipeline system. **Computers and Chemical Engineering**, v. 27, p. 1229–1246, 2003.
- REKLAITIS, G. V. Overview of scheduling and planning of batch process operations. In **Proceedings of the NATO Advanced Study Institute on Batch Processing Systems**, Antalya, Turkey, p. 660–705, 1992.

- RELVAS, S.; BARBOSA-PÓVOA, A. P. F. D.; MATOS, H. A. Heuristic batch sequencing on a multiproduct oil distribution system. **Computers and Chemical Engineering**, v. 33, p. 712–730, 2009.
- RESENDE, M. G. C. Greedy randomized adaptive search procedures (grasp). In: **Encyclopedia of Optimization**. Estados Unidos: C. Floudas and P.M. Pardalos (eds.), Kluwer Academic Press, 2001. v. 2, p. 373–382.
- RESENDE, M. G. C.; RIBEIRO, C. C. Greedy randomized adaptive search procedures. In: **Handbook of Metaheuristics**. Estados Unidos: Glover, F. and Kochenberher, K. (eds.), Kluwer Academic Press, 2003. p. 219–249.
- SASIKUMAR, M. et al. Pipes: A heuristic search model for pipeline schedule generation. **Knowledge-Based Systems**, v. 10, p. 169–175, 1997.
- SILVER, E. A. **An Overview of Heuristic Solution Methods**. Canadá: Working Paper, Haskayne School of Business, University of Calgary, 2002.
- SOUZAFILHO, E. M.; FERREIRAFILHO, V. J. M.; LIMA, L. S. Variable neighborhood search (vns) aplicado ao problema de distribuição dutoviária com restrições de capacidade. **Anais do 4o PDPETRO 4o Congresso Brasileiro de P&D em Petróleo e Gás, Campinas, SP, Outubro, 2007**.
- STEBEL, S. L. **Técnicas de Otimização Aplicadas em Problemas de Scheduling dos Recursos de Estocagem**. Curitiba: Tese de doutorado, Universidade Tecnológica Federal do Paraná, Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, 2006.
- TALBI, E. G. A taxonomy of hybrid metaheuristics. **Journal of Combinatorial Optimization**, v. 8, p. 541–564, 2002.
- TANDON, M.; CUMMINGS, P. T.; LEVAN, M. D. Scheduling of multiple products units with tardiness penalties using simulated annealing. **Computers Chem. Engng.**, v. 19, p. 1069–1076, 1995.
- TANOMARU, J. Motivação, fundamentos e aplicações de algoritmos genéticos. **II Congresso Brasileiro de Redes Neurais, III Escola de Redes Neurais**, Curitiba, 1995.
- TAVARES, M. E. E. **Análise do Refino no Brasil**. Rio de Janeiro: Tese de Doutorado, Universidade Federal do Rio de Janeiro, COPPE, 2005.
- TRANSPETRO. **Mapa de Dutos**. 2009. Disponível em: <<http://www.transpetro.com.br>>. Acesso em: 17 de junho de 2009.
- TRIOLA, M. F. **Introdução à Estatística**. Rio de Janeiro: LTC, 1999.
- VELDHUIZEN, D. A. V. **Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations**. Ohio: Tese de doutorado, Air Force Institute of Technology, Air University, 1999.
- WESTPHAL, H. **Algoritmo Genético Aplicado a Otimização Multiobjetivo em Redes de Distribuição de Petróleos e Derivados**. Curitiba: Dissertação de mestrado, Universidade Tecnológica Federal do Paraná, Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, 2006.

YAMADA, T. **Studies on Metaheuristics for Jobshop and Flowshop Scheduling Problem**. Kyoto: Tese de doutorado, Universidade de Kyoto, 2003.

YAMAMOTO, L. **Uso de Simulated Annealing e Algoritmo Genético no Problema da Reconfiguração de uma Rede de Distribuição de Energia Elétrica**. Curitiba: Dissertação de Mestrado, Universidade Federal do Paraná, Programa de Pós-Graduação em Métodos Numéricos em Engenharia, 2004.

ZITZLER, E.; THIELE, L. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. **IEEE Transactions on Evolutionary Computation**, v. 3, n. 4, p. 257–271, 1999.