



anp
Agência
Nacional do
Petróleo



Geologia, Geofísica &
Informática no Setor de
Petróleo & Gás na UFRN
ANP-MME-MCT

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE CIÊNCIAS EXATAS E DA TERRA
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO**

**Uma abordagem através de Algoritmos
Transgenéticos para o Problema da Configuração
do Traçado de uma Rede de Distribuição de Gás
Natural.**

Cristine Cunha Schmidt

Natal
Rio Grande do Norte - Brasil
18 de Abril de 2007.

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE CIÊNCIAS EXATAS E DA TERRA
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO**

**Uma abordagem através de Algoritmos
Transgenéticos para o Problema da Configuração
do Traçado de uma Rede de Distribuição de Gás
Natural.**

Cristine Cunha Schmidt

Dissertação apresentada ao Departamento de Informática e Matemática Aplicada da Universidade Federal do Rio Grande do Norte, como parte dos requisitos para a obtenção do título de Mestre em Sistemas e Computação.

Natal
Rio Grande do Norte - Brasil
18 de Abril de 2007.

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE CIÊNCIAS EXATAS E DA TERRA
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO**

**Uma abordagem através de Algoritmos
Transgenéticos para o Problema da Configuração
do Traçado de uma Rede de Distribuição de Gás
Natural.**

Cristine Cunha Schmidt

Orientadora: Prof^ª Dr^ª Elizabeth Ferreira
Gouvêa Goldbarg
Co-Orientador: Prof^º Dr^º Marco César
Goldbarg

Dissertação apresentada ao Departamento
de Informática e Matemática Aplicada da
Universidade Federal do Rio Grande do
Norte, como parte dos requisitos para a
obtenção do título de Mestre em Sistemas
e Computação.

