

HERBERT DE MÉLO DUARTE



**Um estudo algorítmico de problemas logísticos
na indústria de petróleo e gás natural**

NATAL
2006

HERBERT DE MÉLO DUARTE



**Um estudo algorítmico de problemas logísticos
na indústria de petróleo e gás natural**

Dissertação de Mestrado apresentada como
pré-requisito parcial para obtenção do título de
Mestre em Sistemas e Computação pela
Universidade Federal do Rio Grande do Norte.
Orientador: Prof. Dr. Marco César Goldberg

NATAL
2006

Divisão de Serviços Técnicos

Catálogo da Publicação na Fonte. UFRN / Biblioteca Central Zila Mamede

Duarte, Herbert de Mélo

Um estudo algorítmico de problemas logísticos na indústria de petróleo e gás natural / Herbert de Mélo Duarte. – Natal, RN, 2006. 121 f.

Orientador: Marco César Goldberg.

Dissertação (Mestrado) – Universidade Federal do Rio Grande do Norte. Centro de Ciências Exatas e da Terra. Programa de Pós-Graduação em Sistemas e Computação.

1. Algoritmos – Dissertação. 2. Metaheurísticas – Dissertação. 3. Roteamento – Dissertação. 4. Pistoneio – Dissertação. 5. Dimensionamento de dutos – Dissertação. I. Goldberg, Marco César. II. Universidade Federal do Rio Grande do Norte. III. Título.

RN/UF/BCZM

CDU 004.421 (043.3)

HERBERT DE MÉLO DUARTE

**Um estudo algorítmico de problemas logísticos
na indústria de petróleo e gás natural**

Esta Dissertação de Mestrado foi julgada adequada e aprovada para o cumprimento de requisito parcial para a obtenção do grau de **Mestre em Sistemas e Computação** pela **Universidade Federal do Rio Grande do Norte**.

Natal, 16 de novembro de 2006.

Banca Examinadora

Prof. Dr. Marco Cesar Goldberg
Orientador

Profa. Dra. Elizabeth Ferreira Gouvêa Goldberg
Membro

Profa. Dra. Maria Teresinha Arns Steiner
Membro

Dedicatória

À minha esposa, Tânia Venditelli Soares Duarte, pelo apoio e incentivo, pelo amor e compreensão.

Aos meus pais, Heliú Bezerra Duarte e Jeanne Almeida de Mélo Duarte, pelos exemplos de vida e por me possibilitarem as condições para esta conquista.

Agradecimentos

A Deus, por me abençoar com grandes oportunidades, infinitamente maiores do que possa merecer, e por me conduzir nos seus caminhos.

À minha esposa, que muito se empenhou para estar sempre ao meu lado e me dar forças para concluir este trabalho. Eu te amo!

A meus pais, por me transmitir as bênçãos de Deus em forma de ensinamentos valiosos e condições de cursar esta pós-graduação. Muito obrigado!

A meus irmãos, Hélder de Mélo Duarte e Helton de Melo Duarte, amigos e companheiros de toda hora. Tenham muito sucesso!

Aos professores Marco Cesar Goldbarg, meu orientador, e Elizabeth Ferreira Gouvêa Goldbarg, “co-orientadora”, por me acrescentar tanto conhecimento, por me apresentar o instigante mundo da pesquisa científica e me desafiar continuamente a superar barreiras que, sozinho, não teria sequer conhecido, quem dera ultrapassado.

Ao professor David Boris Paul Dehárbe, pelas aulas no mestrado e, principalmente, pelo incentivo e treinamento na Maratona de Programação.

Aos meus amigos da maratona, em especial Daniel “Nazário”, Diego Caminha e o “coach” Sérgio Queiroz, por me ajudarem a aprender muito sobre algoritmos e entidades assemelhadas.

Aos citados no último parágrafo e demais amigos do LAE, Ceará, Jones, Givanaldo e JP Farias, pela colaboração nos estudos, pesquisas e risadas.

À Agência Nacional do Petróleo, através do Programa de Recursos Humanos para o Setor de Petróleo e Gás (PRH-22), pela bolsa de estudo e apoio financeiro que foram fundamentais para a conclusão deste curso e certamente também para o de muitos outros estudantes em todo o país.

Resumo

Duarte, Herbert de Mélo. **Um estudo algorítmico de problemas logísticos na indústria de petróleo e gás natural**. 2006. 121p. Dissertação (Mestrado em Sistemas e Computação) – Programa de Pós-Graduação em Sistemas e Computação, Universidade Federal do Rio Grande do Norte, Natal, 2006.

Este trabalho consiste do estudo de dois importantes problemas oriundos das operações das indústrias de petróleo e gás natural. O primeiro problema – do dimensionamento de dutos em uma rede urbana de distribuição de gás natural – consiste em encontrar a combinação de diâmetros de menor custo, a partir de um conjunto de opções comercialmente disponíveis, para os dutos de uma dada rede de distribuição de gás, de forma a respeitar requisitos de pressão mínima em cada nó de demanda e condições de *upstream*. Por sua vez, o segundo problema – do roteamento da unidade móvel do pistoneio – decorre da necessidade de se definir as rotas de visitação da dita unidade móvel do pistoneio aos diversos poços não surgentes do campo de exploração, ou seja, poços que não possuem pressão suficiente para fazer o óleo emergir à superfície. A versão periódica do problema leva em consideração a equação de re-enchimento dos poços, de forma a possibilitar um planejamento mais acurado num horizonte de tempo maior. Além da formulação matemática dos dois problemas, para a solução do primeiro foram desenvolvidos um algoritmo exato e uma busca tabu e para o segundo, um limite superior e um algoritmo transgenético ProtoGene. Os principais conceitos das metaheurísticas são apresentados, juntamente com os detalhes da aplicação destas aos problemas citados. Os resultados obtidos para ambas as aplicações são promissores quando comparados com limites teóricos e soluções alternativas, tanto relativamente à qualidade das soluções como ao tempo computacional envolvido.

Palavras-chave: Petróleo, unidade móvel do pistoneio, roteamento de veículos, gás natural, dimensionamento de dutos, metaheurísticas.

Abstract

Duarte, Herbert de Mélo. **An algorithmic study of logistic problems on petroleum and natural gas industry**. 2006. 121p. Dissertation (Master's Degree in Systems and Computer Science) – Program of Graduation in Systems and Computer Science, Federal University of Rio Grande do Norte, Natal, 2006.

This work consists on the study of two important problems arising from the operations of petroleum and natural gas industries. The first problem – the pipe dimensioning problem on constrained gas distribution networks – consists in finding the least cost combination of diameters from a discrete set of commercially available ones for the pipes of a given gas network, such that it respects minimum pressure requirements at each demand node and upstream pipe conditions. On its turn, the second problem – the piston pump unit routing problem – comes from the need of defining the piston pump unit routes for visiting a number of non-emergent wells in on-shore fields, i.e., wells which don't have enough pressure to make the oil emerge to surface. The periodic version of this problem takes into account the wells re-filling equation to provide a more accurate planning in the long term. Besides the mathematical formulation of both problems, an exact algorithm and a taboo search were developed for the solution of the first problem and a theoretical limit and a ProtoGene transgenetic algorithm were developed for the solution of the second problem. The main concepts of the metaheuristics are presented along with the details of their application to the cited problems. The obtained results for both applications are promising when compared to theoretical limits and alternate solutions, either relative to the quality of the solutions or to associated running time.

Keywords: Petroleum, piston pump unit, vehicle routing, natural gas, pipe dimensioning, metaheuristics.

Lista de Figuras

Figura 1: Rede de coleta de gás com poços <i>offshore</i>	29
Figura 2: (a) Dutos em paralelo (b) Dutos em série.....	30
Figura 3: (a) Rede antes do movimento (b) Rede após o movimento	58
Figura 4: Evolução transgenética	64
Figura 5: Representação de uma solução do PPPP	67
Figura 6: Variação do tempo de execução do algoritmo exato com o tamanho da instância	75
Figura 7: Variação do tempo de execução do algoritmo exato com o número de atribuições parciais.....	75
Figura 8: Número de atribuições parciais por iteração para instância <i>rd400</i>	76
Figura 9: Número de atribuições parciais por iteração para instância <i>d2103</i>	76
Figura 10: Gap médio da busca tabu de Cunha e Ribeiro para a instância <i>pr439</i> ...	78
Figura 11: Gap médio da busca tabu de Cunha e Ribeiro para a instância <i>p654</i>	78
Figura 12: Gap médio da busca tabu proposta para a instância <i>pr439</i>	79
Figura 13: Gap médio da busca tabu proposta para a instância <i>p654</i>	79
Figura 14: Evolução do algoritmo transgenético para a instância <i>ftv100</i>	90
Figura 15: Evolução do algoritmo transgenético para a instância <i>ry48</i>	91

Lista de Quadros

Quadro 1: Pseudo-código do algoritmo exato para o PDD.....	39
Quadro 2: Algoritmo para o problema da mochila 0-1	52
Quadro 3: Algoritmo para a geração do limite superior para o PPPP.....	54
Quadro 4: Pseudo-código da busca tabu proposta para o PDD.....	60
Quadro 5: Esquema básico de um algoritmo ProtoG	65
Quadro 6: Pseudo-código do algoritmo transgenético aplicado ao PPPP.....	71

Lista de Tabelas

Tabela 1: Listas de custos e quedas de pressão dos dutos d_1 , d_2 e d_3	32
Tabela 2: Quadro de cálculos inicial da fusão em paralelo.....	32
Tabela 3: Segundo quadro de cálculos da fusão em paralelo	33
Tabela 4: Lista de atribuições parciais equivalente aos dutos d_1 , d_2 e d_3	33
Tabela 5: Terceiro quadro de cálculos da fusão em paralelo	34
Tabela 6: Nova lista de atribuições parciais equivalente aos dutos d_1 , d_2 e d_3	34
Tabela 7: Quarto quadro de cálculos da fusão em paralelo.....	34
Tabela 8: Quadros de cálculo restantes da fusão em paralelo	35
Tabela 9: Nova lista de atribuições parciais equivalente aos dutos d_1 , d_2 e d_3	35
Tabela 10: Lista de custos e quedas de pressão do duto d_4	36
Tabela 11: Quadro de cálculos inicial da fusão em série	36
Tabela 12: Segundo quadro de cálculos da fusão em série	37
Tabela 13: Terceiro quadro de cálculos da fusão em série	37
Tabela 14: Lista de atribuições parciais equivalente aos dutos d_e e d_4	37
Tabela 15: Lista de atribuições parciais equivalente aos dutos d_e e d_4 após redução	37
Tabela 16: Alguns procedimentos típicos dos vetores transgenéticos.....	63
Tabela 17: Resultados do algoritmo exato para o PDD	74
Tabela 18: Estudo dos critérios de parada para a busca tabu comparada	81
Tabela 19: Comparação da qualidade das soluções para 34 instâncias pequenas e médias do PDD	83
Tabela 20: Comparação da qualidade das soluções para 10 instâncias grandes do PDD.....	84
Tabela 21: Tempo médio (em segundos) de execução dos algoritmos para o PDD.....	85
Tabela 22: Volumes máximos e médios para as diferentes populações iniciais.....	88
Tabela 23: Volumes máximos para os vários tamanhos da população	89
Tabela 24: Volumes médios para os vários tamanhos da população	89
Tabela 25: Volumes máximos para os vários tamanhos do banco de informações.....	89
Tabela 26: Volumes médios para os vários tamanhos do banco de informações	89

Tabela 27: Resultados obtidos pelo algoritmo transgenético ProtoG	92
--	----

Lista de Abreviações

ADN	Ácido Desoxirribonucléico
AG-S	Algoritmo genético multi-objetivo de Surry et al. (1995)
ANP	Agência Nacional do Petróleo
BCP	Bombeio por Cavidades Progressivas
BCS	Bombeio Centrífugo Submerso
BMH	Bombeio Mecânico com Hastes
BT-1	Primeira versão da busca tabu de Cunha e Ribeiro (2004)
BT-2	Segunda versão da busca tabu de Cunha e Ribeiro (2004)
BT-C	Versão da busca tabu de Cunha e Ribeiro (2004) utilizada para comparações
BT-P	Algoritmo busca tabu proposto neste trabalho
COMOGA	<i>Constrained Optimization by Multi-Objective Genetic Algorithms</i>
DFJ	Formulação de Dantzig-Fulkerson-Johnson para o PCV
ETO	Estação de Tratamento de Óleo
GLC	<i>Gas-Lift</i> Contínuo
GLI	<i>Gas-Lift</i> Intermitente
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
PCV	Problema do Caixeiro Viajante
PCVA	Problema do Caixeiro Viajante Assimétrico
PDD	Problema do Dimensionamento de Dutos
PPP	Problema do Passeio do Pistoneio
PPPP	Problema do Passeio do Pistoneio Periódico
PRH	Programa de Recursos Humanos para o Setor de Petróleo e Gás
PRV	Problema do Roteamento de Veículos
PRVP	Problema do Roteamento de Veículos Periódico
PRVS	Problema do Roteamento de Veículos Seletivo
UMP	Unidade Móvel do Pistoneio
UN-RNCE	Unidade de Negócios de Exploração e Produção do Rio Grande do Norte e Ceará
VND	Busca em Vizinhança Variável Descendente

Sumário

1	Introdução.....	15
1.1	Objetivos do Trabalho.....	18
1.2	Estrutura do Trabalho.....	19
2	O Problema do Dimensionamento de Dutos.....	20
2.1	Descrição do Problema.....	21
2.2	Revisão da Literatura.....	22
2.3	Formulação Matemática.....	27
2.4	Algoritmo Exato.....	28
2.4.1	Fusão em Paralelo.....	31
2.4.2	Fusão em Série.....	35
2.4.3	Pseudo-código.....	38
2.4.4	Corretude e Complexidade.....	39
3	O Problema do Passeio do Pistoneio Periódico.....	41
3.1.	Descrição do Problema.....	41
3.2.	Revisão da Literatura.....	43
3.3	Formulação Matemática.....	47
3.4	Limite Superior.....	50
4	Busca Tabu.....	55
4.1	Implementação.....	56
4.1.1	Solução Inicial.....	57
4.1.2	Exploração da Vizinhança.....	57
4.1.3	Procedimento de Diversificação.....	59
4.1.4	Movimentos Tabus e Critério de Aspiração.....	59
4.1.5	Critério de Parada.....	60
4.1.6	Pseudo-código.....	60
5	Transgenética Computacional.....	61
5.1	Implementação.....	66
5.1.1	Atratividade de um Nó.....	66

5.1.2 Representação das Soluções	67
5.1.3 População Inicial	67
5.1.4 Banco de Informações	68
5.1.5 Plasmídeos	68
5.1.6 Critérios de Parada	70
5.1.7 Pseudo-código	70
6 Experimentos Computacionais	72
6.1 Problema do Dimensionamento de Dutos	72
6.1.1 Instâncias	72
6.1.2 Algoritmo Exato	72
6.1.3 Estudo dos Parâmetros	77
6.1.4 Experimentos Finais para o PDD	82
6.2 Problema do Passeio do Pistoneio Periódico	86
6.2.1 Instâncias	86
6.2.2 Estudo dos Parâmetros	87
6.2.3 Experimentos Finais para o PPPP	91
7 Conclusões e Sugestões para Trabalhos Futuros	94
Referências Bibliográficas	97
Apêndice 1: Soluções Ótimas para o PDD	104

1 Introdução

A atual conjuntura econômica mundial, fruto de uma economia globalizada e altamente competitiva, tem exigido das corporações uma excelência na gestão de seus recursos e processos, de forma a garantir altos índices de produtividade, que culminem em agregar valor ao produto ou serviço oferecido. Desta forma, empresas têm envidado esforços no sentido de otimizar seus processos produtivos com o objetivo de obter o melhor proveito dos recursos disponíveis. Nesta direção, as organizações tem se valido de ferramentas oriundas da Pesquisa Operacional e de métodos modernos de solução de problemas de Otimização Combinatória. Assim, o conhecimento especializado e restrito apenas à teoria em áreas como a Matemática e a Ciência da Computação tem se difundido para uma aplicação prática por diversas organizações.

Para Aarts e Korst (1989), um problema de otimização combinatória é especificado por um conjunto de instâncias, podendo ser de maximização ou de minimização. Uma instância de um problema de otimização combinatória pode ser definida como uma tupla (S, F) , onde S é o espaço de soluções, um conjunto finito de todas as soluções possíveis, e F é uma função de custo da solução, definida do espaço de soluções nos números reais. Em um problema de minimização, deseja-se encontrar uma solução $s_{opt} \in S$ que satisfaça a inequação $\forall s \in S, F(s_{opt}) \leq F(s)$. Já quando se trata de um problema de maximização, o objetivo é achar uma solução $s_{opt} \in S$ que satisfaça a inequação $\forall s \in S, F(s_{opt}) \geq F(s)$. Uma solução $s_{opt} \in S$ é denominada solução ótima global, mínima ou máxima, $F_{opt} = F(s_{opt})$ é o chamado custo ótimo para a instância e S_{opt} denota o conjunto de todas as possíveis soluções ótimas globais.

Um grupo importante de problemas de otimização é a chamada classe de problemas NP-árduos (CAMPELLO; MACULAN, 1994). Para esta classe de problemas não são conhecidos algoritmos de solução eficientes, ou seja, algoritmos com complexidade de tempo polinomial no tamanho da entrada. Diversos são os problemas comprovadamente NP-árduos, vários dos quais possuem uma forte relação com outras áreas de pesquisa e aplicações importantes na indústria. Como exemplos, pode-se citar o problema do caixeiro viajante (PCV), o problema geral do

roteamento de veículos, o problema do particionamento, o problema do recobrimento, o problema do número cromático em grafos, o problema da cobertura e conjunto independente de nós, o problema da mochila, etc. Maiores referências são dadas por Campello e Maculan (1994), Goldberg e Luna (2000), e Wilf (1986), entre outros.

Tendo em vista esta dificuldade inerente à resolução de variados e importantes problemas de otimização combinatória, foi proposto na literatura o desenvolvimento de algoritmos que fornecessem boas soluções em um tempo aceitável para os propósitos práticos, ou seja, soluções aproximativas de boa qualidade, mas que não necessariamente representam a solução ótima para uma dada instância do problema. Em muitos casos, não há nem mesmo uma garantia sobre o quão perto ou longe da solução ótima está a solução fornecida pelo algoritmo. Estes algoritmos foram denominados heurísticos e, posteriormente, os paradigmas ou estratégias gerais para o desenvolvimento desta classe de algoritmos foram denominados de metaheurísticas. Uma metaheurística, portanto, é uma estratégia genérica que pode ser utilizada no desenvolvimento de algoritmos aproximativos para uma variedade de problemas distintos.

Uma primeira categoria de heurísticas a aparecer na literatura foram as heurísticas míopes ou gulosas. Esta classe de algoritmos pode ser aplicada a qualquer problema cuja estrutura se caracterize como um sistema de independência. Os algoritmos míopes consideram, a cada iteração ou etapa de decisão, a melhor alternativa segundo um critério de otimização meramente local. Contudo, a otimização local de cada etapa não garante o alcance de uma solução ótima global para muitos dos problemas de interesse, mas representa um procedimento aproximativo de aplicação viável do ponto de vista computacional. Para certos problemas, aqueles que apresentam uma estrutura matemática de matróide (EDMONDS, 1971), existem algoritmos míopes que levam à solução ótima para todas as instâncias (CAMPELLO; MACULAN, 1994; GOLDBARG; LUNA, 2000).

Muitas foram as abordagens metaheurísticas propostas na literatura ao longo das últimas décadas, das quais pode-se destacar algumas bem conhecidas:

- Métodos de busca local (AARTS; LENSTRA, 1997), que atuam iterativamente sobre uma solução corrente, escolhendo a cada passo do algoritmo, segundo algum critério pré-definido, uma solução de sua vizinhança para substituí-

la, até que uma condição de parada seja satisfeita (normalmente quando a solução atual é um ótimo local);

- GRASP – *Greedy Randomized Adaptive Search Procedure* – (FEO; RESENDE, 1995), que agrega um componente aleatório à estratégia gulosa, para a geração de soluções viáveis distintas que são intensificadas iterativamente através de uma etapa de busca local;

- Busca em vizinhança variável (MLADENOVIC; HANSEN, 1997), na qual são utilizadas diferentes estruturas de vizinhança numa busca local, de forma a superar as limitações de cada uma das estruturas individuais e promover diversificação das soluções;

- *Simulated Annealing* (KIRKPATRICK et al., 1983), que explora uma analogia com o processo físico do recozimento de sólidos para a exploração do espaço de soluções – normalmente aplicada em forma de uma busca local que permite probabilisticamente a geração de soluções de qualidade inferior à solução corrente para escapar de ótimos locais;

- Busca Tabu (GLOVER, 1987), que procura escapar dos ótimos locais pela manutenção de uma lista de movimentos proibidos, ou seja, movimentos que não devem ser realizados por poder levar a soluções já visitadas anteriormente;

- Colônia de formigas (DORIGO et al., 1996), uma metaheurística que baseia-se no comportamento das formigas no processo de busca por comida; e

- Algoritmos genéticos (HOLLAND, 1975), *scatter search* (GLOVER, 1977), algoritmos culturais (COELLO; BECERRA, 2004), algoritmos meméticos (MOSCATO, 1989) e algoritmos transgenéticos (GOUVÊA, 2001), que juntos constituem a computação evolucionária e se caracterizam por se inspirar em processos da evolução biológica e cultural dos seres vivos.

Todas estas metaheurísticas têm demonstrado sucesso em diferentes problemas e muito esforço de pesquisa tem sido empregado neste ramo do conhecimento. Uma diferenciação pode ser feita entre métodos de busca local, voltados a explorar mais intensamente um determinado subconjunto do espaço de soluções, e métodos de busca global, que procuram visitar diferentes regiões do espaço de busca, ou seja, procuram diversificar as soluções consideradas. A união das características destes dois tipos de métodos tem sido proposta como uma alternativa de sucesso por diversas das metaheurísticas citadas.

1.1 Objetivos do Trabalho

A condição de extrema competitividade e de influências diretas entre os diversos mercados mundiais tem sido notada particularmente verdadeira nos setores base da economia, incluindo o setor energético. Estes setores são responsáveis por produtos e serviços que são ao mesmo tempo *commodities* e estratégicos. O setor energético é ainda influenciado por uma variedade de fatores, políticos e econômicos, que provocam uma grande instabilidade no valor de mercado de seus produtos finais. Neste sentido, empresas que exploram o petróleo e o gás natural, duas importantes fontes de energia nos dias de hoje, têm seguido a tendência mundial e procurado resolver os problemas de otimização de seus processos como forma de manter-se capaz de atuar neste mercado global.

No contexto das atividades das indústrias de extração e distribuição de petróleo e gás natural, dois relevantes problemas enfrentados relacionam-se com o setor logístico de suas operações, ou seja, com a movimentação ou distribuição de suas matérias-primas ou seus produtos acabados. São eles: o problema do dimensionamento de dutos em uma rede urbana de distribuição de gás natural e o problema do passeio da unidade móvel do pistoneio periódico. Neste sentido, o presente trabalho tem como objeto de estudo a modelagem matemática e resolução computacional dos referidos problemas, através de algoritmos exatos e das já citadas abordagens metaheurísticas.

O problema do dimensionamento de dutos (PDD) resume-se em, dado um traçado de uma rede urbana de distribuição de gás natural e as pressões requeridas por cada cliente ou nó de demanda, determinar o diâmetro ótimo dos dutos de cada seção da rede – a partir de um conjunto discreto de diâmetros comercialmente disponíveis – de forma a garantir o suprimento das pressões desejadas ao menor custo de instalação possível para a estrutura de distribuição.

Por sua vez, o problema do passeio do pistoneio periódico (PPPP) consiste em delinear em um campo terrestre de extração de petróleo, uma seqüência de poços de petróleo a serem pistoneados em cada dia do horizonte de planejamento desejado, respeitando restrições como o tempo máximo diário de operação da unidade móvel do pistoneio e de forma que o volume de óleo coletado seja maximizado.

1.2 Estrutura do Trabalho

O texto do trabalho está estruturado como segue: os dois próximos capítulos detalham o problema do dimensionamento de dutos em uma rede urbana de distribuição de gás natural e o problema do passeio da unidade móvel do pistoneio periódico, respectivamente. Nos dois casos, são apresentadas uma revisão da literatura relevante ao estudo dos problemas e uma formulação matemática dos mesmos. Ainda, são descritos um algoritmo exato para a solução do primeiro problema e um limite superior teórico para o segundo.

O capítulo 4 descreve em detalhes a metaheurística Busca Tabu, especificando cada elemento de seu funcionamento e os detalhes envolvidos na sua aplicação ao problema do dimensionamento de dutos.

O capítulo 5, por sua vez, explica a metáfora da transgenética computacional e como a mesma pode ser aplicada na resolução de problemas de otimização combinatória. Neste capítulo, também, é detalhada a construção de um algoritmo ProtoGene para o dito problema do passeio do pistoneio periódico.

O capítulo 6 mostra os resultados obtidos pelos experimentos computacionais para os dois problemas, realizando comparações com limites teóricos e soluções alternativas que permitem verificar a utilidade e eficiência das abordagens propostas.

Finalmente, o capítulo 7 sintetiza alguns pontos de relevância levantados neste trabalho, suas principais contribuições, e aponta direções para trabalhos futuros.

2 O Problema do Dimensionamento de Dutos

O gás natural é uma fonte energética primária que ocupa lugar de destaque na matriz energética mundial. O mesmo é um combustível fóssil, constituído essencialmente de uma mistura de hidrocarbonetos leves – gases inorgânicos e hidrocarbonetos saturados, em especial o metano e o etano, mas também o propano e o butano, entre outros –, sendo encontrado em rochas porosas no subsolo, associado ou não à reservas de petróleo. O gás natural pode ser transformado em energia térmica, elétrica e mecânica, sendo muito competitivo em relação aos demais tipos de combustível, principalmente por sua queima eficiente e limpa e por ser seguro, de fácil manuseio e com reduzida emissão de gases poluentes. Além disto, o mesmo não necessita de armazenagem, bombeamento ou preparação para utilização, resultando em custos reduzidos para as operações que o aproveitam. Também por possuir um elevado rendimento térmico, não necessita de tratamento para os gases resultantes de sua combustão e prolonga o tempo de vida dos equipamentos que o utilizam. Por estes e outros motivos, esta fonte de energia tem sido progressivamente mais utilizada, destacadamente na indústria petroquímica, e a sua grande disponibilidade indica que será ainda mais consumida durante os próximos anos (ANP, 2006).

A via mais eficiente de distribuição do gás natural é através de gasodutos. Desta forma, redes de distribuição de gás natural tornaram-se uma parte importante da infra-estrutura urbana, servindo tanto a residências como a estabelecimentos comerciais e industriais, e normalmente implicam num enorme custo de implantação. Por estas razões, muitas empresas que provêem este tipo de serviço necessitam utilizar-se de ferramentas de apoio à decisão na tentativa de sintetizar o melhor projeto destas redes. O projeto real de uma rede de distribuição de gás natural usualmente envolve duas etapas. A primeira delas é a definição do *layout* ou traçado da rede, considerando os nós de demanda a serem atendidos. O traçado desta é geralmente determinado na prática por considerações sobre as rotas das estradas e questões de topologia. Quando esta fase é terminada, o tipo e o diâmetro de cada duto a ser assentado devem ser determinados de forma que a pressão mínima requerida esteja disponível em cada nó de demanda. Nas duas etapas, o fator

econômico é bastante importante, e a minimização dos custos de instalação da rede é normalmente um fator de caráter essencial. O presente trabalho preocupa-se com a solução da segunda etapa deste planejamento, e o problema relacionado é descrito a seguir.

2.1 Descrição do Problema

Os dutos para o transporte de gás natural são produzidos e comercializados somente em certo número de materiais e em certos diâmetros fixos. Seus custos por unidade de comprimento normalmente variam com o tipo de material de que são feitos e com o seu diâmetro. Para maiores diâmetros, são esperados dutos mais caros. A pressão também cai ao longo dos dutos numa taxa decrescente com o diâmetro do duto para um fluxo constante, o que significa que diâmetros maiores implicam em redes mais confiáveis, ou seja, com uma maior garantia de que a pressão de demanda poderá ser satisfatoriamente suprida.

O problema estudado, por considerações práticas, presume um traçado pré-definido da rede. Isto implica que só se investiga a segunda etapa do planejamento de uma rede de distribuição de gás natural, só se busca selecionar os diâmetros dos dutos que vão compor a rede, de forma que eles sejam largos o bastante para garantir os requisitos de pressão mínima do projeto a cada nó de demanda, ao menor custo possível. Uma restrição de projeto adicional é que cada duto não incidente a um nó fonte deve ter pelo menos um duto de *upstream* de diâmetro igual ou maior que o seu. A função-objetivo considerada é, então, minimizar a soma dos custos de cada duto da rede. Neste trabalho, são consideradas redes de distribuição com o menor número possível de trechos de dutos, ou seja, com um traçado em forma de uma estrutura de árvore, mas o método de solução proposto também pode se aplicar a redes de distribuição com traçado incluindo malhas. O termo duto é usado para um arco da árvore.

O problema em questão é doravante denominado problema do dimensionamento de dutos (PDD) e é formulado como um problema de otimização combinatória com restrições, mas, uma vez que suas restrições são implícitas e observam equações não-lineares, a viabilidade de uma solução só pode ser

determinada através da solução das equações de fluxo da rede. Por estas razões, uma investigação analítica não parece ser promissora para a solução deste tipo de problema e o uso de metaheurísticas é a alternativa natural para lidar com o mesmo. Apesar do fato de que a solução final (atribuição de diâmetros aos dutos) a ser implantada é normalmente determinada pelo projetista da rede e pode ser influenciada por outros fatores, como, por exemplo, pela expectativa de crescimento da demanda, o fator econômico é claramente um dos mais importantes.

A seguir, apresenta-se uma revisão de alguns trabalhos anteriores em problemas relacionados à otimização de redes de distribuição encontrados na literatura. Uma formulação matemática do problema do dimensionamento de dutos como um problema de otimização não-linear inteira mista é dada adiante. O capítulo termina com a apresentação de um algoritmo exato desenvolvido para o referido problema baseado na técnica apresentada no trabalho de Rothfarb *et al.* (1970).

2.2 Revisão da Literatura

Em um dos primeiros trabalhos publicados sobre um problema relacionado, Rothfarb *et al.* (1970) exploraram, entre outros aspectos de projeto de redes de distribuição de gás de campos marítimos para plantas de separação, o problema de seleção dos diâmetros dos dutos, considerando a soma dos custos de investimento e operação como o critério de minimização. Nesse artigo, as redes consideradas são aquelas que contêm o menor número de seções de dutos que pode levar o gás dos campos às plantas de separação ou, em outras palavras, redes com uma estrutura fixa de árvores. Para tratar com o crescimento exponencial do número de soluções como uma função do número de ramos da árvore, é introduzida uma técnica que usa duas operações, fusão serial e paralela, para sucessivamente reduzir o traçado de árvore da rede a traçados menores equivalentes com uma atribuição parcial de diâmetros. Com esta abordagem, foi possível resolver de forma exata uma instância com 20 nós. A abordagem de Rothfarb *et al.* (1970) é melhor explicada posteriormente neste capítulo, quando detalha-se o algoritmo exato desenvolvido para a solução do problema de dimensionamento de dutos estudado neste trabalho.

Anos depois, Boyd *et al.* (1994) desenvolveram um algoritmo genético para o problema do dimensionamento de dutos e usaram uma função de penalidade para levar em consideração tanto as restrições de pressão mínima como de dutos de *upstream*. As soluções são representadas como uma seqüência de n inteiros, onde n é o número de dutos da rede, cada inteiro indicando o índice do diâmetro a ser escolhido para uma dada seção de dutos. O índice de um duto (i, j) – com o gás fluindo de i para j , ou seja, quando o nó i está no único caminho entre o nó j e a raiz – é igual ao valor j . Em outras palavras, rotula-se cada duto ou arco da árvore com o mesmo rótulo do nó para o qual aquele duto entrega o gás diretamente. Esta representação permite que todo o espaço de busca seja codificado e facilita o uso de operadores genéticos, como a mutação *creep* e o cruzamento uniforme, ambos utilizados no algoritmo genético citado. Pela facilidade de manipulação das soluções com esta representação, a mesma foi utilizada em todos os algoritmos implementados neste trabalho. Uma população não-estruturada de 100 redes é usada durante 100 gerações e permite aos autores relatarem resultados 4% melhores que uma heurística de busca local utilizada pela *British Gas* para uma instância real de 25 dutos. A heurística comparada é simples e serve como uma idéia base para outros métodos de solução deste problema. Ela consiste em gerar alguns diâmetros iniciais para os dutos que levem a uma rede viável e, em seguida, otimizar localmente a solução corrente tentando repetidamente reduzir o diâmetro de um dos dutos até que nenhuma redução adicional possa gerar uma solução ainda viável.

Extendendo esta idéia, o método COMOGA (*Constrained Optimization by Multi-Objective Genetic Algorithms* – Otimização com Restrições através de Algoritmos Genéticos Multi-Objetivo) é introduzido por Surry *et al.* (1995), que trata as restrições do problema, sejam explícitas ou implícitas, como um critério separado em uma formulação multi-objetivo do problema. O método tem a vantagem de diminuir o número de parâmetros livres a serem ajustados quando comparado com a formulação tradicional de função-objetivo com penalidades e de permanecer pouco sensível a esses.

Ele consiste em reduzir o problema a um de bi-critério, condensando todas as restrições em um único critério ao calcular um *ranking* de Pareto para cada solução da população (número de soluções que as dominam) com respeito ao nível de violação das restrições originais. O método então tenta minimizar tanto o custo

quanto o *ranking* de Pareto através de um esquema *annealing* reverso que ajusta a probabilidade de selecionar e substituir indivíduos da população de acordo com um ou outro critério, em uma seleção por torneio binário. O COMOGA ajusta a probabilidade de escolher indivíduos para recombinação baseado no custo da solução ou no *ranking* de Pareto das violações de restrição se a proporção de soluções viáveis na população não está próxima o suficiente a um parâmetro alvo, τ .

O algoritmo representa as soluções como uma seqüência de n inteiros, onde n é o tamanho da instância, indicando o índice do diâmetro para cada duto da instância, de forma idêntica ao do algoritmo proposto por Boyd *et al.* (1994). A população é não-estruturada e composta de 100 indivíduos. O mesmo utiliza uma mutação aleatória (na qual um alelo é substituído aleatoriamente por algum valor válido de diâmetro, podendo manter o valor atual do mesmo) com probabilidade 0,025 e uma mutação *creep* (na qual um valor aleatório é somado ou subtraído ao valor do gene) com taxa de 0,05 simultaneamente. Entre os operadores de cruzamento estudados, o escolhido foi o cruzamento uniforme parametrizado com um viés de 0,6 e uma taxa de cruzamento igual a 1. No cruzamento uniforme parametrizado, os filhos são gerados através da escolha aleatória, para cada alelo, do valor do alelo de um dos pais, dando maior chance à escolha do pai com melhor aptidão (viés). Adicionalmente, utilizou-se da estratégia de elitismo para preservar o melhor membro da população e novos indivíduos substituem os piores indivíduos da população, não permitindo-se duplicação de soluções. O artigo não detalha o critério de parada utilizado.

Os autores argumentam que o método COMOGA atinge um desempenho similar ao do algoritmo genético com função de penalidade, mas necessita de consideravelmente menos experimentação e é muito menos sensível aos parâmetros envolvidos em seu projeto.

No mesmo ano, Osiadacz e Górecki (1995) apresentaram uma revisão razoavelmente abrangente da literatura sobre a otimização de redes de distribuição de água e gás por métodos heurísticos, procedimentos exatos que consideram a disponibilidade de diâmetros contínuos, e métodos de otimização discreta. Eles propõem resolver o problema com o uso de um método iterativo que minimiza a cada passo uma aproximação quadrática a uma função lagrangeana sujeita a aproximações sequencialmente linearizadas para as restrições. Depois deste passo, os diâmetros são corrigidos para o valor discreto disponível mais próximo. Este

método é aplicado a duas diferentes instâncias, levando a uma redução significativa de custos quando comparado com as melhores soluções conhecidas.

Em outra linha de investigação, Boyd *et al.* (1997) estudaram o problema de minimização de custos de combustível e geraram limites inferiores úteis para avaliar a qualidade das soluções fornecidas por algoritmos de otimização de dutos. Um estudo do espaço de soluções e da função objetivo é realizado, contudo, o problema específico que é tratado está mais preocupado com a minimização do consumo de combustível nas estações de compressão e os limites inferiores desenvolvidos não se aplicam diretamente ao problema do dimensionamento de dutos deste texto.

Redes de distribuição compostas basicamente de bombas e dutos (uma vez que acessórios podem ser modelados apenas por seus diâmetros) são considerados no artigo de Castillo e González (1998). As variáveis de decisão são então definidas como as características Q-H das bombas (taxas de pressão e fluxo) e os diâmetros dos dutos e acessórios tomados a partir de um conjunto discreto de diâmetros comercialmente disponíveis. Uma formulação do problema é apresentada, na qual a velocidade do fluxo compreende a restrição mais importante e um algoritmo genético com operadores especializados de cruzamento é proposto. O algoritmo genético também faz uso de uma função de penalidade para desfavorecer soluções inviáveis, somando ao custo dos dutos o maior custo possível de cada elemento da rede para os quais as restrições sejam violadas. Este procedimento foi aplicado a dois casos de teste teóricos e a um real provindo das atividades de uma companhia de laticínios. Os operadores genéticos especializados demonstraram ser melhores na prática do que os tradicionais.

Finalmente, Cunha e Ribeiro (2004) propõem um algoritmo busca tabu para encontrar o projeto de menor custo de redes de distribuição de água com malhas. Mesmo tratando com um diferente tipo de fluido a ser distribuído, a estrutura matemática do problema é idêntica ao problema da rede de distribuição de gás natural, incluindo as preocupações com restrições similares de pressão mínima e dutos de *upstream*. Além de fornecer uma ampla revisão da literatura sobre abordagens anteriores ao problema, o algoritmo sugerido aparenta ser tanto simples como efetivo em lidar com este problema de otimização. Ele começa de uma solução viável trivial onde a todos os dutos é atribuído o maior diâmetro disponível e, para cada movimento sucessivo, ele diminui o diâmetro de um dos dutos, ao mesmo tempo mantendo válida a solução com respeito a todas as restrições, de forma a

obter a solução de menor custo na vizinhança. Quando o algoritmo encontra uma solução na qual nenhum duto pode ter o diâmetro diminuído sem tornar a solução inviável, ele faz um movimento de diversificação, ou seja, aumenta o diâmetro de um dos dutos, selecionado de acordo com um dos seguintes critérios:

- o duto cujo diâmetro é aumentado é aquele que apresenta o menor valor de um parâmetro dado pela multiplicação do número de mudanças no diâmetro do duto durante o procedimento de busca pelo número de vezes em que o duto foi atribuído ao valor do diâmetro aumentado; ou
- o duto escolhido é aquele cujo diâmetro tem se mantido o mesmo pelo maior período nas iterações anteriores.

Estes critérios definem duas diferentes versões do algoritmo, referidas neste trabalho como BT-1 e BT-2, respectivamente. Em cada caso, a modificação do tamanho do duto cujo diâmetro foi mudado torna-se um movimento tabu e entra na lista tabu. Os autores experimentam valores de permanência na lista tabu fixos (entre 1 e 33 iterações) e variáveis e sugerem que os valores variáveis podem levar a melhores resultados, mas não especificam como esta variação é implementada. Também é utilizado um critério de aspiração para permitir que o *status* tabu de um duto seja ignorado se a redução de seu diâmetro levar a uma solução melhor que qualquer outra solução já encontrada. O artigo sugere como critério de parada um número máximo de 3500 iterações sem melhoria da melhor solução já encontrada.

Comparando os resultados dos algoritmos com as melhores soluções conhecidas para cinco instâncias clássicas do problema, os autores demonstram a efetividade do método proposto. Esse algoritmo, por fornecer os melhores resultados conhecidos para as instâncias estudadas, foi escolhido para a comparação com o algoritmo proposto no presente trabalho. O mesmo foi re-implementado pelo autor para a comparação em condições computacionais semelhantes. Foram mantidas todas as suas características originais, mas o parâmetro de permanência na lista tabu e o critério de parada utilizado foram reavaliados para obter uma melhor adequação ao problema estudado. O estudo do desempenho da busca tabu proposta por Cunha e Ribeiro (2004) com a variação destes parâmetros é mostrado na seção 6.1.3. De forma geral, buscou-se implementar de forma precisa as idéias propostas no artigo original, só realizando modificações no algoritmo quando tais mudanças representam um ganho substancial de performance para o mesmo.

2.3 Formulação Matemática

O problema do dimensionamento de dutos em uma rede urbana de distribuição de gás natural pode ser enunciado como segue: selecionar, a partir de um conjunto discreto de diâmetros de dutos comercialmente disponíveis, a combinação de diâmetros que constitui a rede de menor custo de instalação capaz de suprir um conjunto de nós de demanda com uma pressão de projeto mínima e respeitando as condições de dutos de *upstream*. O modelo matemático proposto pode ser escrito como:

$$\min \sum_{k \in ND} c(D_k) \cdot L_k \quad (2.1)$$

sujeito a:

$$P_i \geq PD_i, \forall i \in N - \{n_0\} \quad (2.2)$$

$$D_k \leq DU_k, \forall k \in ND \quad (2.3)$$

$$D_k \in D, \forall k \in ND \quad (2.4)$$

onde ND é o conjunto de dutos, N é o conjunto de nós, com n_0 representando o nó-fonte do gás natural, D é o conjunto de diâmetros comercialmente disponíveis, D_k é o diâmetro atribuído ao duto k , tomados como as variáveis de decisão, L_k é o comprimento do duto k , $c : D \rightarrow \mathfrak{R}^+$ é uma função do conjunto de diâmetros disponíveis para o conjunto dos números reais que representa o custo do duto por unidade de comprimento em função do diâmetro, P_i é a pressão obtida no nó i para uma dada atribuição de diâmetros $\{D_k\}$, PD_i é a pressão requerida no nó i , e DU_k é o diâmetro do duto de *upstream* do duto k .

A função-objetivo (2.1) representa a minimização do custo total da rede de distribuição de gás natural, expresso como a soma dos custos de cada seção de dutos que a compõe. O custo dos dutos, por sua vez, é uma função do diâmetro dos dutos (tomados como variáveis de decisão) e do tamanho da seção a ser instalada.

O conjunto de restrições (2.2) representa os requisitos de pressão mínima de projeto em cada nó. Estas restrições só podem ser verificadas pela solução sucessiva da equação não linear em (2.5) para obter a queda de pressão entre cada seção de duto, desde a fonte até cada nó de demanda.

$$Q = C \frac{T_b}{P_b} D^{2.5} e^{\left(\frac{P_1^2 - P_2^2}{LGT_a Z_a f} \right)^{0.5}} \quad (2.5)$$

onde Q é a taxa de fluxo, C é uma constante para conversão de sistemas de unidades, T_b é a temperatura base, P_b é a pressão base, D é o diâmetro do duto, e é a eficiência do duto, P_1 e P_2 são as pressões de entrada e saída, respectivamente, L é o comprimento do duto, G é a gravidade específica do gás, T_a é a temperatura do gás, Z_a é o fator de compressibilidade do gás, e f é o fator de fricção do duto.

O conjunto de restrições (2.3) indica a necessidade de dutos de *upstream*, significando que cada duto deve ter um diâmetro igual ou inferior ao diâmetro de seu duto de *upstream*. Uma vez que as redes de distribuição consideradas neste trabalho têm uma estrutura fixa de árvore, o duto de *upstream* pode ser facilmente determinado como o duto que se conecta diretamente ao duto dado e está no único caminho entre o nó-fonte da rede e o duto dado. Se redes de distribuição com malhas fossem consideradas, a única diferença implicada seria que as equações de fluxo teriam que ser resolvidas para determinar as direções dos fluxos e os dutos de *upstream*. Finalmente, as constantes em (2.4) restringem os diâmetros possíveis dos dutos aos comercialmente disponíveis.

2.4 Algoritmo Exato

Nesta seção é apresentado um algoritmo exato para o problema do dimensionamento dos dutos em uma rede urbana de distribuição de gás natural. O algoritmo foi desenvolvido a partir do trabalho de Rothfarb *et al.* (1970) para o problema de dimensionamento de dutos em redes coletoras de gás *offshore*. O mesmo representa também uma extensão da idéia aplicada por Costa (2004) para o problema em estudo no presente trabalho.

O algoritmo exato proposto utiliza-se da técnica de programação dinâmica para repetidamente reduzir a instância em questão a um problema equivalente de menor dimensão. Para isto, são aplicadas duas operações sobre os arcos da árvore que representa o *layout* da rede de distribuição: a fusão em série e a fusão em paralelo. Estas operações são explicadas em detalhes nas seções 2.4.1 e 2.4.2. No

decorrer da aplicação destas operações à instância do problema, a restrição da pressão mínima de projeto é o principal fator considerado. Costa (2004) aplicou idéia semelhante, mas não introduziu no seu algoritmo uma forma de lidar com as restrições de *upstream*. Desta forma, o algoritmo heurístico construtivo proposto naquele trabalho necessita de uma fase adicional de viabilização das soluções quanto aos requisitos de *upstream*. Esta fase adicional consiste em, dado um duto k , identificar o conjunto de dutos U que estão no caminho entre k e o nó-fonte e cujos diâmetros sejam inferiores ao diâmetro do duto k . Caso algum duto exista em U , atribui-se o mesmo diâmetro do duto k aos dutos em U , o que pode levar a obtenção de soluções sub-ótimas.

No algoritmo original de Rothfarb *et al.* (1970), a pressão nos poços é conhecida, o que permite determinar com precisão a pressão em cada trecho e evitar a seleção de dutos que violem as restrições de pressão. Sendo dada como entrada a topologia da rede – em forma de árvore –, a abordagem original percorre a árvore, selecionando os dutos a partir dos pontos terminais até alcançar a refinaria. A Figura 1 ilustra uma instância típica do problema atacado por Rothfarb *et al.* (1970), com o nó 1 representando a refinaria que processa o gás e as folhas da árvore, como os nós 11 e 12, indicando os poços *offshore* produtores de gás.

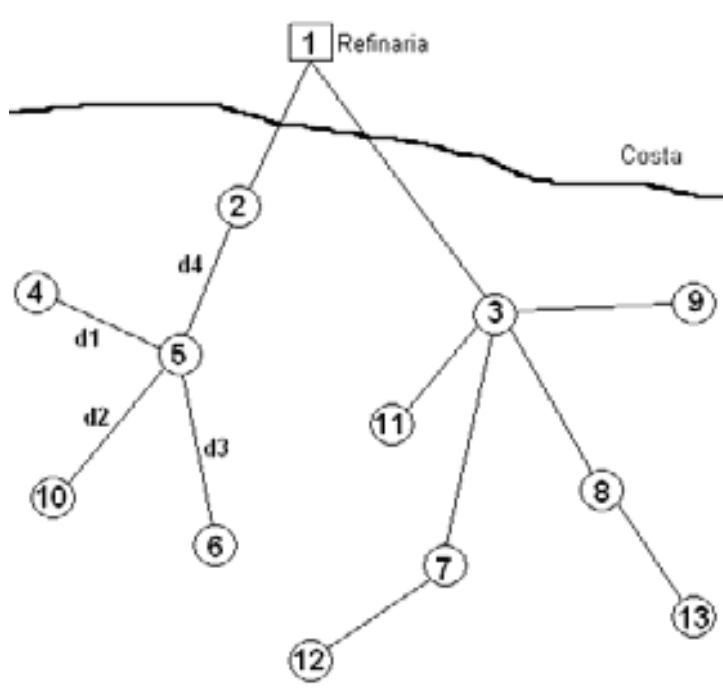


Figura 1: Rede de coleta de gás com poços *offshore*

Em redes de distribuição de gás, por sua vez, são conhecidas as pressões requisitadas pelos clientes e, a partir desses valores, determina-se a pressão mínima em cada trecho da rede que deve ser dimensionado.

Pode-se verificar facilmente que, em uma rede de distribuição de gás com topologia em árvore enraizada, todos os arcos (seções de dutos) estão dispostos em paralelo ou em série, como pode ser visto nas Figuras 2(a) e 2(b). Dois ou mais arcos estão em paralelo quando recebem fluxo a partir de um único nó comum. Por sua vez, dois dutos estão em série quando um dos dutos serve de duto de *upstream* para o outro.

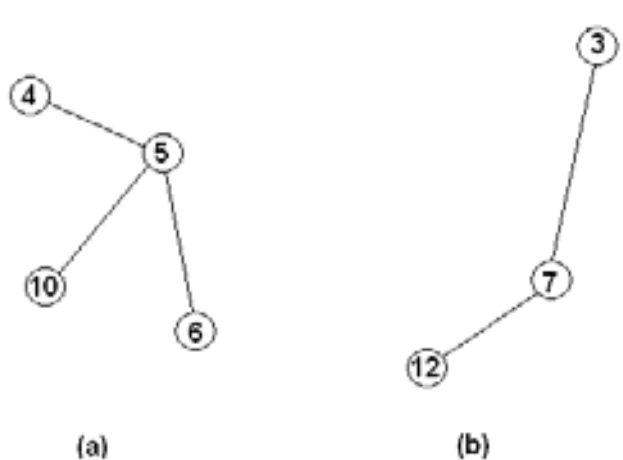


Figura 2: (a) Dutos em paralelo (b) Dutos em série

A escolha dos diâmetros para os dutos da rede de distribuição é determinada, no algoritmo exato proposto, pela diferença entre os quadrados de pressão (PSQ) em cada trecho de tubulação. A partir da equação 2.5, sendo conhecidas a pressão de entrada do gás em um determinado nó e as características do gás e do duto empregado, é possível conhecer o quadrado da pressão de saída na outra extremidade do duto, conforme explicitado nas equações 2.6 e 2.7.

$$P_1^2 - P_2^2 = LGT_a Z_a f \left(\frac{QP_b}{CT_b D^{2.5} e} \right)^2 \quad (2.6)$$

$$P_1^2 - P_2^2 = \frac{Q^2 P_b^2 LGT_a Z_a f}{C^2 T_b^2 D^5 e^2} \quad (2.7)$$

Com isto, pode-se facilmente determinar a queda de pressão em cada trecho de dutos e, dada uma configuração de dimensionamento de dutos, pode-se calcular a pressão final em cada nó de demanda da rede. Uma característica importante a se notar é que as quedas de pressão em cada trecho são dependentes

apenas das características do gás e do duto naquele trecho, isto é, não dependem dos valores específicos das quedas de pressão em nenhum outro trecho. Isto possibilita aplicar a técnica de programação dinâmica para a obtenção da solução ótima para o problema. Outro fato importante a observar é que, trabalhando-se com o quadrado da pressão em cada nó, tem-se uma variação linear desta grandeza entre dois dutos quaisquer em um caminho na árvore. Por exemplo, na Figura 2(b) anterior, se a queda de pressão entre os nós 3 e 7 é $Q_{3,7} = P_3^2 - P_7^2$ e a queda de pressão entre os nós 7 e 12 é $Q_{7,12} = P_7^2 - P_{12}^2$, pode-se verificar facilmente que a diferença de pressão entre os nós 3 e 12 é simplesmente $Q_{3,12} = Q_{3,7} + Q_{7,12} = P_3^2 - P_{12}^2$ e o quadrado da pressão no nó 12 pode ser dado por $P_{12}^2 = P_3^2 - Q_{3,7} - Q_{7,12}$.

Para este algoritmo, a cada arco da árvore será atribuído um conjunto de pares de valores $\{Q_i, C_i\}$, sendo Q_i o quadrado da queda de pressão naquele arco para uma certa atribuição de diâmetro i ao mesmo, e C_i o custo de instalação equivalente. Cada um destes pares de valores será referido como uma atribuição parcial (de diâmetros aos dutos). No início do algoritmo, o conjunto de atribuições parciais correspondente a cada arco da árvore é simplesmente a atribuição de cada possível diâmetro individualmente a cada trecho de duto. No decorrer do algoritmo, o mesmo combina diferentes arcos em trechos equivalentes através das operações de fusão em série e em paralelo, diminuindo o tamanho do problema resultante, como explicado a seguir. Em todo caso, a lista de atribuições parciais é armazenada em ordem crescente de custo – e, conseqüentemente, em ordem decrescente de queda de pressão.

2.4.1 Fusão em Paralelo

A operação de fusão em paralelo pode ser usada com qualquer conjunto de arcos que conectam diretamente nós de grau um a um nó em comum (ver Figura 2(a) anterior). O grau de um nó é o número de arcos que são incidentes àquele nó. Para ilustrar o procedimento, serão usados os arcos d_1 , d_2 e d_3 , cujas atribuições parciais estão exemplificadas na Tabela 1 abaixo. Note-se que, se os três arcos

ainda não passaram por uma operação de fusão em paralelo ou em série pelo algoritmo, os valores de custo e queda de pressão mostrados referem-se ao valor correspondente a cada atribuição individual de diâmetro ao trecho, sendo o menor diâmetro correspondente ao primeiro par de valores e o maior diâmetro correspondendo ao último par.

d_1	Q_i	120	111	92	66	54	40	31
	C_i	13	17	23	29	36	45	67
d_2	Q_i	150	139	118	87	75	70	67
	C_i	6	9	14	21	30	40	56
d_3	Q_i	94	86	80	61	55	48	32
	C_i	8	12	18	26	34	43	57

Tabela 1: Listas de custos e quedas de pressão dos dutos d_1 , d_2 e d_3

O procedimento inicia montando um quadro de cálculos como mostra a Tabela 2. Cada lista de atribuições parciais a ser combinada possui uma coluna no quadro de testes conforme indicado. Se o índice em uma coluna possui um valor i , então as entradas da coluna para Q e C são os Q_i e C_i correspondentes. Inicialmente, todos os índices são iguais a 1, ou seja, as atribuições parciais de menor custo e maior queda de pressão são usadas.

	d_1	d_2	d_3
Q	120	150	94
C	13	6	8
i	1	1	1

Tabela 2: Quadro de cálculos inicial da fusão em paralelo

O procedimento de fusão em paralelo irá combinar as três listas de atribuições parciais em uma lista equivalente, como segue: inicialmente, ele localiza a maior entrada na linha de quedas de pressão (Q) do quadro de cálculos. No exemplo em questão, isto ocorre na coluna d_2 , que é marcada em negrito. Uma atribuição parcial equivalente é então criada, com queda de pressão igual ao da coluna selecionada e custo igual à soma dos custos atuais dos três trechos ($Q_1=150$ e $C_1=13+6+8=27$). Esta entrada na nova lista corresponde à situação em que atribui-

se o menor diâmetro aos três trechos paralelos e esta informação deve ser também armazenada na nova atribuição parcial de diâmetros (para que posteriormente seja possível reconstruir a solução).

Se o menor diâmetro é escolhido para d_2 , então d_1 e d_3 nunca terão uma queda de pressão maior que d_2 . Logo, escolher qualquer outro diâmetro que não o menor possível para d_1 ou d_3 irá aumentar o custo total de instalação de dutos, mas não poderá reduzir a queda de pressão do gás. Claro que isto só é válido no problema estudado se considerarmos como a queda de pressão para um dado arco sua queda de pressão real somada à pressão mínima de demanda no nó atendido por aquele trecho de duto. Assim é como são considerados as quedas de pressão nos dutos para os propósitos deste algoritmo.

Uma vez que nenhuma melhor escolha de diâmetros é possível para d_1 e d_3 com d_2 mantendo o menor diâmetro, promove-se o índice da coluna de d_2 no quadro de cálculos, o que resulta na Tabela 3.

	d_1	d_2	d_3
Q	120	139	94
C	13	9	8
i	1	2	1

Tabela 3: Segundo quadro de cálculos da fusão em paralelo

Neste novo quadro de cálculos, a maior entrada na linha Q ainda ocorre na coluna d_2 . Isto significa que, se d_2 tiver o segundo menor diâmetro possível, ainda não será viável ter d_1 ou d_3 com qualquer outro diâmetro a não ser o menor. A segunda entrada na nova lista de atribuições parciais é construída como a primeira e agora tem-se:

Q_i	150	139
C_i	27	30

Tabela 4: Lista de atribuições parciais equivalente aos dutos d_1 , d_2 e d_3

Esta nova entrada representa uma atribuição parcial do segundo menor diâmetro a d_2 e do menor diâmetro a d_1 e d_3 . Novamente o índice da coluna d_2 é incrementado e o quadro de cálculos da Tabela 5 é obtido.

	d_1	d_2	d_3
Q	120	118	94
C	13	14	8
i	1	3	1

Tabela 5: Terceiro quadro de cálculos da fusão em paralelo

Agora, a maior entrada na linha Q encontra-se na coluna d_1 . Se uma atribuição ótima possuir d_1 com o menor diâmetro, ela não pode conter d_2 com um diâmetro maior que o terceiro menor ou d_3 com um diâmetro maior que o menor possível. Isto leva a uma nova entrada na lista de atribuições parciais equivalentes, como segue:

Q_i	150	139	120
C_i	27	30	35

Tabela 6: Nova lista de atribuições parciais equivalente aos dutos d_1 , d_2 e d_3

Em seguida, promove-se o índice da coluna d_1 , com o resultado mostrado no quadro de cálculos da Tabela 7. O processo termina quando a maior entrada na linha Q do bloco de cálculos ocorrer em uma coluna cujo índice seja igual ao do maior diâmetro disponível. Não será possível aumentar o índice desta coluna e o aumento dos índices das outras colunas corresponderia a atribuições parciais de maior custo e cuja queda de pressão equivalente não diminuiria.

	d_1	d_2	d_3
Q	111	118	94
C	17	14	8
i	2	3	1

Tabela 7: Quarto quadro de cálculos da fusão em paralelo

Os sucessivos quadros de cálculo são mostrados na Tabela 8 abaixo e a lista de atribuições parciais final equivalente à fusão em paralelo dos dutos d_1 , d_2 e d_3 é apresentada na Tabela 9.

	d_1	d_2	d_3
Q	111	87	94
C	17	21	8
i	2	4	1

	d_1	d_2	d_3
Q	92	87	94
C	23	21	8
i	3	4	1

	d_1	d_2	d_3
Q	92	87	86
C	23	21	12
i	3	4	2

	d_1	d_2	d_3
Q	66	87	86
C	29	21	12
i	4	4	2

	d_1	d_2	d_3
Q	66	75	86
C	29	30	12
i	4	5	2

	d_1	d_2	d_3
Q	66	75	80
C	29	30	18
i	4	5	3

	d_1	d_2	d_3
Q	66	75	61
C	29	30	26
i	4	5	4

	d_1	d_2	d_3
Q	66	70	61
C	29	40	26
i	4	6	4

	d_1	d_2	d_3
Q	66	67	61
C	29	56	26
i	4	7	4

Tabela 8: Quadros de cálculo restantes da fusão em paralelo

Q_i	150	139	120	118	111	94	92	87	86	80	75	70	67
C_i	27	30	35	39	46	52	56	62	71	77	85	95	111

Tabela 9: Nova lista de atribuições parciais equivalente aos dutos d_1 , d_2 e d_3

2.4.2 Fusão em Série

A operação de fusão em série pode ser usada com quaisquer dois arcos que sejam incidentes a um nó comum de grau dois se pelo menos um dos arcos é também incidente a um nó de grau um. Os arcos podem ser os trechos originais ou arcos equivalentes gerados em prévias aplicações das operações de fusão em série ou em paralelo.

Um exemplo também será utilizado para explicar o funcionamento da fusão em série. Mais especificamente, o exemplo da fusão em série irá unir a lista de atribuições parciais equivalente obtida no exemplo anterior – que chamaremos d_e e corresponde aos arcos d_1 , d_2 e d_3 – com a lista de atribuições parciais do arco d_4 .

Serão usadas a lista equivalente mostrada na Tabela 9 e a lista de atribuições de d_4 , apresentada na Tabela 10 abaixo.

d_4	Q_i	133	124	104	78	65	51	42
	C_i	6	10	15	23	33	43	59

Tabela 10: Lista de custos e quedas de pressão do duto d_4

Um quadro de cálculos é construído com 7 colunas, como mostrado na Tabela 11. A i -ésima coluna corresponde à escolha do i -ésimo menor diâmetro para o duto d_4 e um índice igual a j em uma coluna corresponde à j -ésima atribuição parcial para d_e . As entradas das linhas de Q e C em uma certa coluna correspondem ao quadrado da queda de pressão e ao custo de instalação associados com tal atribuição parcial de diâmetros para d_1 , d_2 , d_3 e d_4 . Como os dutos estão em série, a linha Q retém o valor da soma das quedas de pressão em d_e e d_4 e a linha C mantém o custo equivalente, dado pela soma dos custos dos dutos em questão.

i	1	2	3	4	5	6	7
Q	283	274	254	228	215	201	192
C	33	37	42	50	60	70	86
j	1	1	1	1	1	1	1

Tabela 11: Quadro de cálculos inicial da fusão em série

O primeiro passo da operação de fusão em série é localizar a maior entrada na linha Q do quadro de cálculos. Inicialmente, todos os índices são iguais a 1 e esta entrada vai sempre ocorrer na primeira coluna. Os valores das linhas Q e C da coluna selecionada tornam-se a primeira entrada na nova lista de atribuições parciais equivalente. O índice da primeira coluna é então incrementado, resultando no quadro de cálculos mostrado na Tabela 12. A nova maior entrada da linha Q é agora a segunda coluna, cuja queda de pressão e custo tornam-se o segundo componente na lista equivalente. A atualização do quadro de cálculos resulta na Tabela 13.

<i>i</i>	1	2	3	4	5	6	7
Q	272	274	254	228	215	201	192
C	26	37	42	50	60	70	86
<i>j</i>	2	1	1	1	1	1	1

Tabela 12: Segundo quadro de cálculos da fusão em série

<i>i</i>	1	2	3	4	5	6	7
Q	272	263	254	228	215	201	192
C	26	40	42	50	60	70	86
<i>j</i>	2	2	1	1	1	1	1

Tabela 13: Terceiro quadro de cálculos da fusão em série

A lista de atribuições parciais equivalente é agora:

Q_i	283	274	272
C_i	33	37	36

Tabela 14: Lista de atribuições parciais equivalente aos dutos d_e e d_4

Cada componente da lista corresponde à uma atribuição parcial de diâmetros aos dutos d_1 , d_2 , d_3 e d_4 . Pelo método de construção empregado, a linha Q sempre terá suas entradas em ordem não-crescente. Note-se, entretanto, que a última entrada na linha C é menor que a segunda entrada na mesma linha. A atribuição parcial correspondente ao terceiro componente da lista é sempre preferível à atribuição parcial correspondente à segunda coluna, uma vez que possui um menor custo e uma menor queda de pressão simultaneamente. Desta forma, elimina-se das considerações posteriores a segunda coluna da lista de atribuições parciais equivalente. A lista se reduz ao mostrado na Tabela 15. Em geral, sempre que uma nova entrada é adicionada à lista equivalente, pode-se eliminar todas as entradas já na lista cujo custo não seja inferior ao da entrada mais recente.

Q_i	283	272
C_i	33	36

Tabela 15: Lista de atribuições parciais equivalente aos dutos d_e e d_4 após redução

Uma outra condição adicional que deve ser considerada na construção da lista de atribuições parciais equivalente para a operação de fusão em série é a restrição de *upstream*. Esta restrição implica que, para cada coluna i do quadro de cálculos, o índice j nunca poderá corresponder a uma atribuição parcial de diâmetros que inclua uma atribuição de diâmetro maior que o diâmetro i . Em outras palavras, sendo d_4 o duto de *upstream* de todos os demais considerados na operação, o diâmetro do duto d_4 sempre deverá ser o maior entre todos os diâmetros atribuídos. Para efeitos de implementação, considera-se que, sempre que o índice j alcançar um valor que corresponde a uma atribuição parcial com algum diâmetro superior a i , esta coluna pode ser ignorada em todas as etapas posteriores, pois a restrição de *upstream* foi violada e nenhuma outra atribuição válida poderá ser obtida a partir da coluna em questão.

O procedimento de fusão em série continua, então, localizando a maior entrada na linha Q e incrementando o índice j correspondente, para gerar novos candidatos a figurarem na lista de atribuições parciais equivalente. O algoritmo continua iterativamente até que todas as colunas viáveis (que respeitam a restrição de *upstream*) contenham o campo índice apontando para a última atribuição parcial do segundo duto, neste exemplo o duto produzido pela fusão em paralelo.

2.4.3 Pseudo-código

Uma vez que as técnicas de fusão em paralelo e em série podem ser aplicadas tanto a listas de dutos reais como a dutos equivalentes, toda a árvore pode ser processada para gerar uma única lista de atribuições parciais equivalentes. A queda de pressão e o custo das atribuições de diâmetros correspondentes a cada entrada desta lista final podem ser avaliados para obter a solução ótima da instância. A atribuição ótima será aquela de menor custo cuja queda de pressão total seja menor ou igual ao valor da pressão de entrada disponível no nó fonte do gás natural. O pseudo-código do algoritmo exato proposto é dado no Quadro 1 abaixo. Note-se que sempre realiza-se primeiro as operações de fusão em série, se possível. Isto é necessário para que se habilitem outras possíveis operações de fusão em paralelo com as listas provenientes da fusão em série.

Para cada aresta da árvore **Faça**

 Crie a lista de atribuições parciais iniciais para o trecho

Fim_Para

Enquanto (número de listas de atribuições parciais > 1) **Faça**

Se (o nó pai de alguma folha f possui apenas 1 filho) **Então**

 Faça fusão em série entre o nó f e seu pai

Senão

 Escolha um nó x que possua apenas folhas como filhos

 Faça fusão em paralelo entre os nós filhos de x

Fim_Se

Fim_Enquanto

Solução_Ótima = Selecione da lista final de atribuições a solução de menor custo cuja queda de pressão total seja menor ou igual ao valor da pressão de entrada

Quadro 1: Pseudo-código do algoritmo exato para o PDD

2.4.4 Corretude e Complexidade

A corretude do algoritmo pode ser demonstrada pelo seguinte argumento. Chama-se dominada uma entrada (Q_i, C_i) numa lista de atribuições parciais se existe alguma outra entrada (Q_j, C_j) na mesma lista, tal que $Q_j < Q_i$ e $C_j < C_i$, ou seja, se ambos o custo e a queda de pressão em j forem inferior aos equivalentes em i . Se considerar-se que, em cada passo do algoritmo, as listas de atribuições parciais sempre contêm todos os possíveis pares (Q_i, C_i) não-dominados para cada trecho de duto (original ou equivalente), então a lista final de atribuições parciais conterá a solução ótima, já que esta não pode ser dominada, e o procedimento listado no último passo do algoritmo do Quadro 1 irá claramente selecionar esta solução ótima.

Resta mostrar que a hipótese é verdadeira, ou seja, as listas de atribuições parciais nunca possuem soluções dominadas. Sem perda de generalidade, pode-se considerar que as listas geradas inicialmente para cada aresta da árvore respeitam esta propriedade. Se não for o caso, as entradas dominadas podem ser removidas já de início pois nunca fará sentido, considerando um trecho individual, utilizar um diâmetro ou material que leve a uma maior queda de pressão e maior custo que

outro também disponível. As duas únicas operações do algoritmo, a fusão em série e a fusão em paralelo, também mantém apenas as atribuições parciais não-dominadas nas listas, conforme explicado nas seções 2.4.1 e 2.4.2. A fusão em paralelo promove o diâmetro de um dos dutos x apenas quando nenhuma melhor escolha de diâmetros é possível para os demais dutos enquanto o diâmetro de x permanecer o mesmo, pelo fato de que x é o que apresenta a maior queda de pressão possível. Por sua vez, a fusão em série sempre remove da lista resultante de atribuições parciais as entradas dominadas pela última entrada inserida, o que, uma vez que as listas são sempre mantidas em ordem crescente de custo e decrescente de queda de pressão, faz com que apenas atribuições não dominadas estejam presentes na lista equivalente. Isto demonstra que o algoritmo encontra corretamente a solução de menor custo que respeita às restrições de pressão mínima. Finalmente, o algoritmo respeita também a restrição dos dutos de *upstream*, pois na fusão em série, único passo em que os dutos são combinados com seus dutos de *upstream*, são descartadas quaisquer soluções cujo diâmetro do duto de *upstream* seja inferior ao do outro duto, conforme explicado na seção 2.4.2. Fica assim, demonstrada a corretude do algoritmo exato proposto.

A complexidade do algoritmo, tanto em tempo como em espaço, é determinada pelo número de atribuições parciais geradas. No pior caso, todo duto só terá disponíveis diâmetros menores que os disponíveis para o duto de *upstream*, então a restrição de *upstream* não removerá nenhuma entrada das atribuições parciais. Por outro lado, a operação de fusão em paralelo gera um número de atribuições parciais da ordem de $\sum_{i=1}^n L_i$, onde n é o número de dutos considerados na operação e L_i é o tamanho da lista de atribuições parciais de cada trecho i . Por sua vez, a operação de fusão em série gera, no pior caso, $L_1 \cdot L_2$ atribuições parciais, sendo $L_1 = C$ se L_1 é o tamanho da lista do duto de *upstream*, onde C é igual ao número de diâmetros disponíveis para este duto individualmente. Considerando C constante para todos os trechos, o número de atribuições parciais geradas no melhor caso – um grafo estrela – é da ordem de $n \cdot C$, enquanto no pior caso – um grafo caminho (*path graph*) –, este número pode ser da ordem de C^n . Portanto, a complexidade do algoritmo desenvolvido, tanto em tempo como em espaço, é $O(n^2 \cdot C)$ no melhor caso e $O(n \cdot C^n)$ no pior caso. Apesar de exponencial no pior caso, para as aplicações do presente trabalho, o mesmo mostrou-se bastante eficiente.

3 O Problema do Passeio do Pistoneio Periódico

De acordo com dados da Unidade de Negócios de Exploração e Produção do Rio Grande do Norte e Ceará (UN-RNCE), da Petrobrás, a Bacia Petrolífera Potiguar possui cerca de 4.000 poços distribuídos nos seguintes campos produtores: Alto da Pedra, Campo de Cajazeiras, Estreito, Salina Cristal, Canto do Amaro, Fazenda Belém e Alto do Rodrigues. Os três últimos utilizam o método de elevação artificial através da unidade móvel do pistoneio (UMP) para a extração de óleo (MEDEIROS NETO, 2004).

A Bacia Potiguar está localizada no extremo nordeste do Brasil, nos estados do Rio Grande do Norte e do Ceará, entre as capitais Natal e Fortaleza. Sua área total é de cerca de 41.000km², abrangendo 21.500km² na parte emersa e 19.500km² na parte submersa. Segundo a Agência Nacional de Petróleo (ANP), o estado do Rio Grande do Norte produziu em 2003 cerca de 79.000 barris de petróleo em campos *on-shore* (terrestres), o que correspondeu a 32% da produção de óleo do país. É relevante destacar que parte relevante dos poços produtores de petróleo neste estado opera com o uso da UMP. O grande volume de óleo coletado sob o regime da UMP, portanto, justifica o estudo de métodos de otimização do passeio da UMP. Melhoramentos na utilização desta técnica podem significar perceptíveis aumentos no volume de óleo produzido nos campos terrestres.

3.1. Descrição do Problema

A exploração de petróleo em campos terrestres é basicamente efetuada em dois tipos de poços: poços surgentes, nos quais a pressão interna dos gases no interior da jazida é suficiente para fazer o óleo aflorar à superfície; e poços não-surgentes, em que é necessário um processo de retirada forçada do petróleo (elevação por meios artificiais), uma vez que a pressão do reservatório é relativamente baixa e os fluidos não alcançam a superfície por si só. Esta situação

também pode ocorrer no final da vida produtiva de poços outrora surgentes ou quando a vazão do poço está em um nível muito abaixo do que poderia produzir.

A retirada do petróleo em poços não-surgentes pode ser efetuada através de um equipamento fixo que é instalado no local do poço, ou através de um equipamento móvel que se desloca entre os poços esgotando seus conteúdos. Segundo Thomas (2001), os métodos de elevação mais comuns na indústria do petróleo são: *gas-lift* contínuo e intermitente (GLC e GLI), bombeio centrífugo submerso (BCS), bombeio mecânico com hastes (BMH) e bombeio por cavidades progressivas (BCP), além da unidade móvel do pistoneio (UMP). Na bacia potiguar, os reservatórios surgentes representam menos de 2% do número de poços. Para os poços não-surgentes, os métodos de bombeio mecânico e bombeio por cavidades progressivas são usados em 81% e 9% dos poços, respectivamente, restando 10% dos poços para os outros métodos, inclusive a UMP.

Quando a vazão do poço não-surgente é muito grande ele é explorado, via de regra, por um equipamento fixo. Quando a vazão do poço não justifica a alocação de um equipamento fixo ele poderá ser explorado por um equipamento móvel denominado unidade móvel de pistoneio (UMP), assim chamado por a operação de extração do petróleo ser realizada pelo processo de pistoneio (GOLDBARG *et al.*, 2001). Uma UMP consiste de um caminhão, equipado com um tanque, guincho hidráulico, cabo de aço, lança móvel e o copo de pistoneio. A UMP normalmente visita mais de um poço a cada dia, esgotando os seus conteúdos e retornando à estação de tratamento ao término de sua jornada de trabalho.

Para uma utilização ótima dos veículos que realizam o pistoneio nos poços não-surgentes, percebe-se a necessidade da programação das rotas de forma a garantir que o maior volume possível de óleo venha a ser coletado. O roteamento da UMP é ainda afetado pelos diferentes volumes de óleo que cada poço pode apresentar no decorrer dos dias, porque após a retirada de fluidos de um poço, inicia-se o processo de reenchimento natural, que irá cessar quando a coluna de fluidos atingir o seu nível estático. A realimentação dos poços obedece à equação (3.1) a seguir:

$$\Delta V = V_{\max} (1 - e^{-t/\tau}) \quad (3.1)$$

Nesta equação, ΔV representa a variação no volume disponível em um poço cujo volume de óleo máximo (coluna estática) é V_{\max} e que foi pistoneado pela última

vez a t unidades de tempo, com a velocidade de reenchimento definida por r . O reenchimento dos poços a diferentes taxas reforça o caráter periódico deste problema. Vale salientar que um poço pode ser explorado antes de atingir o seu nível estático – o que pode levar dias –, desde que o volume do poço seja atrativo em certa rota da UMP. A variação no volume disponível no poço considera que o conteúdo do mesmo é esgotado a cada visita da UMP.

Logo, o problema do passeio do pistoneio periódico consiste em determinar uma rota de visitas aos poços, iniciando e terminando na estação de tratamento de óleo (ETO), para cada dia de um certo horizonte de planejamento, respeitando o limite do turno de trabalho da UMP, de forma que o óleo coletado no conjunto de dias seja máximo. Na versão tratada neste trabalho, considera-se que a UMP sempre dispõe de um caminhão-tanque auxiliar, de forma que o mesmo nunca se encontre completamente cheio e não seja necessário retornar à ETO antes do fim do turno de trabalho ou preocupar-se com a capacidade do tanque da UMP. Em outras palavras, para efeitos deste trabalho, a capacidade de armazenamento da UMP é considerada ilimitada, o que em situações práticas é normalmente válido.

3.2. Revisão da Literatura

Segundo Goldberg e Luna (2000), o Problema do Roteamento de Veículos (PRV) “deseja, com o uso de veículos, visitar uma série de clientes ao menor custo possível, atendendo a todas as demais imposições do problema”. Este problema geral tem inúmeras variantes e generalizações que podem ser encontradas no trabalho referenciado. Backer e Furnon (1997) o definem como NP-árduo, o que mostra a imensa dificuldade envolvida na sua resolução e na solução de problemas correlatos.

Uma vez que mesmo os mais eficazes algoritmos para a solução exata do PRV, como o de Cornuéjols e Harche (1993) e o de Hadjiconstantinou *et al.* (1995), em geral, não conseguem resolver instâncias com mais de 50 clientes, a pesquisa quanto a este problema tem posto um grande foco na sua solução via heurísticas e metaheurísticas, das quais se destacam a busca tabu (TAILLARD, 1993; OSMAN,

1993; GENDREAU *et al.*, 1994) e a heurística em pétala (RENAUD *et al.*, 1996), pelos bons resultados demonstrados.

Uma generalização do PRV é o problema do roteamento de veículos periódico (PRVP). Ainda conforme Goldberg e Luna (2000), “neste caso, um conjunto de clientes requer entregas de certo produto associado a um determinado depósito com uma periodicidade conhecida”. Normalmente, os clientes informam as possíveis combinações de dias de entrega no período de planejamento. Existe um grande número de aplicações práticas deste problema, envolvendo tanto demandas determinísticas quanto estocásticas. Christofides e Beasley (1984) fornecem a formulação clássica deste problema e os trabalhos de Tan e Beasley (1984) e Gaudioso e Paletta (1992) sugerem algoritmos heurísticos para a solução do problema.

Diversos outros trabalhos oferecem soluções para o PRVP e variantes deste através do uso de metaheurísticas. Entre eles, Cordeau *et al.* (1997) propõem uma busca tabu capaz de resolver o dito PRVP e ainda o problema do caixeiro viajante periódico e o problema de roteamento de veículos multi-depósito. Esta busca tabu utiliza principalmente dois movimentos: a troca de um cliente de uma rota para outra do mesmo dia e a troca dos dias de visita de um cliente buscando diminuir o custo total do roteamento.

Por sua vez, Drummond *et al.* (2001) propõem para a solução do PRVP um algoritmo genético paralelo hibridizado com heurísticas de busca local. O algoritmo genético usa um modelo de ilha, mantendo uma baixa frequência de migração, o que levou a um desempenho superior que o de outras heurísticas relatadas.

Baptista *et al.* (2002) abordam o PRVP como um problema multi-nível, composto de um problema de alocação e um problema de roteamento de veículos clássico, observando os aspectos temporal e espacial do problema. Neste trabalho, uma heurística – que consiste em uma atribuição inicial de clientes a dias e veículos a rotas, seguida de uma seqüência de trocas destes parâmetros – é usada para resolver um caso real de roteamento para coleta de contêineres de reciclagem de papel.

Ainda Angelelli e Speranza (2002) aplicam um algoritmo busca tabu na solução do PRVP com instalações intermediárias. Nesta variante do PRVP, os veículos podem renovar sua capacidade em instalações intermediárias e só

precisam retornar ao depósito ao término de suas atividades. A mesma busca tabu é aplicada ao PRVP e apresenta bons resultados com instâncias da literatura.

Um trabalho recente de Alegre *et al.* (2005) sobre o PRVP ataca instâncias com horizonte de planejamento mais longos, oriundas das operações de um fabricante de partes de automóveis da Espanha. Neste caso especial, as decisões críticas são as atribuições de locais a dias específicos, dando-se menos importância às rotas diárias (por constituírem um problema já bem estudado). Os autores argumentam que sua principal contribuição é o desenvolvimento de um algoritmo *scatter search* mais efetivo que as heurísticas da literatura para problemas com horizontes de planejamento mais longos.

Uma outra variante do PRV, ainda pouco estudada, diz respeito ao problema do roteamento de veículos seletivo (PRVS), na qual nem todos os clientes precisam ser atendidos, mas só aqueles que apresentem uma vantagem econômica para tal. Os primeiros trabalhos que tratam do PRVS são o de Butt e Ryan (1999) e o de Gueguen (1999). Para Butt e Ryan a frota de veículos é heterogênea e o problema é definido em um grafo completo. Gueguen introduz janelas de tempo associadas com as datas de visita dos clientes. Para a solução do dito problema, ambos os autores propõem um procedimento aproximativo baseado no método de geração de colunas, adaptando a formulação tradicional do problema de particionamento empregada para o PRV.

Tang (2001) fornece uma formulação e procedimentos de solução para o problema em um trabalho que aborda uma série de problemas de roteamento. Ele desenvolve algoritmos gulosos e uma busca tabu para a solução do PRVS e reporta bons resultados para um conjunto de instâncias encontradas na literatura.

Já Hayari *et al.* (2003) são os primeiros a proporem um algoritmo evolucionário para a versão com janelas de tempo do PRVS. O algoritmo genético proposto utiliza operadores de cruzamento e mutação especializados para problemas de roteamento e esquemas especiais de avaliação de aptidão e seleção. Resolvendo instâncias com até 100 nós e 8 veículos, os autores alegam ter resultados comparáveis aos reportados por Gueguen (1999).

Recentemente, Semet *et al.* (2005) apresentaram uma metaheurística híbrida para a solução do PRVS, na qual uma busca tabu foi combinada com uma busca em vizinhança variável descendente (VND). A VND gera decomposições dos nós em agrupamentos compactos em termos de distância e que possam gerar um

grande lucro. Nestes agrupamentos de vértices são então construídas rotas por meio da busca tabu. A abordagem híbrida produziu bons resultados, levando a descoberta de um número de novas melhores soluções conhecidas.

Existem ainda alguns trabalhos descrevendo algoritmos exatos e aproximados para o problema do passeio do pistoneio na sua versão não periódica. A maioria destes trabalhos são monografias e dissertações de mestrado, com motivações práticas, desenvolvidas no Programa de Pós-Graduação em Sistemas e Computação da Universidade Federal do Rio Grande do Norte.

As primeiras abordagens são feitas por Ferreira (2000), que ataca o problema com o uso de infecções virais, e Neves (2000), que o soluciona através de um algoritmo genético. Dantas (2001), por sua vez, utiliza uma busca tabu para resolver o problema e esboça um algoritmo exato capaz de resolver certas classes de instâncias num horizonte de um dia.

Fernandes (2001) desenvolve e implementa um algoritmo transgenético para o problema não-periódico. Goldberg *et al.* (2001) apresentam este algoritmo, enfatizando a metáfora transgenética e ressaltando bons resultados ao compará-lo com um algoritmo genético para o mesmo problema.

A metaheurística GRASP é utilizada para a solução do PPP nos trabalhos de Barros (2001) e Santos (2002), que desenvolve um sistema de apoio à decisão georeferenciado.

Ferreira (2000), Dantas (2001) e Fernandes (2001) também iniciam o tratamento da versão periódica do problema ao adaptar suas implementações para determinar rotas para 15 dias consecutivos de operação da UMP, mas considerando que não poderia haver repetição de visita aos poços durante este horizonte de tempo. Esta condição, entretanto, não condiz com a realidade, uma vez que um poço pode vir a ser visitado novamente desde que a recuperação de seu volume seja considerada atrativa para a rota, e, por este motivo, estes algoritmos não são comparados com o proposto no presente trabalho.

Por fim, Medeiros Neto (2004) resolveu o problema através de um conjunto de abordagens evolucionárias, incluindo um algoritmo genético, um algoritmo memético e um algoritmo transgenético ProtoG. Com os experimentos computacionais realizados, o autor concluiu que o desempenho do algoritmo ProtoG foi superior tanto em relação à qualidade das soluções quanto ao tempo computacional relacionado.

A primeira tentativa de abordar o real problema do passeio do pistoneio periódico aparece no trabalho de Lima Neto (2005). O mesmo desenvolve um algoritmo transgenético ProtoG e uma heurística baseada na proximidade entre as soluções segundo uma métrica característica, mas seus resultados mostram-se não condizentes com a estrutura das instâncias testadas.

3.3 *Formulação Matemática*

O problema do passeio do pistoneio (PPP) pode ser classificado como um problema de roteamento de veículos seletivo, o que permite que seja modelado como um problema em grafos (GOLDBARG; LUNA, 2000). A modelagem da versão periódica do problema dada a seguir é baseada no trabalho de Medeiros Neto (2004) para a versão não-periódica do mesmo, com as devidas ressalvas, contendo idéias de eliminação de sub-ciclos oriundas da formulação de Dantzig-Fulkerson-Johnson (DFJ) para o problema do caixeiro viajante.

Seja um grafo simples $G = (N, A)$, onde $N = \{0, 1, \dots, n\}$ é o conjunto de nós do grafo, com o nó zero representando o depósito ou estação de tratamento de óleo e os demais representando poços pistoneáveis pela Unidade Móvel do Pistoneio (UMP), e $A = \{(i, j) \mid i, j \in N \wedge i \neq j\}$ é o conjunto de arestas, representando os pares de nós que são acessíveis um a partir do outro no grafo. O peso associado a cada aresta, d_{ij} , é o tempo necessário para a UMP ir do nó i até o nó j . Note-se que o grafo em questão pode ser assimétrico, ou seja, possuir $d_{ij} \neq d_{ji}$. A cada nó associa-se um valor v_i^l , representando o volume de óleo disponível em cada poço i , dado que o poço foi pistoneado pela última vez há l dias (v_i^0 representa o volume de óleo disponível no poço antes de ser pistoneado qualquer vez, ou seja, é o volume de óleo disponível no nível estático de coluna). Cada nó possui ainda um rótulo t_i^l , o tempo de trabalho para pistonear o poço i , incluindo a montagem do equipamento, o pistoneio em si e a desmontagem da máquina, também dado que o poço foi pistoneado pela última vez há l dias (t_i^l representa o tempo de operação no poço i quando este contém o volume de óleo v_i^l). Seja ainda L o turno de trabalho da UMP,

ou seja, o tempo máximo que a mesma pode operar, incluindo os tempos gastos em deslocamento e no trabalho nos poços. Considerando um planejamento para vários dias, seja P o horizonte de planejamento em dias que o problema deve tratar.

O problema do passeio do pistoneio periódico consiste em determinar um conjunto de rotas, uma para cada dia do horizonte de planejamento, tal que o volume de óleo total coletado seja máximo, desde que cada rota respeite o limite do turno de trabalho da UMP. Uma restrição adicional é que ao determinar o volume coletado em cada poço bem como o tempo de pistoneio nos mesmos, deve-se considerar os valores condizentes com o dia no qual aquele poço foi visitado (pistoneado) pela última vez.

Sejam as seguintes variáveis de decisão:

$$x_i^{kl} = \begin{cases} 1 & \text{caso o nó } i \text{ seja visitado na rota do dia } k, \text{ tendo a última visita a este poço} \\ & \text{se dado no dia } l, 1 \leq i \leq n, 1 \leq k \leq P, 1 \leq l \leq k. \text{ Quando } l = k, \text{ considera-se} \\ & \text{que é a primeira visita àquele poço no horizonte de planejamento} \\ & \text{estudado.} \\ 0 & \text{caso contrário.} \end{cases}$$

$$y_{ij}^k = \begin{cases} 1, & \text{caso a aresta } (i, j) \text{ seja percorrida na rota do dia } k, 0 \leq i, j \leq n, i \neq j, 1 \leq k \leq P. \\ 0, & \text{caso contrário.} \end{cases}$$

O problema do passeio do pistoneio periódico pode ser formulado como segue, sendo M é um valor suficientemente grande:

$$\text{Max} \sum_{k=1}^P \sum_{l=1}^k \sum_{i=1}^n x_i^{kl} v_i^{k-l} \quad (3.2)$$

sujeito a:

$$\sum_{i \in N} y_{ij}^k \leq 1, \forall j \in \{0, 1, \dots, n\}, \forall k \in \{1, 2, \dots, P\} \quad (3.3)$$

$$\sum_{j \in N} y_{ij}^k \leq 1, \forall i \in \{0, 1, \dots, n\}, \forall k \in \{1, 2, \dots, P\} \quad (3.4)$$

$$\sum_{j \in N - \{0\}} y_{0j}^k = 1 = \sum_{i \in N - \{0\}} y_{i0}^k, \forall k \in \{1, 2, \dots, P\} \quad (3.5)$$

$$\sum_{j \in N} y_{ij}^k - \sum_{j \in N} y_{ji}^k = 0, \forall i \in \{0, 1, \dots, n\}, \forall k \in \{1, 2, \dots, P\} \quad (3.6)$$

$$\sum_{i \in S} \sum_{j \in S} y_{ij}^k \leq |S| - 1, \forall S \subseteq N - \{0\}, S \neq \emptyset, \forall k \in \{1, 2, \dots, P\} \quad (3.7)$$

$$\sum_{i \in N} \sum_{j \in N} d_{ij} y_{ij}^k + \sum_{i \in N - \{0\}} \sum_{l=1}^k t_i^{k-l} x_i^{kl} \leq L, \forall k \in \{1, 2, \dots, P\} \quad (3.8)$$

$$\sum_{l=1}^k x_i^{kl} \leq 1, \forall i \in \{1, 2, \dots, n\}, \forall k \in \{1, 2, \dots, P\} \quad (3.9)$$

$$x_i^{kl} - \sum_{s=1}^l x_i^{ls} \leq 0, \forall i \in \{1, \dots, n\}, \forall k \in \{1, 2, \dots, P\}, \forall l \in \{1, 2, \dots, k\} \quad (3.10)$$

$$Mx_i^{kl} + \sum_{s=l+1}^{k-1} \sum_{t=1}^s x_i^{st} \leq M, \forall i \in \{1, \dots, n\}, \forall k \in \{1, 2, \dots, P\}, \forall l \in \{1, 2, \dots, k-1\} \quad (3.11)$$

$$Mx_i^{kk} + \sum_{s=1}^{k-1} \sum_{t=1}^s x_i^{st} \leq M, \forall i \in \{1, \dots, n\}, \forall k \in \{1, 2, \dots, P\} \quad (3.12)$$

$$\sum_{l=1}^k x_i^{kl} - \sum_{j \in N} y_{ij}^k = 0, \forall i \in \{1, 2, \dots, n\}, \forall k \in \{1, 2, \dots, P\} \quad (3.13)$$

$$x_i^{kl} \in \{0, 1\}, \forall i \in \{1, \dots, n\}, \forall k \in \{1, 2, \dots, P\}, \forall l \in \{1, 2, \dots, k\} \quad (3.14)$$

$$y_{ij}^k \in \{0, 1\}, \forall i \in \{0, 1, \dots, n\}, \forall j \in \{0, 1, \dots, n\}, \forall k \in \{1, 2, \dots, P\} \quad (3.15)$$

A função-objetivo (3.2) procura maximizar o volume total de óleo coletado em todos os dias do planejamento, considerando qual o volume de óleo que o poço apresentará quando for pistoneado, dado a última visita da UMP ao mesmo. Os conjuntos de restrições (3.3) e (3.4) garantem que a cada dia, no máximo uma aresta chega e uma sai de cada nó do grafo. O conjunto de restrições (3.5) garante que o depósito será incluído nas rotas de todos os dias, havendo sempre uma aresta chegando e uma saindo do nó 0 a cada dia. O conjunto de restrições (3.6) garante a continuidade da rota, ao exigir que, a cada dia, se um nó tem uma aresta de entrada, obrigatoriamente possui também uma aresta de saída. O conjunto de restrições (3.7) evita o surgimento de sub-ciclos, ao exigir que qualquer subconjunto S de nós do grafo que não inclua o depósito (nó 0) não possa ter mais que $|S|-1$ arestas, o que não permite que haja um ciclo entre os nós do subconjunto. Quando o depósito (nó 0) faz parte do subconjunto, entretanto, é possível fechar o ciclo. Como este nó obrigatoriamente possui apenas uma aresta de entrada e uma de saída (conjunto de restrições (3.6)), não mais que um ciclo pode ser formado a cada dia. O conjunto de restrições (3.8) garante que o tempo total de percurso entre os poços e de trabalho nos poços pistoneados seja no máximo igual ao turno de trabalho, levando em conta que o tempo de pistoneio nos poços depende de quando se deu a última visita da

UMP ao poço. O conjunto de restrições (3.9) determina que as visitas a cada poço i para cada dia k possuam uma visita imediatamente anterior única e bem definida. O conjunto de restrições (3.10) implica que se um poço i é visitado no dia k e tem como visita anterior o dia l , então este poço também é visitado no dia l . O conjunto de restrições (3.11) garante que se um poço é visitado no dia k , tendo seu último pistoneio se dado no dia l , então nenhuma visita a este poço pode existir entre os dias l e k , exclusive. As restrições em (3.12) impõem que se o primeiro pistoneio a um poço se dá no dia k , então nenhum pistoneio pode acontecer nos dias anteriores a k . Finalmente, o conjunto de restrições (3.13) faz o acoplamento das variáveis de decisão e as restrições em (3.14) e (3.15) exigem que estas variáveis assumam apenas valores inteiros binários.

3.4 Limite Superior

Dada a escassez de abordagens alternativas para comparação e a grande dificuldade de solução exata, um limite superior para o problema do passeio do pistoneio periódico foi desenvolvido através do uso do ferramental da programação dinâmica. Segundo Brassard e Bratley (1996), a programação dinâmica é uma técnica *bottom-up*, na qual usualmente inicia-se resolvendo as menores (mais simples) sub-instâncias e, combinando suas soluções, obtém-se as respostas para sub-instâncias de tamanhos crescentes, até finalmente alcançar a solução da instância original. O método baseia-se no princípio da otimalidade. Este princípio afirma que em uma seqüência ótima de decisões ou escolhas, cada sub-seqüência deve também ser ótima. Isto significa que as soluções para as sub-instâncias devem ser ótimas para se alcançar um resultado ótimo para a instância original. Este método é normalmente implementado através da manutenção das soluções para os sub-casos em formato tabular. Brassard e Bratley (1996) trazem diversos exemplos de resolução de problemas através desta técnica, enquanto Cormen *et al.* (2002) dão um tratamento matemático mais profundo e ilustram com outros diferentes exemplos de problemas resolvidos pelo método.

O procedimento criado para obter um limite superior para o PPPP assemelha-se ao algoritmo de resolução do problema da mochila através da

programação dinâmica. No dito problema da mochila, o objetivo é preencher uma mochila com diferentes objetos, cada um deles associado a certos peso e valor, de forma que o valor carregado seja o maior possível e a dada capacidade da mochila não seja excedida. Este é um problema de grande aplicação prática e inerente dificuldade de solução, possuindo ainda um vasto número de casos especiais e problemas relacionados. O problema pode ser formulado como segue (GOLDBARG; LUNA, 2000):

$$\text{Max} \sum_{j=1}^n c_j x_j \quad (3.16)$$

sujeito a:

$$\sum_{j=1}^n w_j x_j \leq B \quad (3.17)$$

$$x_j \in Z^+ \quad (3.18)$$

onde x_j representa o número de objetos do tipo j selecionados para serem incluídos em uma mochila com capacidade total de B unidades. O coeficiente c_j representa o valor de cada item e w_j o peso do item. Pode-se considerar ainda o caso especial em que apenas um objeto de cada tipo pode ser escolhido, denominado problema da mochila 0-1, no qual a restrição da variável inteira é substituída por $x_j \in \{0,1\}$.

Conforme Garey e Johnson (1979), o problema da mochila é NP-árduo, mas o mesmo pode ser resolvido através de um algoritmo de programação dinâmica com tempo de execução pseudo-polinomial, ou seja, dependente não apenas do número de itens disponíveis (grandeza representativa do tamanho da instância), mas também do valor da capacidade da mochila (BRASSARD; BRATLEY, 1996). A versão do algoritmo aplicada para a geração do limite superior do PPPP é baseada na solução para o problema da mochila 0-1, portanto este é o algoritmo apresentado no Quadro 2 mais adiante.

No caso da aplicação do problema da mochila 0-1 à geração de um limite superior para o PPPP, a analogia é a seguinte: os objetos a serem guardados na mochila são visitas aos poços pistoneáveis, o valor dos objetos relaciona-se com o volume de óleo a ser coletado na visita e o peso dos objetos – assim como a capacidade da mochila – refere-se ao tempo que precisa ser empregado na referida visita – ou o tempo total disponível para a rota da UMP. Esta analogia permite a

aplicação quase direta do algoritmo de solução do problema da mochila 0-1 ao problema de gerar um limite superior para o PPPP. Em seguida, descreve-se em detalhes cada um destes elementos do algoritmo, o que demonstra que o valor obtido pelo método é de fato um limite superior para o PPPP, uma vez que utiliza-se apenas condições conservadoras em relação à instância original do problema. Também esboça-se um estudo da complexidade do algoritmo em questão, o que mostra-se relativamente simples, devido à estrutura do algoritmo.

```

seja uma matriz M [n+1][B]; // número de objetos por capacidade da mochila
seja um vetor c [n]; // valor dos objetos
seja um vetor w [n]; // peso dos objetos
para j = 0 até B faça
    M [0][j] = 0;
fim_para
para i = 1 até n faça
    para j = 0 até w [i] – 1 faça
        M [i][j] = M [i-1][j];
    fim_para
    para j = w [i] até B faça
        M [i][j] = max (M [i-1][j], M [i][j-1]);
        se (M [i-1][j – w[i]] + c [i] > M [i][j]) então
            M [i][j] = M [i-1][j – w [i]] + c [i];
        fim_se
    fim_para
fim_para
// M [n][B] contém o valor máximo possível de ser carregado na mochila

```

Quadro 2: Algoritmo para o problema da mochila 0-1

Como não é possível determinar a priori quais os volumes de óleo a serem encontrados nos poços em cada rota para a solução ótima (até porque isto resolveria o problema de forma exata), deve-se trabalhar com todos os possíveis volumes de óleo em cada poço. Em outras palavras, consideram-se como “objetos” possíveis de serem guardados na “mochila” cada um dos poços com o volume total

e com o volume após um dia depois de pistoneado, após dois dias depois de pistoneado, etc.

Considera-se o valor de um poço como sendo o seu volume de óleo e o peso do mesmo como sendo o tempo gasto no seu pistoneio somado ao menor tempo possível para chegar ou deixar o poço (menor aresta que chega ou deixa o nó). Todo poço incluído em alguma rota deve ser alcançado por alguma aresta; o tempo gasto para percorrê-la é pelo menos igual ao peso da menor aresta. A capacidade da mochila, por sua vez, será o turno de trabalho da UMP menos o valor da menor aresta que chega ou deixa o depósito. Igualmente, esta suposição é conservadora o suficiente por garantir que não obteremos uma solução pior que a solução ótima. Adicionalmente, para cada dia do horizonte de planejamento, só são considerados os volumes de óleo disponíveis nos poços possíveis até aquele dia, ou seja, os volumes que os poços podem apresentar se tiverem sido pistoneados em algum dos dias anteriores apenas ou se nunca foram pistoneados. Em outras palavras, em um dado dia j , os poços não podem ter um volume de óleo igual ao que apresentariam se houvessem sido pistoneado há j ou mais dias. O pseudocódigo deste algoritmo é dado no Quadro 3 mais adiante.

Vale ressaltar que o problema estudado possui uma formulação matemática de grande complexidade, incluindo restrições para evitar a formação de sub-ciclos que crescem exponencialmente em número conforme o número de nós da instância. Por este motivo, extrair limites superiores por relaxações desta formulação pode ser consideravelmente custoso. Logo, o limite superior desenvolvido a partir da técnica da programação dinâmica é um considerável avanço para a análise da qualidade das soluções para as instâncias do problema tratado.

O algoritmo apresentado é capaz de dar um limite superior para o problema do passeio do pistoneio periódico em um tempo bastante reduzido. Sua complexidade em tempo é claramente determinada pelos três laços aninhados, que resultam em uma ordem de grandeza igual a $O(P \cdot n \cdot L)$. Esta é uma complexidade bastante razoável para os tamanhos das instâncias estudadas neste problema. Como pode ser visto no capítulo 6 quando descreve-se a estrutura das instâncias, o número de operações a serem realizadas pelo algoritmo foi normalmente limitada a uma ordem de grandeza não maior que 10^7 . O espaço em memória necessário para a execução do algoritmo também é bem reduzido, sendo determinado

principalmente pela matriz M, que ocupou não mais que poucos *megabytes* (2^{20} bytes).

```

seja uma matriz M [n+1][L]; // número de nós por duração do turno de trabalho
seja um vetor d [n]; // menor aresta que chega ou deixa o nó i
limite = 0.0;
L = L - d [0];
para k = 0 até L faça
    M [0][k] = 0;
fim_para
para j = 0 até P-1 faça
    volume = 0.0;
    para i = 1 até n faça
        t =  $t_i^j + d[i]$ ;
        para k = 0 até L faça
            M [i][k] = max (M [i][k], M [i][k-1]);
            se (k ≥ t e M [i-1][k-t] +  $v_i^j > M [i][k]$ ) então
                M [i][k] = M [i-1][k-t] +  $v_i^j$ ;
            volume = max (volume, M [i][k]);
        fim_para
    fim_para
    limite = limite + volume;
fim_para
// “limite” contém o valor do limite superior teórico para a instância

```

Quadro 3: Algoritmo para a geração do limite superior para o PPPP

4 Busca Tabu

Entre os numerosos métodos heurísticos propostos nas últimas décadas para a solução de problemas complexos de otimização, a metaheurística busca tabu se destaca como uma heurística de sucesso em muitos campos de aplicação. Suas principais características e partes componentes se baseiam em analogias com o processo de memória humano e são detalhadamente explicadas por Glover e Laguna (1997).

O método da busca tabu propõe, a partir de uma solução gerada inicialmente por um método mais simples – aleatoriamente em muitos casos –, iterativamente realizar uma busca na vizinhança da solução corrente em uma ordem tal que permita que os melhores movimentos sejam explorados e ao mesmo tempo previna a repetição de soluções previamente visitadas. Para alcançar este objetivo, em primeiro lugar certa estrutura de vizinhança e uma função para geração de soluções vizinhas por meio de movimentos estruturados têm que ser definidas. A critério do projetista do algoritmo, a exploração da vizinhança pode ser feita completamente ou restrita a certo número de soluções.

Uma parte adicional da funcionalidade da busca tabu é prevenir a visita repetida às mesmas soluções. Isto é alcançado pela manutenção de uma lista de movimentos tabu, construída ao atribuir-se um *status* de tabu (proibido) a alguns dos possíveis movimentos, significando que estes movimentos não podem ser realizados pela função de geração de soluções vizinhas por um determinado número de iterações. Este mecanismo pode evitar a ciclagem entre soluções já visitadas. O parâmetro de permanência na lista tabu define o número de iterações que os movimentos adicionados à lista tabu mantêm esta condição, tornando a busca mais ou menos restritiva. Uma implementação de uma busca tabu pode ou não fazer uso de um critério de aspiração. Este critério adicional permite que um movimento tabu seja executado em uma dada iteração se ele levar a uma solução que satisfaz alguma condição, como, por exemplo, se a nova solução é melhor que qualquer outra solução encontrada previamente. Uma regra usualmente implementada em algoritmos de busca tabu é a de que as sucessivas soluções analisadas com o uso da função de geração de soluções vizinhas devem ser progressivamente melhores.

Contudo, eventualmente, a busca vai levar a uma solução ótima local e nenhum movimento na estrutura de vizinhança escolhida será capaz de trazer melhorias ao processo. Nestes casos, o algoritmo pode utilizar um procedimento de diversificação que irá permitir a exploração de regiões do espaço de busca que ainda não foram consideradas. A habilidade de determinar estas regiões não exploradas do espaço de soluções é suportada pelo uso de estruturas de memória de longo prazo que mantêm um registro das vizinhanças já visitadas. Desta forma, a busca pode “saltar” para outras regiões do espaço de soluções, reiniciando a partir de uma solução possivelmente pior – mas essencialmente diferente – que a solução atual. Finalmente, a busca é concluída quando um ou mais critérios de parada previamente definidos são alcançados. Com esta descrição, fica evidente que esta metaheurística incorpora estratégias para evitar eficientemente os ótimos locais mesmo mantendo uma implementação relativamente simples e este fato pode explicar parcialmente a popularidade do método.

4.1 Implementação

Neste trabalho, um algoritmo busca tabu é proposto para a solução do problema de dimensionamento de dutos em uma rede urbana de distribuição de gás natural. O algoritmo é baseado em características simples mas que mostram-se efetivas através dos testes realizados. O algoritmo proposto é implementado de forma a trabalhar apenas com soluções viáveis, o que significa que os movimentos considerados em sua estrutura de vizinhança nunca geram soluções que violam as restrições do problema. Isto simplifica a função objetivo utilizada, no sentido de que não é necessária a aplicação de funções de penalidade para tratar com quebras das restrições e o objetivo é simplesmente minimizar o custo total de instalação da rede de distribuição de gás projetada.

Esta seção descreve, nos próximos sub-tópicos, as características básicas envolvidas na implementação do algoritmo proposto. Esta descrição nada mais é que um mapeamento entre as variáveis do modelo geral proposto pela metaheurística e sua aplicação frente aos dados e particularidades do problema estudado. Estas características são: a solução inicial, a exploração da vizinhança

(incluindo os movimentos de geração de vizinhos, o procedimento de diversificação, o critério de aspiração, os movimentos tabus e o parâmetro de permanência na lista tabu), e o critério de parada. Adicionalmente, um pseudocódigo do algoritmo descrito é fornecido no final do capítulo.

4.1.1 Solução Inicial

Uma vez que se está interessado em manter a viabilidade das soluções por todas as iterações, é necessário considerar um método de geração da solução inicial que garanta que a mesma irá respeitar ambas as restrições de pressão mínima e *upstream*. Portanto, a opção escolhida foi empregar como solução inicial a rede de distribuição mais segura possível (apesar de ser também a mais cara): aquela em que a cada duto é atribuído o maior diâmetro disponível, o que garante a menor queda de pressão possível ao longo de todas as seções. Esta forma de gerar a solução inicial ainda traz o benefício adicional de determinar se a instância em questão tem alguma solução viável já no início da execução do algoritmo. Se a solução inicial proposta não for viável com relação à restrição de pressão, nenhuma outra solução o será. A restrição de pressão mínima de projeto é, em realidade, a restrição de mais importante verificação, pois sempre é possível obter soluções que respeitam a restrição de *upstream* (e de fato, a solução inicial escolhida respeita esta restrição). Uma solução inicial aleatória não seria adequada pelo fato de que a densidade do espaço de busca com relação a soluções viáveis pode ser extremamente baixa e poderia ser necessário um enorme tempo computacional para encontrar aleatoriamente uma delas.

4.1.2 Exploração da Vizinhança

Para detalhar como é caracterizada e como o algoritmo proposto explora a vizinhança da solução atual, o movimento de geração de vizinhos é explicado. Primeiro, note-se que as soluções são representadas como uma lista de dutos,

ordenada pela profundidade dos nós incidentes na árvore enraizada da instância do problema. O movimento básico consiste em diminuir o diâmetro atribuído a um duto até o menor valor possível que ainda mantém a solução viável. Isto é feito a partir das folhas até a raiz da árvore (ou seja, do final da lista de dutos para o início) para favorecer o respeito das restrições de *upstream*. A cada passo, dentre os n possíveis vizinhos da solução atual – sendo n o número de dutos da instância –, o duto escolhido para ter o diâmetro diminuído é aquele cuja diminuição resulte na maior redução de custo possível. Devido à complexidade das restrições envolvidas, a pressão em cada nó deve ser recalculada a cada redução individual de diâmetro para garantir que a rede ainda respeita todos os requisitos. A Figura 3 exemplifica uma seqüência de movimentos possível de ser realizada pela busca tabu.

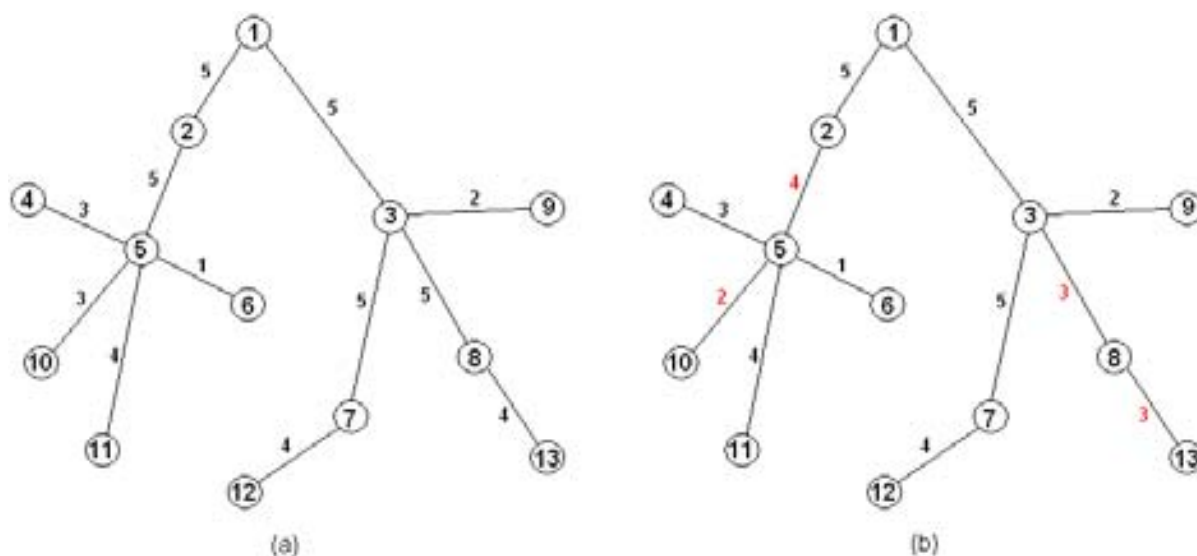


Figura 3: (a) Rede antes do movimento (b) Rede após o movimento

Neste exemplo, a busca tabu gerou a rede em (b) a partir da rede em (a) através de quatro reduções de diâmetros. Pela forma como os dutos são armazenados, as reduções foram feitas na seguinte ordem: dutos (8,13), (5,10), (3,8) e (2,5). Pode-se notar que a diminuição do diâmetro do duto (3,8) para o valor 3 só foi possível porque a redução do diâmetro do duto (8,13) ocorreu anteriormente, devido à ordem em que a análise dos dutos é executada. Note-se ainda que este mesmo duto teve seu diâmetro reduzido em mais de uma unidade (ou mais de um índice na lista de diâmetros disponíveis). Isto, em última instância, acelera o processo de busca.

4.1.3 Procedimento de Diversificação

Quando uma solução é alcançada na qual nenhuma redução de diâmetro adicional é possível, um ótimo local foi encontrado. Para dar continuidade à busca tabu, um procedimento de diversificação é aplicado neste instante. A próxima solução a ser examinada é então gerada pelo incremento do diâmetro de um dos dutos. Um duto aleatório é selecionado (dentre aqueles cujo diâmetro não seja já o maior disponível) e seu diâmetro é aumentado para o próximo valor entre os diâmetros disponíveis. Este movimento sempre mantém a rede válida com respeito às restrições de pressão (pois a queda de pressão no trecho alterado sempre será reduzida), mas pode fazer com que o duto selecionado passe a violar os requisitos de *upstream*. Isto é facilmente corrigido em seguida ao se aumentar recursivamente o diâmetro do duto de *upstream* do duto alterado, repetindo enquanto necessário (possivelmente até alcançar a raiz da árvore).

4.1.4 Movimentos Tabus e Critério de Aspiração

Quer o diâmetro do duto seja diminuído quer seja aumentado, a seção de duto (arco da árvore) cujo diâmetro foi alterado recebe um *status* tabu e ingressa na lista tabu. A partir de então, a modificação do seu diâmetro fica proibida por certo número de iterações. O parâmetro de permanência na lista tabu foi implementado em duas versões: fixo ou aleatório. Para os valores fixos deste parâmetro, os arcos deixam a lista tabu após um determinado número de iterações. No caso aleatório, a cada iteração, cada seção presente na lista tabu tem uma probabilidade p de deixar a lista. Experimentos com os valores destes parâmetros foram realizados e o capítulo 6 descreve a abordagem mais efetiva para o problema estudado. Esta abordagem é utilizada em todos os testes finais reportados neste trabalho. Também um critério de aspiração foi usado para permitir que um movimento tabu seja usado se o mesmo levar a uma solução melhor que qualquer outra previamente encontrada.

4.1.5 Critério de Parada

O critério de parada utilizado foi um número máximo de iterações sem melhoria da melhor solução já encontrada. Sendo n o número de dutos de uma dada instância, o número máximo de iterações sem melhoria da melhor solução é de $2n$. Isto garante que para maiores instâncias do problema, proporcionalmente mais capacidade computacional será concedida.

4.1.6 Pseudo-código

O esquema básico do algoritmo busca tabu proposto neste trabalho pode ser visto no Quadro 4 a seguir:

```

solução_atual = gerar_solução_inicial ();
lista_tabu = {};
repetir
  se (nenhuma diminuição de diâmetro é possível) então
    solução_atual = diversificação (solução_atual);
  senão
    solução_nova = diminuir_diâmetro (solução_atual);
    se (duto reduzido não está na lista tabu ou critério de aspiração vale) então
      solução_atual = solução_nova;
    fim_se
  para cada duto  $i$  da lista tabu faça
    se (duto  $i$  já cumpriu requisitos de permanência na lista tabu) então
      remover duto  $i$  da lista tabu;
    fim_para
  inserir último duto alterado na lista tabu;
fim_se
até que critério de parada seja atingido

```

Quadro 4: Pseudo-código da busca tabu proposta para o PDD

5 Transgenética Computacional

A evolução das espécies pode ser vista como um conceito inspirador para algumas abordagens metaheurísticas, como, por exemplo, os algoritmos genéticos, que imitam a evolução Darwiniana, buscando a melhora da aptidão de uma população de indivíduos através da evolução pela aplicação de operadores de cruzamento e mutação na mesma (HOLLAND, 1975). Os algoritmos meméticos, por outro lado, esperam incrementar esta evolução através da utilização de mecanismos de busca local nos indivíduos da população – numa alusão à uma possível interferência da cultura no processo evolutivo, conceito que não mostra-se muito acurado. A transgenética computacional, por sua vez, possui sua fundamentação em outros mecanismos da evolução dos seres vivos que complementam o paradigma da evolução biológica como adotado na metáfora dos algoritmos genéticos, a saber, as alterações genéticas pela via intracelular e pela via epigenética (GOLDBARG *et al.*, 2001).

Os algoritmos transgenéticos pertencem à classe dos algoritmos evolucionários e apóiam sua metáfora na endosimbiose e propriedades do fluxo intracelular (GOLDBARG; GOLDBARG, 2002). A endosimbiose é uma teoria evolucionária que baseia sua formulação na união de indivíduos de naturezas distintas – de diferentes espécies – para a constituição de saltos adaptativos ou a formação de espécies híbridas ou novas (MOROWITZ, 1992). A endosimbiose enfatiza a possibilidade de inter-relações genéticas e funcionais entre diferentes indivíduos. Os algoritmos transgenéticos, adaptando os conceitos da endosimbiose e de outras características do fluxo intracelular ao contexto computacional, propõem a execução da evolução através de um processo de troca de informações realizado entre populações de espécies diferentes – via de regra uma população de cromossomos e populações de vetores transgenéticos. Os vetores transgenéticos, mimetizando atuadores microbiais e celulares, operacionalizam a obtenção e compartilhamento de informações genéticas. Essas informações são obtidas ao longo do processo evolucionário ou de fontes externas. As informações são compostas e transportadas pelos vetores transgenéticos com base no paradigma do “*Quorum Sensing*” que propõe a possibilidade de direcionamento e orquestração dos

vetores do contexto microbial e celular (WATERS; BASSLER, 2005). A proposta sugere, portanto, que a competição não representa a única via promotora do aperfeiçoamento genético.

O fluxo intracelular é um caminho aberto para interferências sobre o ADN, sendo, na microbiologia, a principal via de interferência na informação genética. O paradigma epigenético, por sua vez, diz respeito à co-evolução da cultura junto ao ADN e mais especificamente, à influência da primeira sobre o desenvolvimento do segundo. Estes dois paradigmas são essencialmente diferentes do paradigma extracelular no qual se baseiam os algoritmos genéticos, mas são responsáveis por boa parte da evolução observada na natureza, o que permite visualizar uma aplicação de boas perspectivas como metáfora para uma meta-heurística.

A metáfora da transgenética pode, portanto, ser computacionalmente vislumbrada, principalmente no intuito de se resolver problemas de otimização combinatória, de uma grande variedade de formas. De modo geral, pode-se conceber um algoritmo transgenético através de um mecanismo de vetores transgenéticos. Os vetores são responsáveis por operar sobre um conjunto de soluções para o problema abordado, transcrevendo seu código nas informações genéticas, em um formato compatível. Esses vetores podem ser obtidos de um banco de informações, contendo trechos de boas informações genéticas ou culturais. As sucessivas transcrições de boas informações pelos vetores levam à evolução da população.

O processo de evolução de um algoritmo transgenético se dá mediante a interação de uma população de cromossomos e um grupo de atuadores ditos vetores transgenéticos, consideradas as possibilidades de interferência de um dado meio ambiente. Um vetor transgenético, λ , é uma dupla $\lambda = (I, \Phi)$, onde I é uma cadeia de informação e Φ é o método de manipulação, $\Phi = (p_1, \dots, p_s)$, $p_j, j=1, \dots, s$, são procedimentos que definem a atuação do vetor. Ao infiltrarem sua cadeia de informação I os vetores transgenéticos provavelmente alterarão o cromossomo S_i e sua adequação. Em uma analogia à terminologia empregada pela microbiologia são previstos vários tipos de vetores transgenéticos. A Tabela 16 mais adiante resume os procedimentos que compõem o método de manipulação dos dois tipos de vetores mais comumente utilizados nos algoritmos transgenéticos (GOLDBARG *et al.*, 2006).

Um vetor λ é dito um plasmídeo quando sua cadeia de informação I é traduzida no formato genético – uma subcadeia de DNA – e seu método utiliza somente os procedimentos p_1 e p_2 . Um vetor λ é dito um transposon quando seu método utiliza os procedimentos p_1, p_2, p_3 e p_4 .

O processo de evolução endossimbiótica dos algoritmos transgenéticos é regulado através de regras que simulam a orquestração do “*Quorum Sensing*” e a atuação dos transposons, denominadas regras transgenéticas. No processo evolucionário transgenético três contextos podem interagir:

1. A população de cromossomos (supostamente células / microorganismos procarióticos).

2. A população de vetores transgenéticos que são representados pelos simbioses e transposons. Os simbioses são diversos tipos de microorganismos ou vetores capazes de incorporar novas informações genéticas às células. Os transposons são vetores capazes de atuar sobre a informação genética da célula alterando-a.

3. O contexto extracelular rico em informações ambientais capazes de influenciar simbioses, transposons e células.

p_1 Ataque (A)	Define o critério de avaliação que estabelece quando um cromossomo é suscetível à manipulação do vetor. $A: S_i \rightarrow \{\text{falso, verdadeiro}\} \ i=1,2,\dots,n$
p_2 Operador de Transcrição	Se $A(S_i) = \text{“verdadeiro”}$, o procedimento define como a informação I , transportada pelo vetor, será transferida para o cromossomo.
p_3 Operador de Identificação	Identifica posições que serão utilizadas para limitar a operação do vetor.
p_4 Operador de Recombinação	Identifica a origem e o comprimento de cada subcadeia de informação transportada pelo vetor de manipulação.

Tabela 16: Alguns procedimentos típicos dos vetores transgenéticos

O banco de informações ambientais é o repositório geral de informações. Portanto pode somar informações obtidas *a priori* com as que são alcançadas

durante a execução do processo evolucionário. As informações do banco podem ser codificadas tanto no formato genético como, por exemplo, soluções parciais do problema, quanto sob um formato abstrato representam outras formas de informação como, por exemplo, procedimentos ou regras de compartilhamento genético. O controle da evolução das populações de vetores, cromossomos e do meio ambiente é realizado através de três classes de regras denominadas “transgenéticas”. A Figura 4 localiza, no processo evolucionário transgenético, as principais etapas reguladas por essas regras.

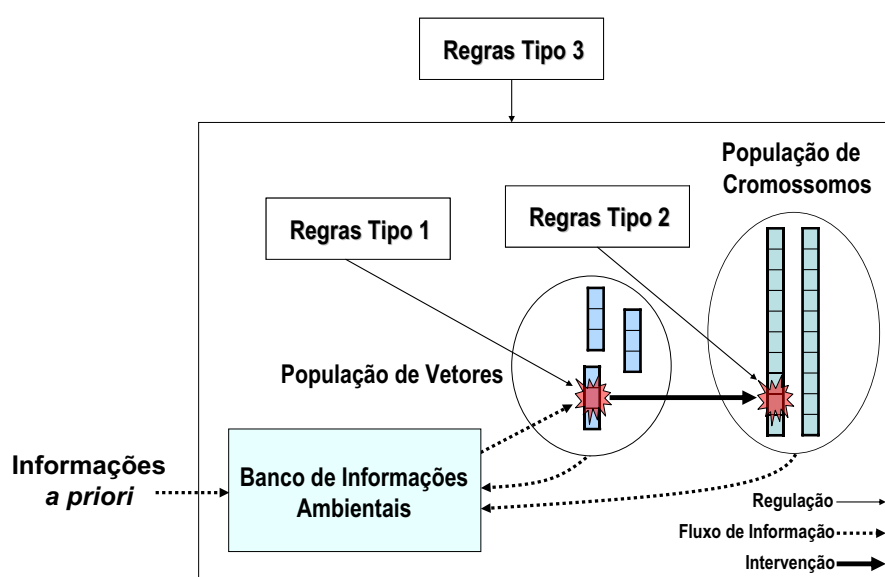


Figura 4: Evolução transgenética

As regras do tipo 1 dirigem a construção da cadeia de informação I que será transportada pelos vetores transgenéticos, vetores simbiotes e do fluxo intracelular. As regras tipo 1 podem utilizar qualquer conhecimento armazenado no banco de informações ambientais. As regras do tipo 2 definem como a informação I será transcrita no cromossomo – o operador que será utilizado por λ . As regras de transcrição podem evoluir em conformidade com a resistência à manipulação demonstrada pelos cromossomos. Em um algoritmo poderão existir várias regras do tipo 1 e 2. As regras tipo 3 orquestram o processo completo, desempenhando parte do papel atribuído ao “*Quorum Sensing*”, mas também acrescidas de outras especificidades necessárias à implementação computacional do processo. As regras tipo 3 definem que tipo de vetores serão empregados, quantos cromossomos serão

atacados em uma iteração, quantos vetores de manipulação serão criados, a condição de parada, etc.

Dentre as diversas maneiras de adaptar os conceitos da transgenética para a resolução computacional de problemas de otimização, talvez a mais comum seja através do algoritmo denominado Proto-Gene (ProtoG). Este algoritmo utiliza-se somente da evolução por associação. Estes algoritmos são modelados em duas fases. Na primeira delas, há a criação dos trechos de boa informação genética e a definição dos procedimentos de manipulação das soluções. Na segunda fase, os vetores transcrevem as informações ambientais nos indivíduos da população e o banco de informações pode ser realimentado com trechos dos melhores cromossomos, representando a co-evolução. Neste tipo de algoritmo, os vetores utilizados são os plasmídeos. Estes vetores são constituídos apenas de uma cadeia de informação e um método de transcrição da informação em um formato compatível com as informações genéticas dos indivíduos da população. A evolução da população é garantida pela regra de sensibilidade ao ataque do plasmídeo, que é a da estrita melhoria da adequação (GOLDBARG *et al.*, 2001). O esquema básico de um algoritmo ProtoG é descrito em forma de pseudocódigo no Quadro 5 abaixo:

```

gerar e avaliar uma população inicial de cromossomos;
carregar as regras transgenéticas;
gerar o banco de informações ambientais;
repetir
    gerar vetores transgenéticos a partir do banco de informações ambientais e das
        regras transgenéticas;
    para cada cromossomo da população faça
        se o cromossomo é sensível à manipulação então
            realizar manipulação conforme as regras transgenéticas;
            reavaliar o cromossomo manipulado;
            atualizar banco de informações e regras transgenéticas;
        fim_se
    fim_para
até que critério de parada seja atendido

```

Quadro 5: Esquema básico de um algoritmo ProtoG

5.1 Implementação

Para a solução do problema do passeio do pistoneio periódico foi desenvolvido e aplicado um algoritmo transgenético ProtoG. A metaheurística transgenética faz parte dos chamados algoritmos evolucionários, sendo baseada, dentre outros conceitos, em uma população de indivíduos e na endosimbiose (evolução por associação) que ocorre entre seres vivos de constituição mais simples.

As próximas seções descrevem em detalhes cada parte do algoritmo ProtoG implementado, a saber: o conceito de atratividade de um nó, a representação computacional das soluções, a geração da população inicial e do banco de informações, a seleção e o ataque dos vetores transgenéticos (plasmídeos), a regra de realimentação do banco de informações e os critérios de parada utilizados.

5.1.1 Atratividade de um Nó

Antes de entrar em detalhes sobre aspectos fundamentais do algoritmo ProtoG desenvolvido, é útil apresentar um conceito importante para a sua implementação. O conceito de atratividade de um nó é utilizado em vários pontos do algoritmo proposto e mede o quão bom é para a UMP visitar um determinado poço, dados a posição atual do veículo e o último dia no qual aquele poço foi visitado. A atratividade pode ser dada por:

$$A_i^{jk} = \frac{v_{ik}^2}{d_{ji}} \quad (5.1)$$

onde A_i^{jk} é a atratividade do nó i quando a UMP está no nó j e o poço i foi visitado pela última vez há k dias, v_{ik} é o volume atual do poço i dado que ele foi visitado pela última vez há k dias e d_{ji} é o tempo de viagem entre os nós j e i . Esta é uma das inúmeras possíveis definições para este conceito, mas é a que levou aos melhores resultados quando aplicada às instâncias do problema.

5.1.2 Representação das Soluções

As soluções para o PPPP são representadas computacionalmente por um vetor de rotas, uma para cada dia do horizonte de planejamento desejado. As rotas, por sua vez, indicam a seqüência de nós visitados a partir do nó 0 (que representa o depósito ou estação de tratamento de óleo) para um determinado dia. Este formato de representação pode ser visualizado na Figura 5 abaixo.

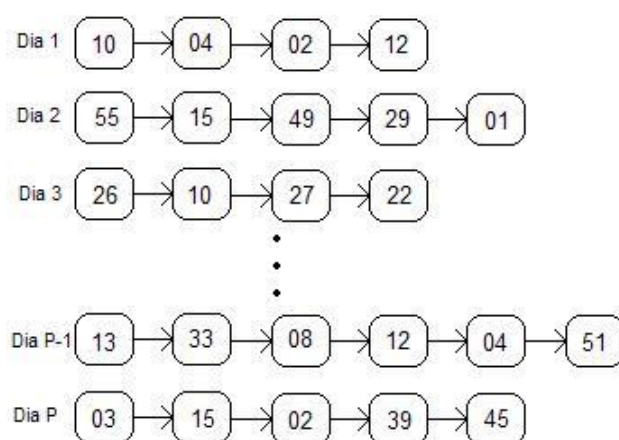


Figura 5: Representação de uma solução do PPPP

5.1.3 População Inicial

O algoritmo transgenético ProtoG trabalha em cima de uma população de cromossomos ou indivíduos (soluções para o problema). Nesta implementação, a população inicial é gerada aleatoriamente ou através de um procedimento guloso aleatório semelhante à primeira fase de um algoritmo GRASP (FEO; RESENDE, 1995). Quando gerada aleatoriamente, uma rota é construída para cada dia, iniciando-se do depósito e sorteando-se os poços a serem visitados até que o turno de trabalho da UMP esgote. O método guloso utilizado neste algoritmo, por sua vez, também partindo da estação de tratamento de óleo, insere seqüencialmente os poços em cada rota baseado no critério de atratividade (relacionado ao volume e à distância a partir do nó atual) do poço para cada dia. Para dar um caráter aleatório e possibilitar a geração de diferentes soluções para a população por este algoritmo, o

próximo poço a ser visitado é então escolhido com equi-probabilidade entre os dois poços ainda não visitados de maior atratividade a partir do depósito ou do poço atual.

5.1.4 Banco de Informações

O banco de informações é outra parte fundamental do algoritmo, por ser o maior responsável pelo sucesso da evolução da população ao armazenar trechos de boas soluções. Os trechos de ADN utilizados são constituídos por blocos de montagem formados por seqüências de poços, podendo ser aplicados à rota de qualquer dia no horizonte de planejamento. Os trechos de informação possuem comprimento variável entre 2 e 5 poços e são constituídos a partir das melhores soluções da população, que, espera-se, formam um bom conjunto de poços.

Desta forma, o banco de informações é gerado a partir da população inicial, extraindo-se desta as boas seqüências de poços que formarão os trechos de informação. Adicionalmente, sempre que há uma melhora no cromossomo campeão, busca-se extrair as boas seqüências do mesmo. De forma geral, o banco de informações sempre mantém os melhores trechos de ADN. Os trechos de informação do banco de informações são também avaliados por um critério de atratividade dado pela razão entre o volume total de óleo coletado nos poços e o tempo necessário para a coleta (a soma do tempo gasto no percurso e no pistoneio dos mesmos), considerando-se que os poços estejam cheios.

5.1.5 Plasmídeos

A cada iteração, um vetor transgenético é gerado e este tenta atacar todos os cromossomos da população, que podem resistir ao ataque. Os vetores transgenéticos utilizados nesta implementação são os plasmídeos, que possuem uma cadeia de informação e um método de contaminação. A cadeia de informação dos plasmídeos pode ser constituída de um trecho de ADN do banco de informações

(extraído das melhores soluções da população) ou ser gerada dinamicamente por um dos três procedimentos heurísticos descritos a seguir. Uma ou outra fonte de informação é sorteada com equi-probabilidade a cada passo do algoritmo. Caso os procedimentos heurísticos sejam sorteados, o método escolhido para ser aplicado é também sorteado, mas baseado em pesos ajustados dinamicamente e definidos pelo número de melhorias que aquele método foi capaz de realizar anteriormente. Os três pesos iniciais são iguais ao número de indivíduos da população, para evitar que o sucesso inicial de um dos métodos se sobreponha completamente às possíveis melhorias que os outros dois métodos podem vir a realizar.

O primeiro procedimento heurístico é um algoritmo guloso aleatório baseado na atratividade, semelhante ao utilizado na geração da população inicial. Ele é realizado essencialmente da mesma forma como se constrói uma rota para uma solução da população inicial, mas o número de poços no mesmo é limitado a 5. O segundo método sorteia um conjunto de 2 a 5 poços considerados de rápido re-enchimento, com probabilidade de sorteio de cada poço proporcional à taxa de re-enchimento do mesmo. Uma taxa de re-enchimento pode ser obtida dividindo-se o volume total de óleo do poço cheio pelo volume que o mesmo apresenta um dia após ter sido pistoneado. O terceiro procedimento reúne poços próximos, de forma a minimizar o tempo gasto no percurso entre os poços e maximizar o aproveitamento do tempo gasto no pistoneio em si. Os poços próximos são obtidos por um processo semelhante ao da construção de uma árvore geradora mínima pelo algoritmo de Prim. A árvore é enraizada em um nó sorteado e os demais poços vão sendo incluídos até que o tamanho do trecho atinja um valor aleatório, entre 2 e 5 poços.

Ao se atacar um indivíduo, o vetor avalia cada rota para testar se a mesma é sensível à manipulação. Uma rota será considerada sensível se não contiver algum dos poços presentes na informação do vetor. Repetidamente, uma das rotas sensíveis é selecionada de forma aleatória para ser manipulada pelo vetor, até que a manipulação de uma rota gere uma solução que não viola a restrição do turno de trabalho da UMP para os outros dias – encerrando com sucesso o ataque – ou até que a lista de rotas sensíveis acabe – quando o ataque falha. Em um ataque de um vetor a uma dada rota, os poços da seqüência ainda não incluídos na rota são adicionados através de uma heurística de inserção, descartando-se, se necessário, os poços de menor volume que não fazem parte do trecho de ADN. Finalmente, a solução atacada é mantida se for viável e melhor que a solução original.

5.1.6 Critérios de Parada

Alguns critérios de parada foram considerados e testados preliminarmente, entre eles, o tempo de execução do algoritmo. Entretanto, optou-se por não utilizar este critério para permitir que fossem observadas as condições de convergência da heurística proposta. O critério de parada adotado nos experimentos finais realizados, após o estudo feito na seção 6.2.2, é então um número máximo de 10.000 iterações do algoritmo ou 1.000 iterações sem melhoria da melhor solução da população.

5.1.7 Pseudo-código

O esquema geral dos algoritmos transgenéticos, conforme descrito no Quadro 5, pode então ser adaptado para a solução do problema do passeio do pistoneio periódico como mostrado no Quadro 6 abaixo. Inicialmente, gera-se uma população de cromossomos aleatoriamente ou através do procedimento guloso. O banco de informações é construído com trechos desta população inicial. As variáveis de controle da regra de parada e da seleção das heurísticas construtivas são então inicializadas. Então, até que o critério de parada seja atingido, seleciona-se um trecho de ADN para compor a informação do vetor transgenético. Este trecho pode ser obtido do banco de informações ou gerado dinamicamente através de uma das heurísticas construtivas. Se o método heurístico for o escolhido, uma dos três procedimentos disponíveis é selecionado aleatoriamente, com viés baseado nas melhorias que cada procedimento já pôde realizar até a iteração corrente. Em seguida, para cada cromossomo da população, busca-se uma rota passível de manipulação e realiza-se o ataque com o vetor transgenético, isto é, transcreve-se a informação genética na rota do cromossomo. Se o novo cromossomo gerado é melhor que o original, ele é mantido. Se ainda o novo cromossomo é a melhor solução já encontrada pelo algoritmo, todos os trechos de 2 a 5 poços do mesmo são avaliados e substituem os piores trechos do banco de informações se forem melhores pelo critério da atratividade, como descrito na seção 5.1.4; e o contador de iterações sem melhoria da melhor solução da população é reiniciado.

```

gerar e avaliar uma população inicial de cromossomos;
gerar o banco de informações com trechos de cromossomos da população inicial;
iterações  $\leftarrow$  0; sem_melhoria  $\leftarrow$  0;
heurística[1]  $\leftarrow$  n; heurística[2]  $\leftarrow$  n; heurística[3]  $\leftarrow$  n; melhorias  $\leftarrow$  3n;
repetir
  se random(0, 1) < 0,5 então
    ADN  $\leftarrow$  selecionar aleatoriamente um dos trechos do banco de informações;
  senão
    r  $\leftarrow$  random(0, melhorias); // número aleatório  $0 \leq r < \text{melhorias}$ 
    se r < heurística[1] então i  $\leftarrow$  1;
    senão se r < heurística[1] + heurística[2] então i  $\leftarrow$  2;
    senão i  $\leftarrow$  3;
    ADN  $\leftarrow$  gerar informação através da heurística i; // i = 1, 2 ou 3
  fim_se
  para cada cromossomo C da população de indivíduos faça
    rotas  $\leftarrow$  conjunto de rotas de C sensíveis à manipulação pelo plasmídeo;
    enquanto rota  $\neq$  {} e R' ainda não foi atribuído faça
      R  $\leftarrow$  selecionar um elemento aleatório do conjunto rotas;
      C'  $\leftarrow$  atacar(C, R, ADN); // inserir informações de ADN na rota R de C
      se C' é viável então R'  $\leftarrow$  R;
      senão remover elemento R do conjunto rotas;
    fim_enquanto
    se R'  $\neq$  nulo e C' é melhor que C então
      se C' é melhor que o melhor indivíduo da população atual então
        inserir trechos de C' no banco de informações;
        sem_melhoria  $\leftarrow$  0;
      fim_se
      C  $\leftarrow$  C'; heurística[i]  $\leftarrow$  heurística[i] + 1; melhorias  $\leftarrow$  melhorias + 1;
    fim_se
  fim_para
  iterações  $\leftarrow$  iterações + 1; sem_melhoria  $\leftarrow$  sem_melhoria + 1;
até que iterações > 10000 ou sem_melhoria > 1000

```

Quadro 6: Pseudo-código do algoritmo transgenético aplicado ao PPPP

6 Experimentos Computacionais

6.1 Problema do Dimensionamento de Dutos

6.1.1 Instâncias

Para propósitos de validação dos algoritmos propostos, os mesmos foram aplicados para a solução de 71 instâncias teóricas, com tamanhos variando entre 51 e 4461 nós e, conseqüentemente, de 50 a 4460 arcos ou seções de dutos. Estas instâncias são baseadas nas instâncias da TSPLIB – um banco de instâncias do problema do caixeiro viajante (REINELT, 1991) – e foram geradas como segue. Primeiro, a árvore geradora mínima do grafo completo obtido do problema do caixeiro viajante foi calculada e o primeiro nó fixado como a fonte do gás. Então, valores para o fluxo em cada duto e pressões mínimas requeridas por cada nó de demanda foram gerados aleatoriamente numa faixa previamente definida. Finalmente, um conjunto de 6 diferentes diâmetros foi escolhido a partir de opções comercialmente disponíveis na realidade. Cada instância do problema do dimensionamento de dutos é referida com o mesmo nome da instância do problema do caixeiro viajante que a originou. Maiores detalhes sobre estas instâncias podem ser obtidas nos trabalhos que as criaram originalmente, o de Costa (2004) e o de Castro (2004).

6.1.2 Algoritmo Exato

Inicialmente, são reportadas as soluções ótimas obtidas pelo algoritmo exato para o problema do dimensionamento de dutos. Este algoritmo foi testado em um micro-computador Pentium D de 2.8GHz com 2GB de memória RAM. Os resultados para todas as instâncias estudadas são resumidos na Tabela 17.

Instância	Tamanho	Custo Ótimo	Tempo (s)	Atribuições Parciais
eil51	51	3862,50	0,172	300
berlin52	52	62603,40	0,593	909
st70	70	5798,90	1,031	864
eil76	76	4768,90	1,265	1104
pr76	76	913473,10	1,437	1187
rat99	99	11402,10	2,735	1460
kroA100	100	193351,60	3,063	1473
kroB100	100	198357,40	2,563	1874
kroC100	100	189540,60	2,766	2074
kroD100	100	191538,80	1,359	1135
kroE100	100	197996,90	2,25	1632
rd100	100	71708,60	2,437	2004
eil101	101	5675,30	1,547	1176
lin105	105	134466,50	3,297	1794
pr107	107	392422,10	3,625	2694
pr124	124	532125,00	5,734	2326
bier127	127	975471,80	2,109	1841
ch130	130	53209,80	3,343	1681
pr136	136	932218,70	2,672	1712
pr144	144	539456,30	9,172	3879
ch150	150	60543,40	6,375	2661
kroA150	150	253311,10	9,703	3470
kroB150	150	234850,30	5,609	2511
pr152	152	658830,80	17,843	5482
u159	159	391681,80	8,75	3088
rat195	195	22196,50	10,25	3117
d198	198	128528,90	13,344	3624
kroA200	200	274397,80	11,953	3438
kroB200	200	273240,10	8,063	2996
ts225	225	1198600,00	15,062	4862
tsp225	225	36647,40	23,813	4967
pr226	226	763475,40	33,281	9456
gil262	262	21516,70	6,578	1791
pr264	264	473496,60	21,203	5440
a280	280	25080,50	20,047	5347
pr299	299	479309,90	40,641	8168
lin318	318	405353,80	24,015	6330
rd400	400	140471,40	18,359	5793
fl417	417	107799,80	53,391	14833
pr439	439	1053042,50	111,156	13187
pcb442	442	504545,90	45,906	7592
d493	493	331245,30	58,765	8519
u574	574	348626,20	79,828	9689
rat575	575	65209,40	37,266	6396
p654	654	361242,50	265,954	22460
d657	657	460791,80	59,469	9909
u724	724	414143,80	74,156	8353
rat783	783	86522,50	81,016	8096
dsj1000	1000	209307010,20	121,718	12229
pr1002	1002	2606168,60	166,828	11711
u1060	1060	2240625,70	190,672	12909

vm1084	1084	2306299,60	245,531	19087
pcb1173	1173	581985,40	345,078	20793
d1291	1291	521404,60	338,859	20710
rl1304	1304	2610784,50	514,218	23912
rl1323	1323	2910196,00	521,203	21410
nrw1379	1379	561381,10	188,781	12683
fl1400	1400	200500,30	839,75	43549
u1432	1432	1578549,50	157,375	12666
fl1577	1577	224361,20	1794,64	44459
d1655	1655	639834,30	409,657	20597
vm1748	1748	3869832,60	4208,656	71565
u1817	1817	609552,60	485,39	23434
rl1889	1889	3386116,40	934,469	29106
d2103	2103	1146729,80	16310,89	208064
u2152	2152	680370,90	657,922	24626
u2319	2319	2463620,00	255,531	16087
pr2392	2392	4201780,60	1350,797	35232
pcb3038	3038	1443186,20	1270,313	26633
fl3795	3795	323832,60	28714,08	139922
fni4461	4461	1898208,30	1365,312	36624

Tabela 17: Resultados do algoritmo exato para o PDD

As duas primeiras colunas da Tabela 17 indicam a instância resolvida e seu tamanho em número de nós respectivamente. A terceira coluna contém o custo da solução ótima da instância correspondente. A quarta coluna indica o tempo em segundos necessários para a resolução exata da instância pelo algoritmo proposto quando rodando no computador especificado. Finalmente, a última coluna mostra o número de atribuições parciais resultantes na lista equivalente ao término da execução do algoritmo. As soluções detalhadas são listadas no Apêndice 1.

Como pode ser observado, o tempo necessário para a resolução exata do algoritmo é bastante razoável para a maioria das instâncias; em particular para as instâncias pequenas, com menos de 600 nós, todas resolvidas em menos de 2 minutos. O gráfico da Figura 6 mostra a variação do tempo necessário para a execução do algoritmo em função do tamanho da instância. Pode-se verificar que, apesar do tempo de execução ser certamente dependente do tamanho da instância, esta não é a única variável. A dificuldade inerente da instância é mais importante neste sentido, como mostra o gráfico da Figura 7, com a variação do tempo de execução em função do número de atribuições parciais na lista equivalente final.

As instâncias mais difíceis, do ponto de vista do tempo computacional necessário para sua resolução, foram d2103 e fl3795, necessitando de cerca de 4 horas e meia e 8 horas, respectivamente, para o término do algoritmo.

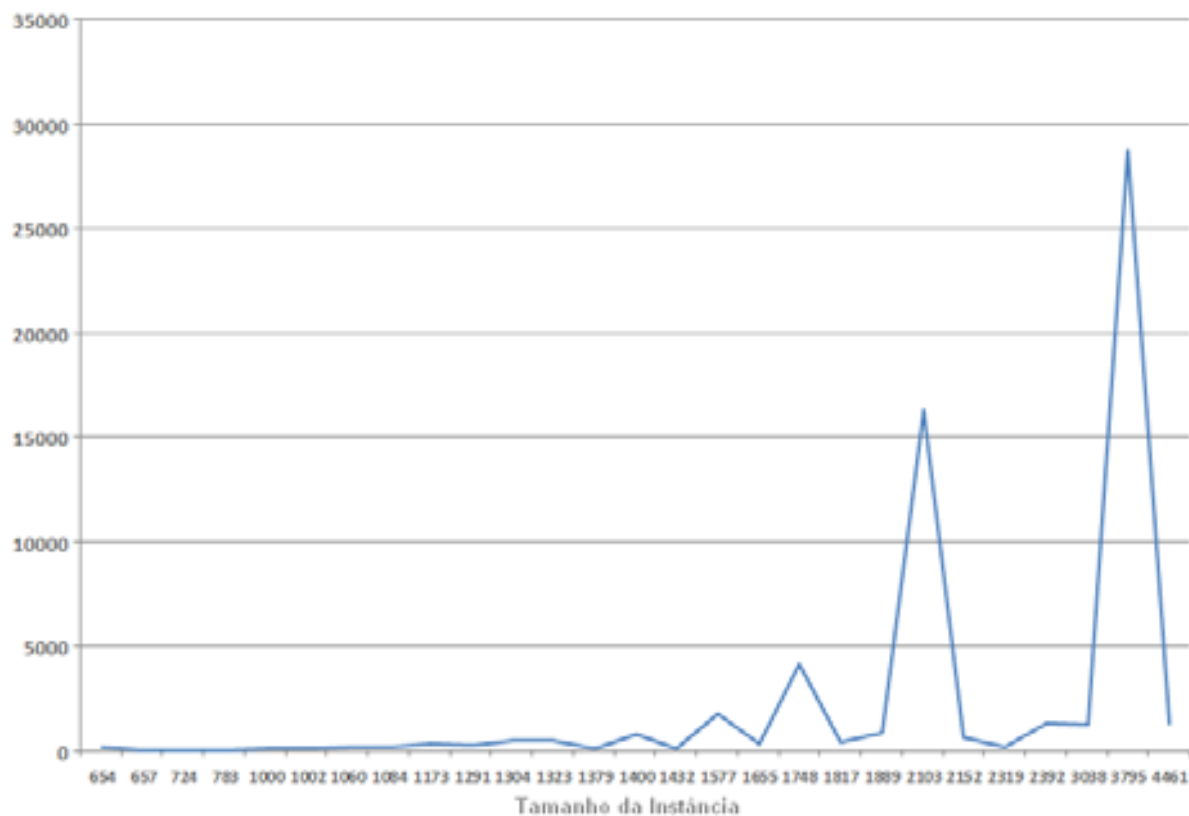


Figura 6: Variação do tempo de execução do algoritmo exato com o tamanho da instância

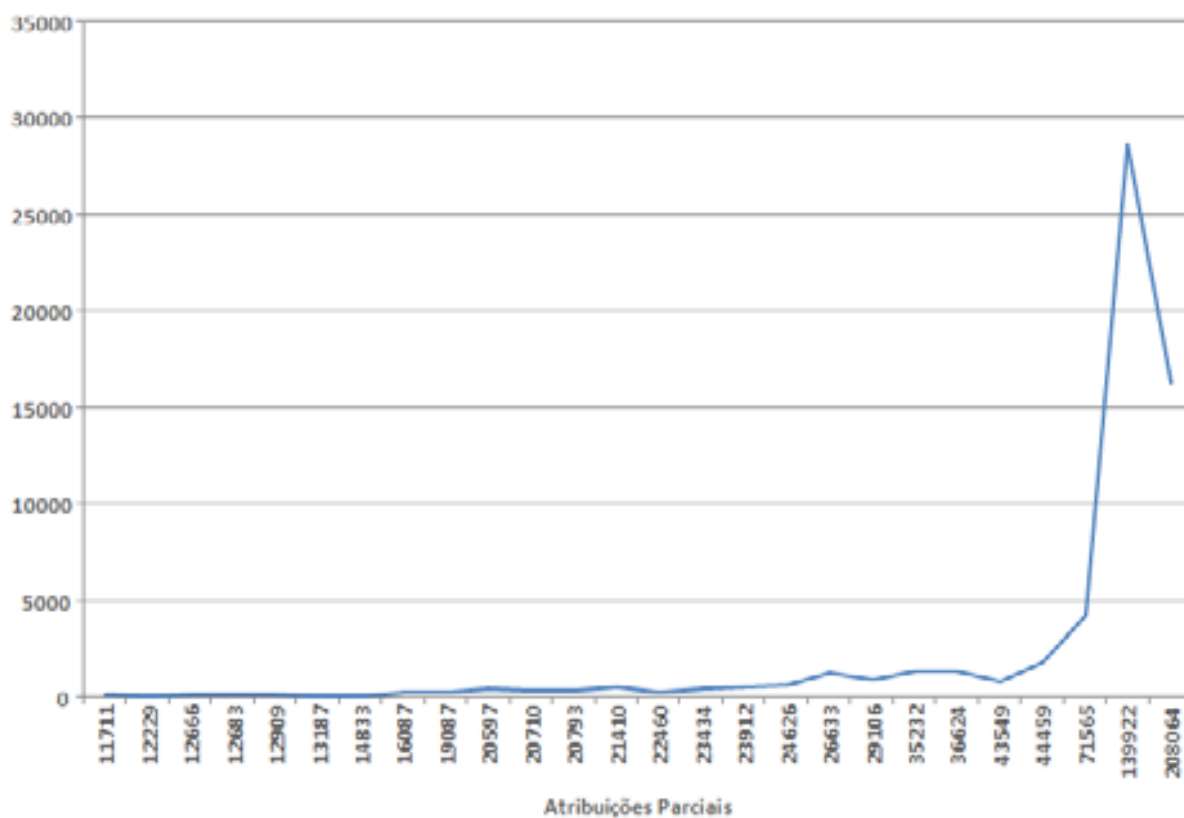


Figura 7: Variação do tempo de execução do algoritmo exato com o número de atribuições parciais

Finalmente, para exemplificar a evolução do número de atribuições parciais existentes no decorrer do algoritmo, as Figuras 8 e 9 mostram, respectivamente, os gráficos para o número de atribuições parciais em função da iteração atual para as instâncias *rd400* e *d2103*, respectivamente.

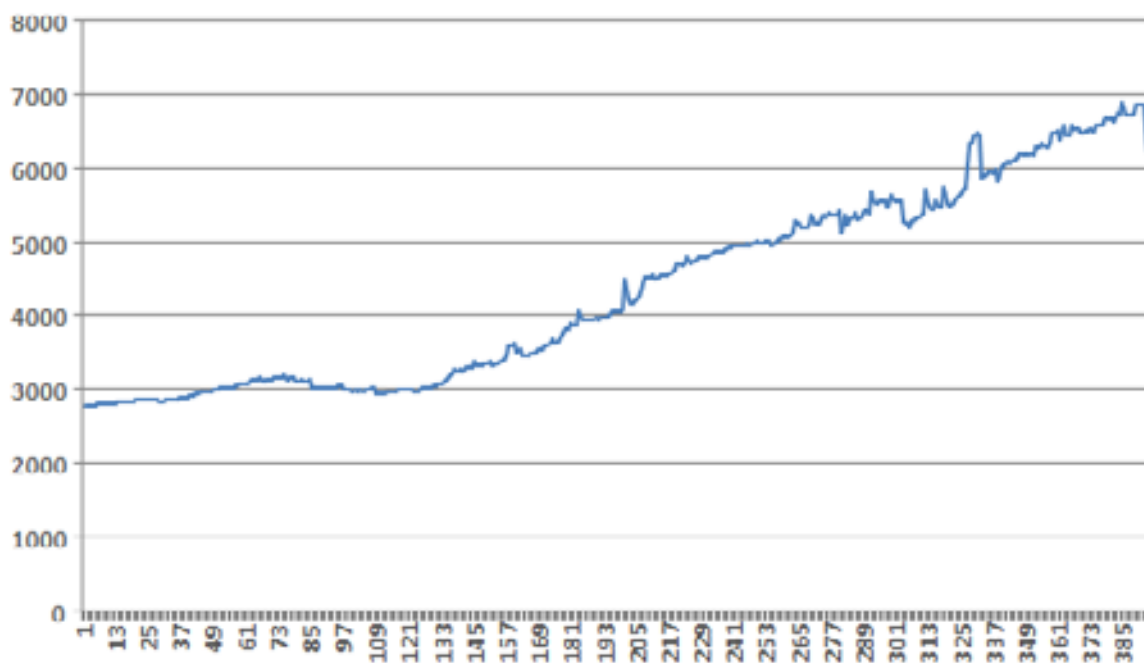


Figura 8: Número de atribuições parciais por iteração para instância *rd400*

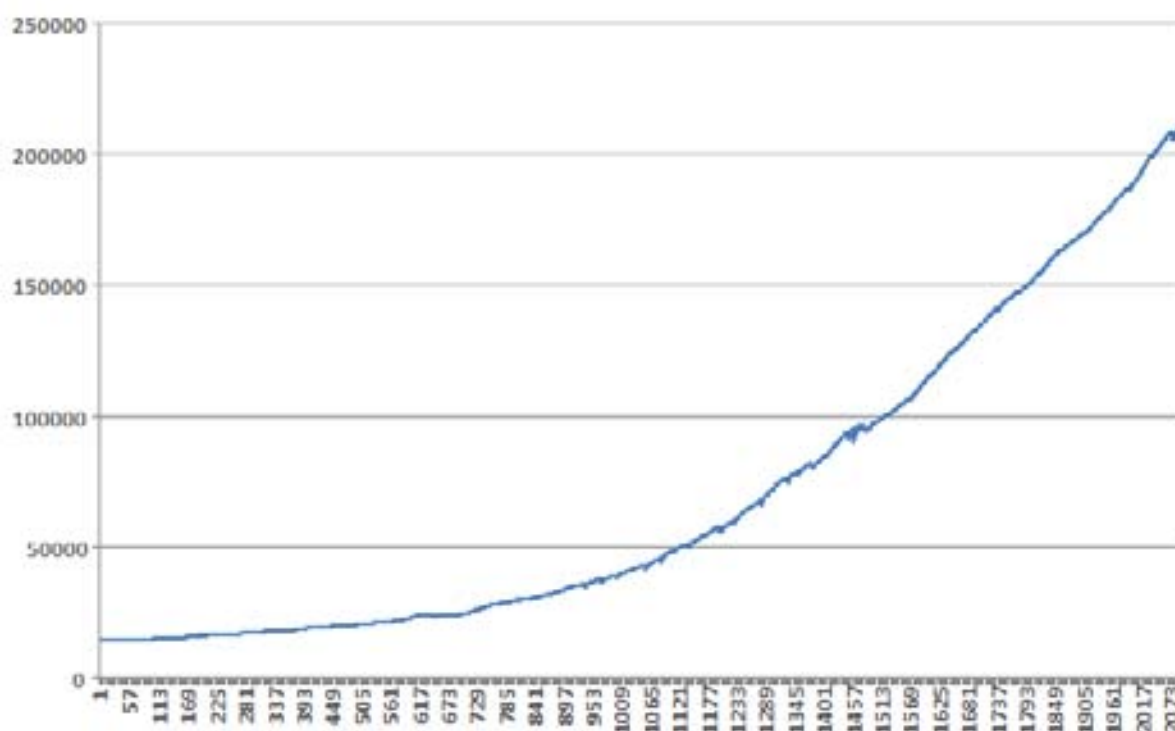


Figura 9: Número de atribuições parciais por iteração para instância *d2103*

Note-se que para a instância *rd400*, que pode-se considerar fácil devido ao tempo total de execução e número de atribuições parciais geradas, o crescimento das listas equivalentes é lento, aproximadamente linear, e o número de atribuições parciais chega até a decrescer em certos pontos da execução do algoritmo, destacadamente no final da mesma. Por sua vez, o crescimento do número de atribuições parciais para a instância *d2103*, visivelmente mais difícil, é bem mais regular e acentuado, aproximando-se no mínimo de uma curva quadrática e, mais provavelmente, de um crescimento exponencial com expoente não muito elevado.

6.1.3 Estudo dos Parâmetros

Foram realizados estudos dos parâmetros da busca tabu proposta e da principal metaheurística comparada, a busca tabu de Cunha e Ribeiro (2004). Para o algoritmo proposto foi analisado o parâmetro de tempo de permanência na lista tabu, em duas variantes: constante ou aleatório. Já para a metaheurística de Cunha e Ribeiro, foram estudados os desempenhos das duas versões do algoritmo (baseadas nas diferentes formas de selecionar o duto a ser modificado no procedimento de diversificação), bem como o parâmetro de permanência na lista tabu, apenas com valores constantes, pois a forma variável não é descrita no artigo.

O estudo dos parâmetros destes algoritmos foi realizado em um micro-computador Pentium D de 2.8GHz com 2GB de memória RAM. Os valores constantes testados para a permanência na lista tabu foram 5, 10, 15, 20, 25 e 30 iterações. Para o algoritmo proposto, ainda foi testada a permanência aleatória na lista tabu com probabilidade de 2,5%, 5%, 10%, 15%, 20% e 25%.

Os resultados para os experimentos com o algoritmo de Cunha e Ribeiro são ilustrados nas Figuras 10 e 11, que mostram a variação do *gap* médio obtido para cada valor de permanência na lista tabu para as instâncias escolhidas para realizar o estudo dos parâmetros: *pr439* e *p654*, respectivamente. Para cada valor do parâmetro estudado, o algoritmo foi executado 10 vezes, e a média dos custos obtida para as dez rodadas foi utilizada na escolha da versão 2 do algoritmo com a permanência na lista tabu por 20 iterações como a padrão para os demais testes, uma vez que os tempos computacionais foram próximos em todos os casos.

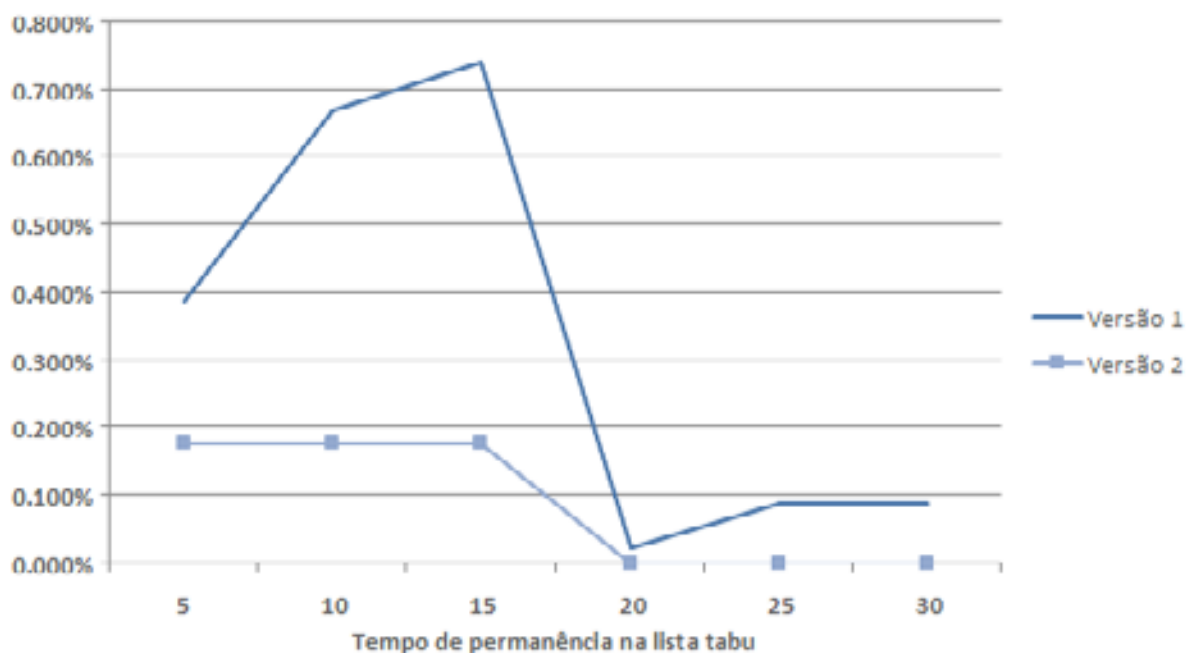


Figura 10: Gap médio da busca tabu de Cunha e Ribeiro para a instância *pr439*

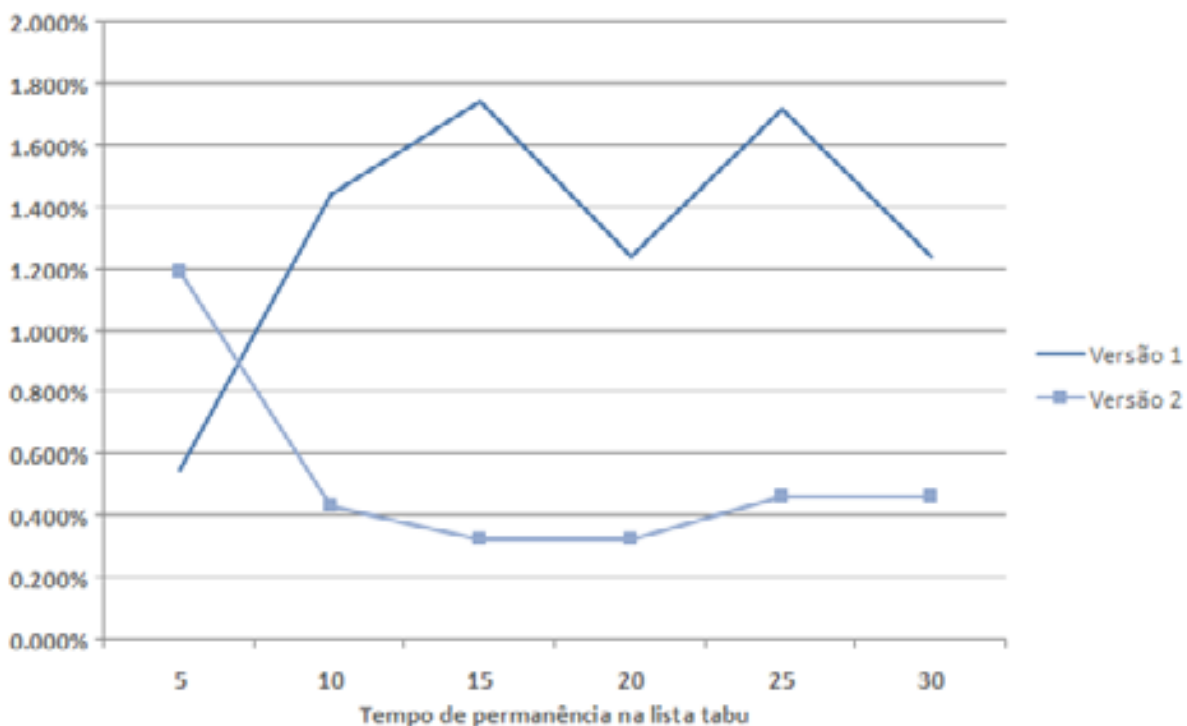


Figura 11: Gap médio da busca tabu de Cunha e Ribeiro para a instância *p654*

Nos testes com o algoritmo de busca tabu proposto, também foram realizadas 10 execuções para cada valor do parâmetro analisado e os resultados obtidos são mostrados nas Figuras 12 e 13. Como, também neste caso, os tempos

computacionais foram semelhantes, o *gap* médio foi o critério utilizado para se escolher uma permanência aleatória com probabilidade igual a 5% de deixar a lista tabu a cada iteração como o critério a ser utilizado em todos os testes adicionais, reportados na próxima seção.

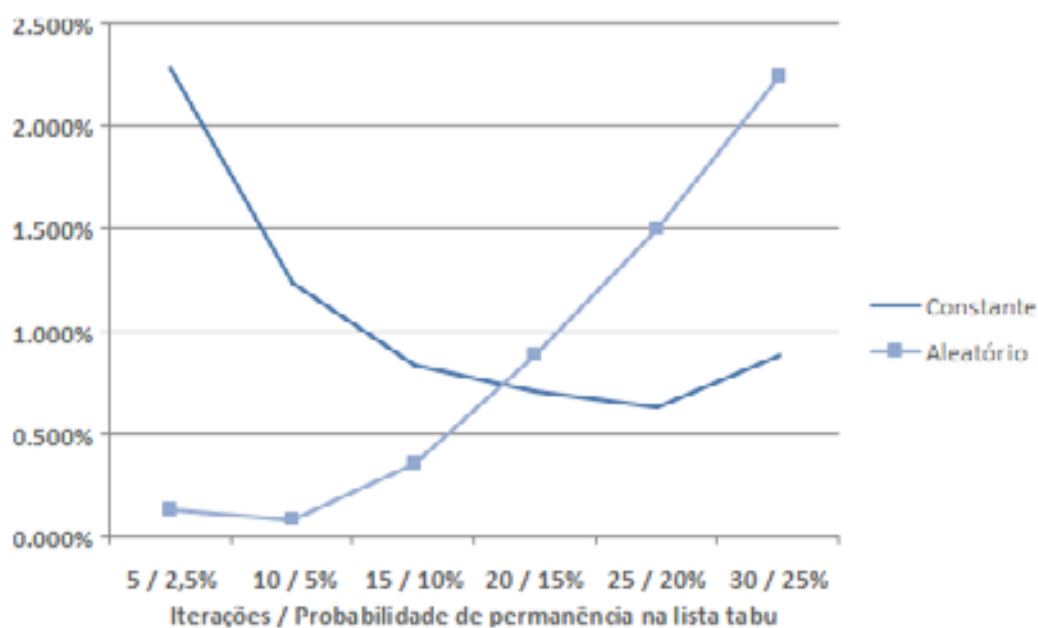


Figura 12: *Gap* médio da busca tabu proposta para a instância *pr439*

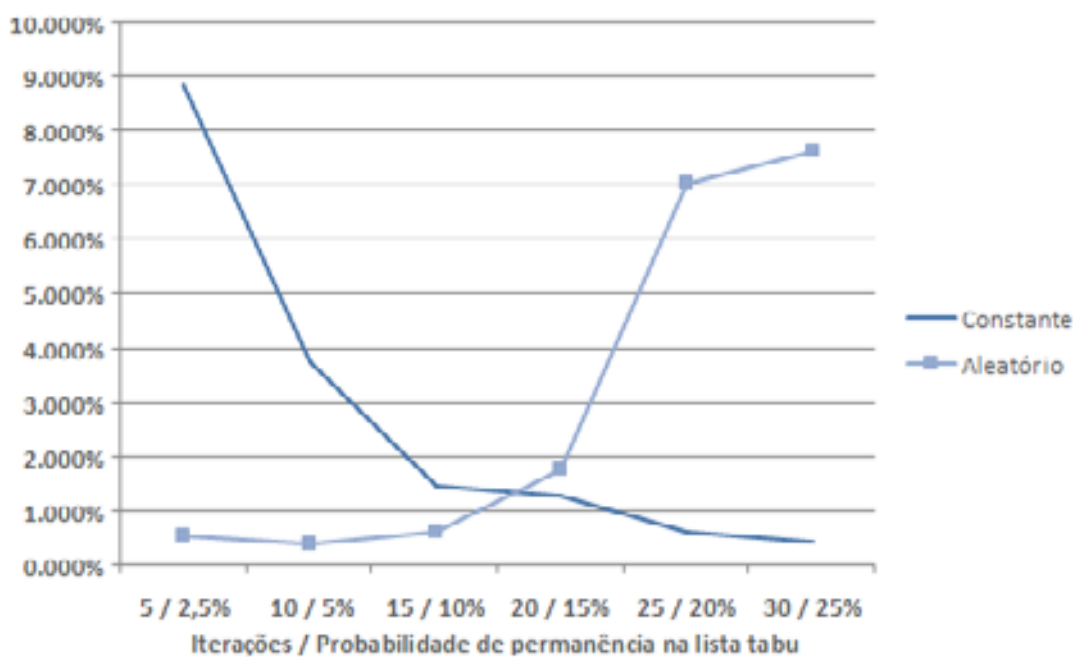


Figura 13: *Gap* médio da busca tabu proposta para a instância *p654*

Em relação ao critério de parada adotado no algoritmo de Cunha e Ribeiro (2004), o trabalho cita que diferentes testes com um número máximo de 10.000 iterações e um número máximo de 3.500 iterações sem melhoria da melhor solução já encontrada apresentaram resultados similares, mas o segundo critério trouxe resultados mais rápidos e é preferido. Considerando que o algoritmo possa rodar ainda mais rapidamente e produzir resultados comparáveis quando submetido a um critério de parada mais restritivo, avaliou-se a possibilidade de utilizar o mesmo critério de parada adotado para a busca tabu proposta, isto é, um número máximo de $2n$ iterações sem melhoria da melhor solução já encontrada ou um tempo máximo computacional de n segundos, onde n é o número de nós da instância. Os resultados obtidos para um conjunto de 40 instâncias de teste é mostrado na Tabela 18 mais adiante. A tabela mostra o *gap* e o tempo de execução médios para 10 rodadas com cada uma das duas condições de parada.

De acordo com este experimento estruturado em torno do critério de parada para o algoritmo do trabalho de Cunha e Ribeiro (2004), ao utilizar-se o mesmo critério de parada que a busca tabu proposta, o algoritmo comparado realiza menos iterações (pois $2n < 3500$ para todas as instâncias testadas) e, mesmo assim, consegue resultados no mínimo equivalentes num tempo bem mais reduzido do que quando submetido ao critério de parada original. Desta forma, constata-se que o algoritmo não produz melhora significativa se for dado um número máximo de iterações consideravelmente maior para sua execução. Isto é suficiente para mostrar que, ao menos para as instâncias estudadas neste trabalho, utilizar o mesmo critério de parada que a busca tabu proposta é não somente justo com o algoritmo comparado, mas é até benéfico para o mesmo, pois reduz o tempo computacional necessário para sua execução.

Para todas as instâncias mostradas na Tabela 18, o tempo computacional médio gasto com o critério de parada de $2n$ iterações é inferior ao tempo para o critério de parada de 3500 iterações, em alguns casos em torno de uma ordem de grandeza menor. Adicionalmente, em apenas 5 casos o algoritmo conseguiu algum ganho adicional nas iterações extras e a maior melhoria foi de apenas 0,26%. Finalmente, o experimento tende a ser confirmado em instâncias de tamanhos cada vez maiores, uma vez que para estes casos o valor de $2n$ se aproxima e até ultrapassa a quantidade fixa de 3500 iterações.

Instância	2n iterações		3500 iterações	
	Gap médio	Tempo médio (s)	Gap médio	Tempo médio (s)
pr76	0,00%	0,719	0,00%	5,594
kroA100	0,00%	1,141	0,00%	1,250
kroE100	0,00%	1,218	0,00%	1,250
lin105	0,00%	1,297	0,00%	1,468
pr107	0,00%	2,594	0,00%	13,515
pr124	0,00%	3,031	0,00%	17,140
bier127	0,00%	2,563	0,00%	2,563
pr136	0,00%	4,078	0,00%	16,234
pr144	0,00%	5,047	0,00%	27,078
kroA150	0,00%	7,235	0,00%	36,015
kroB150	0,00%	3,968	0,00%	4,000
pr152	0,00%	6,703	0,00%	34,875
u159	0,00%	6,953	0,00%	28,985
d198	0,00%	13,453	0,00%	41,531
kroA200	0,00%	13,141	0,00%	45,657
kroB200	0,00%	11,969	0,00%	36,734
ts225	0,00%	18,906	0,00%	57,844
tsp225	0,00%	12,328	0,00%	12,218
pr226	0,00%	19,453	0,00%	67,328
gil262	0,00%	20,984	0,00%	21,062
a280	0,00%	24,782	0,00%	24,750
pr299	5,25%	54,579	5,25%	162,437
lin318	0,00%	54,219	0,00%	120,640
rd400	0,00%	70,984	0,00%	70,797
fl417	0,00%	113,110	0,00%	167,609
pr439	13,20%	214,625	14,55%	451,641
pcb442	0,00%	154,203	0,00%	289,484
d493	27,20%	202,984	27,20%	355,609
u574	8,24%	321,359	8,24%	773,203
rat575	0,00%	264,329	0,00%	358,969
p654	27,64%	561,609	27,53%	857,031
d657	10,13%	450,297	9,87%	835,640
u724	10,55%	607,188	10,55%	834,032
rat783	3,56%	715,125	3,56%	906,484
dsj1000	21,10%	1000,790	21,06%	3460,297
pr1002	10,65%	1002,276	10,66%	2841,203
u1060	11,77%	1060,587	11,77%	2773,062
vm1084	11,53%	1084,377	11,50%	5175,984
pcb1173	29,01%	1173,494	29,11%	10604,860
d1291	80,31%	1292,106	80,17%	12579,328

Tabela 18: Estudo dos critérios de parada para a busca tabu comparada

6.1.4 Experimentos Finais para o PDD

A versão escolhida do algoritmo busca tabu de Cunha e Ribeiro (2004) e o algoritmo genético multi-objetivo de Surry *et al.* (1995) foram aplicados ao problema, mantendo todas as suas características básicas intactas, conforme já descrito na revisão da literatura. O algoritmo proposto é então comparado a estes. O critério de parada destes algoritmos foi definido como o mesmo do algoritmo proposto, ou seja, um número máximo de $2n$ iterações sem melhoria da melhor solução já encontrada ou um tempo computacional máximo de n segundos, onde n é o número de nós da instância. Para cada uma das 58 instâncias escolhidas para serem estudadas com estas abordagens, cada algoritmo foi executado 30 vezes em um computador Pentium IV 2.8 GHz com 512MB de RAM, exceto o algoritmo genético para instâncias com mais de 1.000 nós, pelas razões explicadas adiante.

A Tabela 19 mostra os resultados obtidos pelos algoritmos para as instâncias menores (com menos de 1.000 nós). A solução ótima encontrada pelo algoritmo exato desenvolvido bem como o afastamento percentual em relação à solução ótima (*gap*) mínimo e médio e o desvio padrão do afastamento obtidos pelos algoritmos testados para cada instância são apresentados. Foram omitidas 14 instâncias pequenas (com até 264 nós), uma vez que os três métodos alcançaram os mesmos resultados para estas, a saber, obtiveram a solução ótima para todas as rodadas. Apenas os resultados de BT-2 (CUNHA; RIBEIRO, 2004) são mostrados porque a performance da primeira versão mostrou-se inferior, conforme estudos de parâmetros. O algoritmo genético não encontrou nenhuma solução viável para as instâncias pr299 e u724. BT-P refere-se ao algoritmo proposto, enquanto BT-C e AG-S referem-se à busca tabu (CUNHA; RIBEIRO, 2004) e ao algoritmo genético (SURRY *et al.*, 1995) comparados, respectivamente. Os valores em negrito na tabela indicam que certo algoritmo obteve uma performance superior que os outros dois para aquela determinada grandeza na instância em questão.

A efetividade do algoritmo proposto é claramente demonstrada nesta primeira tabela. A busca tabu proposta encontrou a solução ótima para todas as instâncias com menos de 1.000 nós. Além disto, para as três instâncias em que a melhor solução não foi encontrada em todas as execuções, o afastamento médio não chegou a 0,5%.

Instância	Solução Ótima	BT-P			BT-C			AG-S		
		Gap Mín.	Gap Méd.	DP Gap	Gap Mín.	Gap Méd.	DP Gap	Gap Mín.	Gap Méd.	DP Gap
pr76	913473,10	0,00	0,00	0,00	0,00	0,00	0,00	3,50	4,56	0,58
kroA100	193351,60	0,00	0,00	0,00	0,00	0,00	0,00	2,65	4,43	0,95
kroE100	197996,90	0,00	0,00	0,00	0,00	0,00	0,00	1,23	2,52	1,45
lin105	134466,50	0,00	0,00	0,00	0,00	0,00	0,00	5,05	5,98	0,74
pr107	392422,10	0,00	0,00	0,00	0,00	5,09	27,88	0,34	0,94	0,23
pr124	532125,00	0,00	0,00	0,00	0,00	0,00	0,00	8,06	11,06	2,36
bier127	975471,80	0,00	0,00	0,00	0,00	0,00	0,00	0,79	1,54	1,07
pr136	932218,70	0,00	0,00	0,00	0,00	0,00	0,00	5,05	7,54	1,58
pr144	539456,30	0,00	0,00	0,00	0,00	5,40	29,57	5,62	9,10	1,50
kroA150	253311,10	0,00	0,00	0,00	0,00	6,64	5,14	12,73	18,45	3,73
kroB150	234850,30	0,00	0,00	0,00	0,00	0,00	0,00	2,02	2,92	0,64
pr152	658830,80	0,00	0,00	0,00	0,00	14,38	11,13	9,14	12,51	3,09
u159	391681,80	0,00	0,00	0,00	0,00	0,00	0,00	8,05	12,21	3,20
d198	128528,90	0,00	0,00	0,00	0,00	0,00	0,00	5,49	8,54	2,37
kroA200	274397,80	0,00	0,00	0,00	0,00	0,00	0,00	5,29	8,42	3,13
kroB200	273240,10	0,00	0,00	0,00	0,00	0,00	0,00	4,47	8,25	2,40
ts225	1198600,00	0,00	0,00	0,00	0,00	5,67	31,08	7,83	23,82	11,55
tsp225	36647,40	0,00	0,00	0,00	0,00	0,00	0,00	3,98	4,83	0,49
pr226	763475,40	0,00	0,00	0,00	0,00	0,00	0,00	14,71	25,60	6,73
gil262	21516,70	0,00	0,00	0,00	0,00	0,00	0,00	0,31	0,48	0,23
a280	25080,50	0,00	0,00	0,00	0,00	0,00	0,00	0,32	0,50	0,11
pr299	479309,90	0,00	0,00	0,00	5,25	5,25	0,00	-	-	-
lin318	405353,80	0,00	0,00	0,01	0,55	8,64	3,68	7,09	19,90	13,04
rd400	140256,80	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,47	5,03
fl417	107799,80	0,00	0,00	0,00	0,00	0,00	0,00	5,46	7,83	1,66
pr439	1053042,50	0,00	0,09	0,15	0,02	21,02	7,12	59,34	59,34	0,00
pcb442	504545,90	0,00	0,00	0,00	0,00	0,49	1,85	35,98	35,98	0,00
d493	331245,30	0,00	0,00	0,00	27,20	27,20	0,00	60,83	60,83	0,00
u574	348626,20	0,00	0,00	0,02	8,24	8,24	0,00	42,62	42,62	0,00
rat575	65209,40	0,00	0,00	0,00	0,00	0,00	0,00	6,45	8,21	0,46
p654	361242,50	0,00	0,43	0,11	26,76	27,13	2,03	48,40	48,40	0,00
d657	460791,80	0,00	0,02	0,11	5,73	10,28	24,61	67,83	67,83	0,00
u724	414143,80	0,00	0,00	0,01	10,55	10,55	0,00	-	-	-
rat783	86522,50	0,00	0,00	0,00	3,56	3,56	0,00	38,98	38,98	0,00

Tabela 19: Comparação da qualidade das soluções para 34 instâncias pequenas e médias do PDD

Por sua vez, a busca tabu de Cunha e Ribeiro (2004) também mostrou um desempenho consistente, mas não foi capaz de encontrar a solução ótima para 9 instâncias e seu afastamento médio foi pior do que o afastamento médio do algoritmo proposto para 15 destas instâncias. Os resultados também indicam que a performance da BT-C é superior a do AG-S, uma vez que para apenas duas instâncias a busca tabu apresentou um afastamento médio pior que o do algoritmo genético e este último encontrou apenas uma solução ótima, para a instância *rd400*.

Como o desempenho do algoritmo genético deteriora-se seriamente para as últimas instâncias da Tabela 19, o algoritmo proposto foi comparado somente com a busca tabu de Cunha e Ribeiro (2004) para as instâncias maiores. A Tabela 20

mostra os resultados destes dois algoritmos para 10 instâncias com mais de 1.000 nós no mesmo formato da Tabela 19. É evidente que o algoritmo proposto supera o método de referência uma vez que ele obtém o menor afastamento mínimo e médio para todas as redes. Adicionalmente, a robustez do algoritmo é visível, pois o afastamento médio da melhor configuração é sempre inferior a 9% e o desvio padrão excede 1% em apenas um dos casos. Pode ser visto que a BT-C falha em obter soluções competitivas para instâncias grandes, possivelmente pelo fato de este algoritmo efetuar apenas pequenas mudanças na solução a cada iteração.

Instância	Solução Ótima	BT-P			BT-C		
		Gap Mín.	Gap Méd.	DP Gap	Gap Mín.	Gap Méd.	DP Gap
pr1002	2662871,80	2,18	3,35	0,82	12,57	15,13	14,40
u1060	2302421,10	2,76	4,35	0,97	14,73	23,05	25,40
vm1084	2324953,00	0,81	1,91	0,85	12,21	13,44	7,57
pcb1173	591012,00	1,55	2,85	0,74	32,08	43,88	22,14
d1291	528294,10	1,32	2,02	0,74	81,13	86,78	12,71
rl1304	2701706,20	3,48	5,37	0,78	52,27	64,09	20,11
rl1323	3134737,40	7,72	8,95	0,76	14,75	14,75	7,72
nrw1379	561381,10	0,00	0,25	0,28	70,30	71,09	0,46
fl1400	202637,50	1,07	2,78	1,08	89,48	89,54	1,11
u1432	1584001,50	0,35	0,67	0,20	114,25	114,98	0,77

Tabela 20: Comparação da qualidade das soluções para 10 instâncias grandes do PDD

A boa qualidade das soluções apresentadas pelo algoritmo proposto nas tabelas anteriores já pode consistentemente apontar para sua utilidade. Este fato é ratificado na Tabela 21, onde os tempos de execução médios (em segundos) são dados para cada instância. Os tempos do AG-S são omitidos porque este método utilizou todo o tempo disponível para todas as execuções de todas as instâncias.

Uma vez mais, o algoritmo proposto mostrou bons resultados, gastando, para instâncias menores, menos que um décimo do tempo gasto pela BT-C e notavelmente menos tempo nos outros casos. É relevante notar que o critério de parada de tempo máximo de execução dificilmente foi necessário para o algoritmo proposto. Ademais, a BT-C mostra um resultado pobre para as 12 maiores instâncias, provavelmente por causa do tempo máximo de execução permitido, que foi alcançado em todas as execuções para estas instâncias. O tempo necessário para o algoritmo exato encontrar a solução ótima para as instâncias relatadas também é mostrado, como forma de avaliar a eficiência da busca tabu proposta para a obtenção de boas soluções para este problema.

Instância	Algoritmo Exato	BT-P	BT-C
pr76	1,437	0,119	1,076
kroA100	3,063	0,039	0,852
kroE100	2,250	0,039	0,865
lin105	3,297	0,043	0,895
pr107	3,625	0,313	3,620
pr124	5,734	0,406	4,403
bier127	2,109	0,063	1,801
pr136	2,672	0,494	5,736
pr144	9,172	0,634	8,450
kroA150	9,703	0,923	10,774
kroB150	5,609	0,090	2,730
pr152	17,843	0,966	9,238
u159	8,750	0,744	7,889
d198	13,344	1,149	17,115
kroA200	11,953	1,027	23,772
kroB200	8,063	1,200	38,465
ts225	15,062	1,694	31,311
tsp225	23,813	0,202	9,578
pr226	33,281	2,158	17,576
gil262	6,578	0,271	14,643
a280	20,047	0,312	17,470
pr299	40,641	8,923	86,404
lin318	24,015	10,880	65,950
rd400	18,359	1,560	49,649
fl417	53,391	5,026	174,955
pr439	111,156	62,854	218,321
pcb442	45,906	16,048	213,174
d493	58,765	18,130	354,374
u574	79,828	36,154	504,909
rat575	37,266	6,559	464,208
p654	265,954	72,481	654,269
d657	59,469	26,071	642,325
u724	74,156	91,008	724,510
rat783	81,016	26,242	783,501
pr1002	166,828	557,313	1002,387
u1060	190,672	918,610	1060,277
vm1084	245,531	944,113	1084,526
pcb1173	345,078	1032,390	1173,583
d1291	338,859	456,914	1292,244
rl1304	514,218	1227,100	1304,494
rl1323	521,203	1291,161	1323,406
nrv1379	188,781	1117,667	1379,547
fl1400	839,750	462,311	1401,127
u1432	157,375	1251,755	1433,146

Tabela 21: Tempo médio (em segundos) de execução dos algoritmos para o PDD

Uma vez que um algoritmo exato e soluções ótimas globais são conhecidas para este problema, é importante avaliar se os diferentes métodos heurísticos são adequados para a resolução do problema. Da Tabela 21, pode-se observar que a busca tabu proposta consegue obter as soluções ótimas para todas as instâncias com menos de 1.000 nós – com exceção da instância *u724* – em um tempo

computacional muito inferior ao necessário para o algoritmo exato. Para as instâncias maiores, entretanto, mostra-se viável utilizar o algoritmo exato – já que o mesmo precisou de menos tempo de execução em 9 das 10 instâncias com mais de 1.000 nós – desde que tenha-se disponível uma máquina com memória suficiente para a execução do mesmo.

Através das análises desenvolvidas, o algoritmo busca tabu proposto demonstra ser muito efetivo ao tratar as dificuldades do problema. Ainda, os algoritmos comparados são de reconhecido valor para esta classe de problema, então o estudo pode concluir que este é de fato um método adequado para este tipo de problema de otimização. Isto é razoavelmente esperado, uma vez que abordagens por busca tabu têm apresentado bons resultados em outros problemas relacionados, como o problema de otimização de redes de distribuição de água, estudado por Cunha e Ribeiro (2004).

6.2 Problema do Passeio do Pistoneio Periódico

6.2.1 Instâncias

Foram construídas 27 instâncias para o problema do passeio do pistoneio periódico a partir das instâncias do problema do caixeiro viajante assimétrico (PCVA) da TSPLIB (REINELT, 1991). As distâncias entre os nós da instância foram utilizadas para refletir o tempo de viagem entre cada par de poços. Entretanto, calculou-se o menor caminho entre cada par de nós, de forma a garantir que os tempos de viagem respeitem a desigualdade triangular. Finalmente, os tempos de percurso entre cada par de poços são ajustados de forma a que, em média, a distância entre um poço e outro seja igual a 30 minutos.

Optou-se por considerar um turno de trabalho da UMP de 16 horas (960 minutos) e um horizonte de planejamento de 30 dias. Valores aleatórios foram gerados, equiprovavelmente dentro de faixas de valores obtidas junto a Petrobrás, para o tempo de re-enchimento de 99,9% do volume máximo dos poços (entre 7 e 30 dias) e para o volume máximo (coluna estática) dos poços (entre 4 e 12 barris).

Os tempos de instalação e desinstalação do equipamento durante a visita a cada poço são fixados em 3 minutos cada. Adicionalmente, estabeleceu-se o tempo médio de pistoneio de um barril de óleo em 22,5 minutos. Da forma como foram geradas, as instâncias devem produzir no horizonte de planejamento citado um volume esperado de 1.080 e não maior que 1.280 barris de petróleo.

Para obter o parâmetro τ (indicativo da velocidade do reenchimento para cada poço) da equação 3.1, como citado, foi gerado aleatoriamente o tempo necessário para que o poço recuperasse 99,9% do volume máximo (entre 7 e 30 dias). Logo, τ é obtido como segue:

$$\Delta V = V_{\max} (1 - e^{-t/\tau}) = 0,999 \cdot V_{\max} \quad (6.1)$$

$$(1 - e^{-t/\tau}) = 0,999 \quad (6.2)$$

$$1 - 0,999 = e^{-t/\tau} = 0,001 \quad (6.3)$$

$$\frac{-t}{\tau} = \ln(0,001) \quad (6.4)$$

$$\tau = -\frac{t}{\ln(0,001)} \cong \frac{t}{6,907755} \quad (6.5)$$

onde t nesta última fórmula é o número de dias necessários para que o poço recupere 99,9% do volume máximo (novamente, entre 7 e 30 dias).

6.2.2 Estudo dos Parâmetros

Foram realizados estudos de ajuste dos parâmetros envolvidos, a saber: a geração da população inicial, o tamanho da população de indivíduos e o tamanho do banco de informações e os resultados são reportados a seguir. Em todos os testes de parâmetros, os experimentos foram realizados em um micro-computador Pentium IV 2.8GHz com 512MB de RAM. Para estes testes, foram escolhidas 9 instâncias (um terço do total), representativas das famílias de instâncias, e para cada valor dos parâmetros o algoritmo foi executado 10 vezes para cada uma das instâncias, tendo como critério de parada um tempo de execução de 300 segundos.

Inicialmente foi testada a geração da população inicial através do método totalmente aleatório ou do procedimento guloso, ou ainda dando igual chance a cada

elemento da população de ser gerado por um ou outro método. Os resultados são apresentados na Tabela 22 abaixo, que mostra a melhor solução e a média das soluções encontradas para cada um dos valores dos parâmetros.

Instância	Aleatório		Construtivo		Misto (equiprovável)	
	Vol. Máximo	Vol. Médio	Vol. Máximo	Vol. Médio	Vol. Máximo	Vol. Médio
br17	1041,64	1032,67	1022,06	1018,41	1033,70	1023,15
ft53	1134,19	1125,77	1110,36	1108,27	1114,76	1112,02
ftv150	1165,95	1164,28	1181,66	1180,70	1181,22	1179,47
ftv33	1134,05	1126,73	1129,51	1126,31	1128,68	1125,80
ftv70	1165,26	1160,64	1170,92	1169,62	1170,28	1168,58
kro124	1154,01	1149,62	1165,22	1162,98	1164,13	1162,27
p43	1217,32	1209,12	1217,51	1213,60	1215,76	1212,83
rbg323	1096,31	1093,15	1077,66	1075,01	1083,90	1077,42
ry48	1125,76	1122,45	1137,25	1134,82	1136,16	1133,90

Tabela 22: Volumes máximos e médios para as diferentes populações iniciais

Pode-se observar que os resultados são diferentes para as diferentes famílias de instâncias, não aparentando relacionar-se com o tamanho das mesmas. Desta forma, optou-se por utilizar as soluções iniciais aleatórias para as famílias *br*, *ft* e *rbg*; e gerar a população inicial pelo método construtivo para as famílias *ftv*, *kro*, *p* e *ry*. Apesar da instância *ftv33* ter apresentado resultados melhores com a geração aleatória, optou-se por utilizar o método construtivo também para esta porque os volumes médios foram muito próximos e para padronizar a família de instâncias.

Em seguida, foram testados os tamanhos da população e do banco de informações, conforme os mesmos critérios para o teste da geração da população inicial. Os tamanhos testados para a população variaram de 100 a 5.000 e os volumes máximos e médios obtidos são mostrados nas Tabelas 23 e 24. Por sua vez, os tamanhos testados para o banco de informações variaram de 500 a 6.000 e os resultados são exibidos nas Tabelas 25 e 26.

Com base nos resultados obtidos, também se optou por utilizar valores diferentes para as diferentes famílias de instâncias, uma vez que nenhum valor para o tamanho da população destacou-se dos demais para todas as instâncias. Assim, foi escolhido o tamanho de 1.000 cromossomos para a família *br*, 500 para a família *kro*, 400 para a família *ft*, 200 para as famílias *ftv*, *p* e *ry*, e 100 para a família *rbg*. Às famílias *ftv* e *ry* foi atribuído o tamanho 200 principalmente pelo valor médio dos volumes obtidos, uma vez que o valor máximo pode não ser representativo do conjunto de execuções do algoritmo. Optou-se mais uma vez por padronizar o valor

do parâmetro para uma mesma família e, assim, os resultados da instância *ftv150* para o tamanho 200 foram considerados suficientes para a escolha deste valor.

Instância	Tamanho da População							
	100	200	300	400	500	1000	2000	5000
br17	1034,61	1040,52	1040,71	1036,71	1044,08	1048,72	1038,57	1024,13
ft53	1128,98	1126,88	1131,90	1132,85	1129,34	1126,06	1122,41	1108,51
ftv150	1167,47	1167,38	1167,15	1170,86	1169,09	1167,76	1166,94	1159,64
ftv33	1133,30	1133,34	1135,09	1136,94	1136,09	1133,66	1125,47	1112,80
ftv70	1165,45	1169,02	1168,10	1169,41	1170,78	1165,63	1161,41	1149,99
kro124	1151,67	1153,14	1154,91	1152,36	1156,77	1156,56	1149,06	1142,27
p43	1216,69	1218,26	1216,59	1213,52	1213,47	1213,54	1212,14	1198,64
rbg323	1099,78	1095,49	1092,43	1091,10	1091,63	1082,77	1074,02	1054,00
ry48	1131,70	1130,92	1133,67	1130,58	1127,39	1127,82	1122,90	1115,20

Tabela 23: Volumes máximos para os vários tamanhos da população

Instância	Tamanho da População							
	100	200	300	400	500	1000	2000	5000
br17	1030,84	1031,60	1033,38	1031,78	1034,28	1035,45	1032,31	1018,48
ft53	1123,37	1122,97	1124,47	1125,45	1125,04	1119,99	1116,63	1105,37
ftv150	1164,71	1165,39	1165,97	1167,33	1166,04	1165,28	1164,54	1157,04
ftv33	1127,63	1130,55	1129,81	1128,44	1127,80	1123,37	1118,89	1108,17
ftv70	1160,95	1164,91	1163,46	1163,14	1162,11	1161,53	1157,28	1146,57
kro124	1148,21	1149,89	1150,73	1149,92	1150,94	1150,30	1147,50	1139,08
p43	1205,72	1208,77	1208,67	1205,61	1205,70	1204,79	1203,90	1189,97
rbg323	1094,54	1092,59	1090,18	1087,94	1088,87	1079,28	1068,85	1047,40
ry48	1122,62	1125,53	1124,12	1124,77	1123,62	1121,92	1117,12	1109,96

Tabela 24: Volumes médios para os vários tamanhos da população

Instância	Tamanho do Banco de Informações						
	500	1000	2000	3000	4000	5000	6000
br17	1045,17	1038,26	1032,86	1034,49	1031,27	1025,57	1025,53
ft53	1125,95	1128,79	1139,45	1133,60	1130,82	1135,05	1132,62
ftv150	1165,73	1166,81	1171,66	1169,65	1169,99	1171,02	1170,27
ftv33	1140,93	1132,80	1134,67	1139,89	1130,96	1128,88	1129,21
ftv70	1164,79	1165,39	1167,77	1168,97	1166,93	1170,21	1166,40
kro124	1150,47	1151,82	1156,15	1156,26	1154,27	1155,04	1154,41
p43	1215,50	1217,22	1221,13	1215,79	1224,05	1221,17	1221,46
rbg323	1089,90	1097,04	1101,41	1103,98	1106,10	1107,46	1105,73
ry48	1126,81	1128,18	1134,63	1130,46	1129,43	1130,14	1132,34

Tabela 25: Volumes máximos para os vários tamanhos do banco de informações

Instância	Tamanho do Banco de Informações						
	500	1000	2000	3000	4000	5000	6000
br17	1034,68	1031,37	1028,14	1026,47	1024,56	1022,93	1021,11
ft53	1120,62	1121,95	1128,81	1125,54	1125,54	1126,80	1126,91
ftv150	1162,56	1164,64	1167,40	1167,96	1167,83	1168,56	1168,29
ftv33	1125,36	1124,64	1129,22	1126,69	1126,80	1123,69	1121,92
ftv70	1158,65	1162,59	1164,13	1162,65	1163,26	1162,10	1161,76
kro124	1145,61	1149,49	1151,78	1153,77	1151,96	1152,34	1151,27
p43	1199,35	1207,54	1208,96	1209,43	1209,32	1214,08	1210,89
rbg323	1087,23	1093,56	1099,78	1102,18	1104,33	1104,81	1104,24
ry48	1120,19	1125,14	1124,88	1126,07	1122,46	1124,74	1125,42

Tabela 26: Volumes médios para os vários tamanhos do banco de informações

Para a escolha do tamanho do banco de informações, os critérios utilizados foram os mesmos dos outros parâmetros e, semelhantemente, vários valores foram selecionados para as diversas famílias de instâncias. Assim, o tamanho do banco de informações para a família *br* foi fixado em 500, para as famílias *ft* e *ftv* em 2.000, para as famílias *kro* e *ry* em 3.000, e para as famílias *p* e *rbg* em 5.000.

Por fim, pode-se ver nas Figuras 14 e 15 abaixo, a evolução do algoritmo transgenético implementado em uma aplicação, respectivamente, às instâncias *ftv100* e *ry48*, escolhidas para demonstrar a efetividade do critério de parada selecionado. Pode-se claramente observar que a escolha baseada no número máximo de iterações e/ou número de iterações sem melhoria da melhor solução da população é razoável, pois há uma estagnação na melhoria do algoritmo após certo ponto. Isto significa que o algoritmo converge rapidamente e não existe motivo para continuar o investimento de tempo. Desta forma, a escolha está justificada e sua sensibilidade analisada, não sendo, portanto, arbitrária.

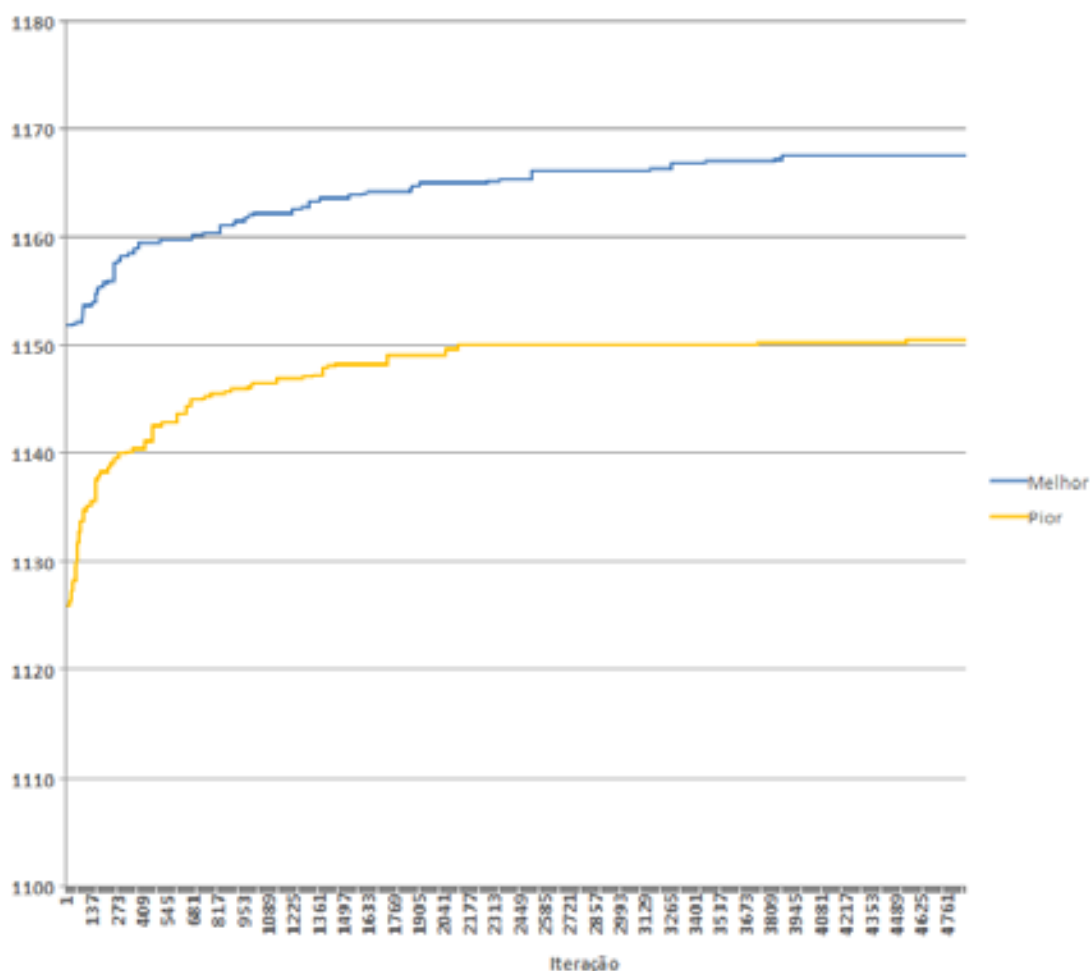


Figura 14: Evolução do algoritmo transgenético para a instância *ftv100*

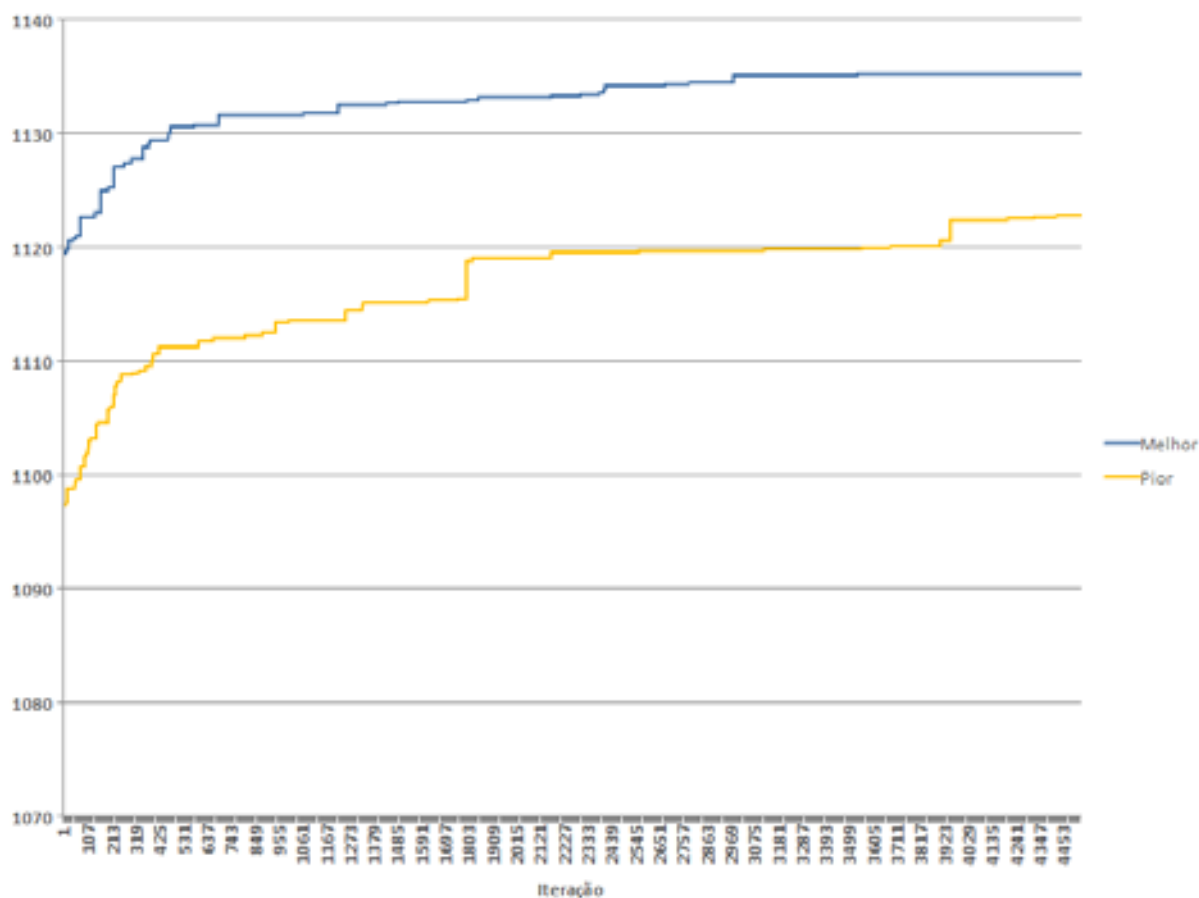


Figura 15: Evolução do algoritmo transgenético para a instância *ry48*

6.2.3 Experimentos Finais para o PPPP

Os experimentos computacionais finais com a versão implementada do algoritmo transgenético aplicado ao problema do passeio do pistoneio periódico foram realizados em um microcomputador Pentium IV 3.2GHz com 1GB de RAM. Nestes experimentos, o algoritmo foi executado 30 vezes para cada uma das 27 instâncias testadas. Em todos os casos, o critério de parada adotado foi um número máximo de 10.000 iterações ou 1.000 iterações sem melhoria do melhor indivíduo da população.

Os resultados obtidos por esta implementação são então comparados na Tabela 27 abaixo com o limite superior gerado pelo método da programação dinâmica. São apresentados o valor do limite superior e dos volumes máximos e mínimos obtidos pelo algoritmo transgenético no decorrer das 30 rodadas, bem

como o *gap* percentual (afastamento da solução em relação ao limite superior) para o volume máximo encontrado, o desvio-padrão do volume obtido no conjunto de rodadas, e o tempo médio gasto pelo algoritmo em cada uma das instâncias.

Considerando que o limite superior desenvolvido é bastante conservador, pois elimina do problema a complexidade do roteamento ao contar apenas a menor aresta chegando ou saindo de um nó, pode-se verificar que os resultados reportados pelo algoritmo transgenético desenvolvido são promissores. Para quase todas as instâncias foi possível obter um roteamento que leva a um volume com menos de 10% de diferença para o limite teórico, sendo que, em média, este afastamento foi de apenas 6%. Os resultados médios obtidos também colaboram para verificar a consistência do método, que apresentou desvios-padrão muito reduzidos, nunca alcançando 5 barris ou 0,5% do total do volume.

Instância	Limite Superior	Algoritmo Transgenético ProtoG					
		Volume Máximo	Gap Vol. Máximo	Volume Mínimo	Volume Médio	DP do Volume	Tempo Médio (s)
br17	1234,28	1047,29	-15,15%	1028,97	1037,56	4,24	288,28
ft53	1229,88	1127,47	-8,33%	1113,53	1118,72	3,20	105,31
ft70	1177,38	1099,74	-6,59%	1082,73	1091,46	3,53	93,32
ftv100	1240,35	1168,97	-5,75%	1162,63	1165,75	1,46	35,95
ftv110	1240,35	1168,58	-5,79%	1164,50	1165,91	0,90	34,30
ftv120	1243,51	1176,92	-5,36%	1173,16	1175,05	1,03	37,27
ftv130	1243,59	1174,97	-5,52%	1171,15	1172,51	1,02	35,21
ftv140	1243,59	1179,06	-5,19%	1172,02	1175,01	1,44	32,26
ftv150	1243,62	1181,29	-5,01%	1177,21	1179,44	0,90	31,49
ftv160	1243,65	1178,13	-5,27%	1174,36	1176,55	1,00	38,02
ftv170	1243,73	1183,14	-4,87%	1178,39	1180,69	1,08	35,72
ftv33	1224,33	1129,01	-7,79%	1122,89	1125,60	1,45	26,28
ftv35	1228,08	1129,95	-7,99%	1125,07	1127,52	1,36	28,29
ftv38	1228,86	1140,88	-7,16%	1134,81	1137,02	1,47	27,90
ftv44	1231,49	1141,08	-7,34%	1133,65	1136,13	1,58	27,28
ftv47	1222,80	1128,71	-7,69%	1122,99	1125,46	1,47	29,19
ftv55	1235,60	1141,98	-7,58%	1137,18	1139,30	1,16	31,34
ftv64	1240,80	1167,26	-5,93%	1164,11	1165,54	0,98	29,36
ftv70	1240,92	1172,97	-5,48%	1165,91	1168,86	1,41	27,14
ftv90	1240,27	1166,04	-5,98%	1159,85	1162,62	1,71	33,57
kro124	1230,87	1164,64	-5,38%	1160,87	1162,41	0,86	76,91
p43	1242,64	1214,27	-2,28%	1209,62	1212,56	1,05	28,77
rbg323	1178,90	1105,74	-6,21%	1095,46	1099,88	2,54	40,76
rbg358	1156,46	1099,23	-4,95%	1091,48	1094,44	1,75	40,22
rbg403	1140,30	1133,16	-0,63%	1126,19	1129,45	1,65	40,66
rbg443	1244,27	1236,88	-0,59%	1229,14	1232,37	2,04	46,78
ry48p	1223,47	1137,67	-7,01%	1130,12	1133,71	1,43	29,36

Tabela 27: Resultados obtidos pelo algoritmo transgenético ProtoG

Ao confrontar os resultados individuais com a estrutura das instâncias, é possível verificar fatos que corroboram com a afirmação de que o limite superior é bastante conservador e, portanto, os resultados apresentados pelo algoritmo são de boa qualidade. Para a instância *br17*, por exemplo, na qual os resultados obtidos mais se afastam do limite teórico, as rotas usualmente envolvem um maior número de poços, acima de 10 ao dia, diferentemente das demais instâncias, nas quais só são visitados cerca de 5 ou 6 poços diariamente. Logo, espera-se que o limite teórico para esta instância seja mais afastado, pois a perda de informação é multiplicada pelo número de poços visitados a cada dia. Por outro lado, nas instâncias *rbg403* e *rbg443*, há uma estrutura contendo *clusters* nos quais um grande número de poços estão muito próximos (a 1 minuto uns dos outros), e outros poços estão bastante afastados. Assim, é de se esperar que o roteamento aproveite a proximidade dos poços e consiga um maior volume de óleo, o que é refletido no melhor desempenho do algoritmo para estas instâncias quando comparado com o limite teórico, pois há relativamente pouca perda de informação ao sempre utilizar a menor aresta que chega ou deixa um nó.

Adicionalmente, o tempo de execução mostrou-se bem reduzido e de uma ordem de grandeza razoável (normalmente em torno de 30s) para permitir a utilização do algoritmo em situações práticas. Vale ressaltar que o tamanho das instâncias estudadas também alcança níveis satisfatórios para a aplicação em casos reais, já que o número de poços encontrados nas maiores instâncias é certamente superior ao que pode ser efetivamente aproveitado pela operação de uma única UMP. Desta forma, pode-se concluir que a heurística desenvolvida atende aos requisitos originais da pesquisa.

7 Conclusões e Sugestões para Trabalhos Futuros

No presente trabalho, dois importantes problemas da área de produção e distribuição de petróleo e gás natural foram abordados. Formulações matemáticas na forma de problemas de programação linear inteira mista foram fornecidas tanto para o problema do dimensionamento de dutos em uma rede urbana de distribuição de gás natural como para o problema do passeio da unidade móvel do pistoneio periódico.

O primeiro problema – o do dimensionamento de dutos – foi resolvido de forma exata utilizando-se uma adaptação do método de fusão paralela e serial, de Rothfarb *et al.* (1970), através da consideração das restrições de *upstream*, o que permitiu determinar a solução ótima de todas as instâncias disponíveis do mesmo em um tempo bastante razoável. A principal restrição deste algoritmo é a sua grande utilização de memória para o armazenamento das listas de atribuições parciais, mas foi possível obter resultados exatos para instâncias de até 4460 dutos num tempo computacional da ordem de horas em um computador com 2GB de memória RAM.

Um algoritmo busca tabu foi também proposto e implementado para a resolução deste problema. Suas características especializadas permitiram a obtenção de soluções de alta qualidade em um tempo de execução bastante reduzido. Isto demonstra a capacidade da metaheurística de lidar com as restrições complexas deste problema de otimização combinatória. A busca tabu proposta foi comparada com outros dois algoritmos metaheurísticos, a saber, uma outra busca tabu originalmente desenvolvida para a otimização dos diâmetros em uma rede de distribuição de água e um algoritmo genético especializado no tratamento das complexas restrições do problema em questão. Os experimentos computacionais realizados consolidam as conclusões obtidas de que a busca tabu proposta é realmente eficaz na solução do problema estudado.

Para a resolução do problema do passeio do pistoneio periódico, por sua vez, foi desenvolvido um algoritmo transgenético ProtoG, que utilizou-se de um número de diferentes mecanismos para geração de bons trechos de informação, e pode-se concluir que, através dos procedimentos utilizados e do fino ajuste dos parâmetros envolvidos, os resultados obtidos são bastante promissores, pois a qualidade das soluções foi satisfatória e estas foram obtidas em um tempo

computacional que permite a aplicação na prática da abordagem. Sendo um problema inédito na literatura de programação matemática e metaheurísticas, não foram feitas comparações com outros métodos. O problema é ainda de difícil solução exata, por se tratar de uma variação do problema de roteamento de veículos seletivo, conhecidamente NP-árduo. Deste modo, também se pode apontar como contribuição ao estudo do problema o desenvolvimento de um limite superior teórico para o volume de óleo coletado, computacionalmente eficiente, que pode ser facilmente aplicado a outros problemas de roteamento seletivo. A comparação dos resultados obtidos pelo algoritmo transgenético proposto com os valores do conservativo limite superior também sustentam as conclusões de eficácia do algoritmo apresentado, pois o *gap* percentual mínimo foi inferior a 9% para todas as instâncias testadas exceto uma, e o desvio padrão mostrou-se bastante reduzido.

Um grande número de linhas de investigação podem ser derivadas do presente trabalho. Vale, portanto, destacar algumas delas que mostram-se mais promissoras que as demais. Em relação ao problema do dimensionamento de dutos, propõe-se analisar como continuidade deste trabalho a aplicação de variações da metaheurística busca tabu, como a recente busca tabu com vizinhança variável (SEMET *et al.*, 2005), a busca tabu reativa ou ainda a hibridização com outros métodos, como a metaheurística *simulated annealing*. Outras alternativas não abordadas neste trabalho para o gerenciamento de memória a curto e longo prazos também podem levar a bons resultados, como, por exemplo, o uso de uma taxa variável de permanência na lista tabu (mas não necessariamente aleatória).

Confirmando-se e aperfeiçoando-se os resultados promissores do trabalho corrente, trabalhos futuros podem ser encaminhados considerando incrementos nos modelos das redes de distribuição, como por exemplo, estudando redes contendo malhas (laços) ou adicionando novos equipamentos ao modelo da rede de distribuição, como bombas e compressores, tornando-o o mais próximo possível das aplicações na realidade.

Já com respeito ao problema do passeio do pistoneio periódico, a evolução deste trabalho pode se dar, por exemplo, através do desenvolvimento e testes de novas metaheurísticas, como algoritmos genéticos e meméticos, para o problema apresentado. Melhorias para o algoritmo proposto podem ser obtidas através de diferentes esquemas de ataque, incluindo diferentes plasmídeos, que não puderam ser testados neste trabalho, ou através de mais experimentação dos parâmetros da

abordagem. Um outro teste promissor consiste em mudar a distribuição dos ataques dos vetores de um sorteio equiprovável entre vetores do banco de informações e vetores dos procedimentos heurísticos para um esquema que favoreça os procedimentos heurísticos gerarem as cadeias no início do algoritmo e dê mais chances para o banco de informações no final do processo evolucionário. Outra possibilidade é melhorar o banco de informações já no início do algoritmo pela inserção de boas soluções heurísticas no mesmo.

A continuidade do trabalho também poderá se dar pela aplicação do algoritmo já desenvolvido a outros problemas de roteamento de veículos, principalmente a versão seletiva e a versão periódica deste. Adaptações do algoritmo e novos estudos de parâmetros serão necessários para estas novas aplicações e os resultados obtidos poderão ser confrontados com os de outros algoritmos da literatura para os referidos problemas de roteamento como forma de validação do método ora proposto.

Ainda como trabalho futuro, sugere-se a extensão do modelo do presente trabalho para a utilização de mais de uma unidade móvel de pistoneio; a criação e teste de novos e diferentes vetores, bem como um estudo mais aprofundado dos possíveis mecanismos de ataque e transcrição de informação. A utilização de novas regras de construção de trechos de ADN, baseadas em heurísticas para problemas de roteamento periódicos, também demonstram bom potencial de sucesso.

Referências Bibliográficas

AARTS, E., KORST, J. **Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing**. Chichester: John Wiley and Sons, 1989.

AARTS, E., LENSTRA, J. K. **Local search in combinatorial optimization**. In: Series in Discrete Mathematics and Optimization. Wiley-Interscience, 1997.

ALEGRE, J., LAGUNA, M., PACHECO, J. Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts. **European Journal of Operational Research**, 2005.

ANGELELLI, E., SPERANZA, M. G. The periodic vehicle routing problem with intermediate facilities. **European Journal of Operational Research**, v. 137, p. 233-247, 2002.

ANP – Agência Nacional de Petróleo. Disponível em: <http://www.anp.gov.br>. Acesso em: 16/10/2006.

BACKER, B., FURNON, V. Metaheuristics in constraint programming experiments with Tabu Search on the vehicle routing problem. **2nd International Conference on Metaheuristics**, 1997.

BAPTISTA, S., OLIVEIRA, R. C., ZÚQUETE, E. A period vehicle routing case study. **European Journal of Operational Research**, v. 139, p. 220-229, 2002.

BARROS, C. A. Uma aplicação do GRASP na otimização do emprego da unidade móvel de pistoneio. Dissertação (Mestrado em Sistemas e Computação). Universidade Federal do Rio Grande do Norte, 2001.

BOYD, I. D., SURRY, P. D., RADCLIFFE, N. J. Constrained gas network pipe sizing with genetic algorithms. University of Edinburgh, 1994.

BOYD, E. A., SCOTT, L. R., WU, S. Evaluating the quality of pipeline optimization algorithms. 1997. Disponível em: <http://www.psig.org/paper/1997/9709.pdf>.

BRASSARD, G., BRATLEY, P. **Fundamentals of algorithmcs**. Englewood Cliffs: Prentice-Hall, 1996.

BUTT, S. E., RYAN, D. M. An optimal solution procedure for the multiple tour maximum collection problem using column generation. **Computer and Operation Research**, p. 427-441, 1999.

CAMPELLO, R. E., MACULAN, N. **Algoritmos e heurísticas**: desenvolvimento e análise de performance. Niterói: Universidade Federal Fluminense, 1994.

CASTILLO, L., GONZÁLEZ, A. Distribution network optimization: finding the most economic solution by using genetic algorithms. **European Journal of Operational Research**, v. 108(3), p. 527-537, 1998.

CASTRO, M. P. Algoritmos evolucionários para o problema de dimensionamento dos dutos de uma rede urbana de distribuição de gás natural. Dissertação (Mestrado em Sistemas e Computação). Universidade Federal do Rio Grande do Norte, 2004.

CHRISTOFIDES, N., BEASLEY, J. E. The period routing problem. **Networks**, v. 14, p. 237-256, 1984.

COELLO, C. A. C., BECERRA, R. L. Efficient evolutionary optimization through the use of a cultural algorithm. **Engineering Optimization**, v. 36(2), p. 219-236, 2004.

CORDEAU, J-F., GENDREAU, M., LAPORTE, G. A tabu search heuristic for periodic and multi-depot vehicle routing problems. **Networks**, v. 30, p. 105-119, 1997.

CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., STEIN, C. **Algoritmos**: teoria e prática. 2. ed. Rio de Janeiro: Campus, 2002.

CORNUÉJOLS, G., HARCHE, F. Polyhedral study of the capacitated vehicle routing problem. **Mathematical Programming**, v. 60, p. 21-52, 1993.

COSTA, W. E. Um estudo algorítmico para o problema do dimensionamento de dutos em uma rede urbana de distribuição de gás natural. Dissertação (Mestrado em Sistemas e Computação). Universidade Federal do Rio Grande do Norte, 2004.

CUNHA, M. C., RIBEIRO, L. Tabu search algorithms for water network optimization. **European Journal of Operational Research**, v. 157(3), p. 746-758, 2004.

DANTAS, N. M. Metaheurística busca tabu aplicada a solução do problema da unidade móvel do pistoneio. Dissertação (Mestrado em Sistemas e Computação). Universidade Federal do Rio Grande do Norte, 2001.

DORIGO, M., MANIEZZO, V., COLORNI, A. The ant system: optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man and Cybernetics**, v. 26(B), p. 29-41, 1996.

DRUMMOND, L. M. A, OCHI, L. S., VIANNA, D. S. A parallel hybrid evolutionary metaheuristic for the period vehicle routing problem. **Lecture Notes in Computer Science**, v. 1586, p. 183-191, 2001.

EDMONDS, J. Matroids and the greedy algorithm. **Mathematical Programming**, v. 1, p. 127-136, 1971.

FEO, T., RESENDE, M. Greedy randomized adaptive search procedures. **Journal of Global Optimization**, v. 6, p. 109 -133, 1995.

FERNANDES, A. C. Algoritmo transgenético para o problema do passeio do pistoneio. Dissertação (Mestrado em Sistemas e Computação). Universidade Federal do Rio Grande do Norte, 2001.

FERREIRA, I. Contaminações virais na aceleração da solução do problema do passeio do pistoneio. Dissertação (Mestrado em Sistemas e Computação). Universidade Federal do Rio Grande do Norte, 2000.

GAREY, M. R., JOHNSON, D. S. **Computers and intractability**: a guide to the theory of NP-completeness. New York: W. H. Freeman & Co., 1979.

GAUDIOSO, M., PALETTA, G. A heuristic for the periodic vehicle routing problem. **Transportation Science**, v. 26(2), 1992.

GENDREAU, M., HERTZ, A., LAPORTE, G. A tabu search heuristic for the vehicle routing problem. **Management Science**, v. 40, p. 1276-1290, 1994.

GLOVER, F. Heuristic for integer programming using surrogate constraints. **Decision Science**, p. 156-166, 1977.

GLOVER, F. Tabu search methods in artificial intelligence and operations research. ORSA, **Artificial Intelligence Newsletter**, v. 1, 1987.

GLOVER, F., Laguna, M. **Tabu search**. Dordrecht: Kluwer Academic Publishers, 1997.

GOLDBARG, M. C., LUNA, H. **Otimização combinatória e programação linear**: modelos e algoritmos. Rio de Janeiro: Campus, 2000.

GOLDBARG, M. C., GOUVÊA, E. F., SOUZA, A. C. Algoritmo transgenético para a solução do problema do passeio do pistoneio. **Anais do XXXIII Simpósio Brasileiro de Pesquisa Operacional**, 2001.

GOLDBARG, M. C., GOLDBARG, E. F. Transgenética computacional: uma aplicação ao problema quadrático de alocação. **Pesquisa Operacional**, v. 22(3), p. 359-386, 2002.

GOLDBARG, M. C., BAGI, L. B., GOLDBARG, E. F. Algoritmo transgenético para a solução do problema do caixeiro comprador capacitado. *To appear in Pesquisa Operacional*, 2006.

GOUVÊA, E. F. Transgenética computacional: um estudo algorítmico. Tese (Doutorado em Engenharia de Produção). Universidade Federal do Rio de Janeiro, 2001.

GUEGUEN, C. Méthodes de résolution exacte pour les problèmes de tournées de véhicules. Tese (Doutorado). Laboratoire Productique Logistique, Ecole Centrale, 1999.

HADJICONSTANTINO, E., CHRISTOFIDES, N., MINGOZZI, A. A new exact algorithm for the vehicle routing problem based on q-path and k-shortest paths relaxations. **Annals of Operations Research**, v. 61, p. 21-43, 1995.

HAYARI, N., MANIER, M.-A., BLOCH, C., EL MOUDNI, A. Un algorithme évolutionniste pour le problème de tournées sélectives avec contraintes de fenêtres de temps. **Proceedings of MOSIM'03**, 2003.

HOLLAND, J. H. **Adaptation in natural and artificial systems**: an introductory analysis with applications to biology, control and artificial intelligence. Ann Arbor: MIT Press, 1975.

KIRKPATRICK, S., GELLOT, J., VECCHI, M. Optimization by simulated annealing. **Science**, v. 220, p. 671-680, 1983.

LIMA NETO, W. S.. O problema do passeio do pistoneio periódico. Monografia (Bacharelado em Engenharia de Computação). Universidade Federal do Rio Grande do Norte, 2005.

MEDEIROS NETO, F. D. Abordagem evolucionária para problemas da cogeração e do roteamento da unidade móvel do pistoneio. Dissertação (Mestrado em Sistemas e Computação). Universidade Federal do Rio Grande do Norte, 2004.

MLADENOVIC, N., HANSEN, P. Variable neighborhood search. **Computers and Operations Research**, v. 24, p. 1097-1100, 1997.

MOROWITZ, H. J. **Beginning of cellular life**. New Haven: Yale University Press, 1992.

MOSCATO, P. On evolution, search, optimization, genetic algorithms and martial arts. **Caltech Concurrent Computation Program**, p. 158-179, 1989.

NEVES, J. A. Uma aplicação de algoritmo genético na otimização do emprego da unidade móvel de pistoneio. Dissertação (Mestrado em Sistemas e Computação). Universidade Federal do Rio Grande do Norte, 2000.

OSIADACZ, A. J., GÓRECKI, M. Optimization of pipe sizes for distribution gas network design, 1995. Disponível em: <http://www.psig.org/papers/1987/9511.pdf>.

OSMAN, I. H. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. **Annals of Operations Research**, v. 41, p. 421-451, 1993.

REINELT, G. A traveling salesman problem library. **ORSA Journal on Computing**, v. 3, p. 376-384, 1991.

RENAUD, J., BOCTOR, F. F., LAPORTE, G. An improved petal heuristic for the vehicle routing problem. **Journal of the Operational Research Society**, v. 47, p. 329-336, 1996.

ROTHFARB, B., FRANK, H., ROSENBAUM, D., STEIGLITZ, K., KLEITMAN, D. Optimal design of offshore natural-gas pipeline systems. **Operations Research**, v. 6(18), p. 922-1020, 1970.

SANTOS, A. C. SGUMP: Um sistema computacional georreferenciado de apoio à decisão para otimização do gerenciamento de unidades móveis de pistoneio. Dissertação (Mestrado em Sistemas e Computação). Universidade Federal do Rio Grande do Norte, 2002.

SEMET F., KHEMAKHEM M., CHABCHOUB H. A hybrid heuristic for the selective vehicle routing problem. **6th International Conference On Metaheuristics**, 2005.

SURRY, P. D., RADCLIFFE, N. J., BOYD, I. D. A multi-objective approach to constrained optimization of gas supply networks: the COMOGA method. In: **Evolutionary Computing**. Springer-Verlag, 1995.

TAILLARD, E. D. Parallel iteration search methods for vehicle routing. **Networks**, v. 23, p. 661-673, 1993.

TAN, C. C. R., BEASLEY, J. E. A heuristic algorithm for the Period Vehicle Routing Problem. **Omega**, v. 12, p. 497-504, 1984.

TANG, H. Routing problems with selection decisions: algorithms and implementations. Dissertação (Mestrado). The Pennsylvania State University, 2001.

THOMAS, J. E. **Fundamentos de engenharia de petróleo**. Rio de Janeiro: Petrobras, 2001.

WATERS, C. M., BASSLER, B. L. Quorum sensing: cell-to-cell communication in bacteria. **Annual Review of Cell and Developmental Biology**, v. 21, p. 319-346, 2005.

WILF, H. S. **Algorithms and complexity**. Englewood Cliffs: Prentice-Hall, 1986.

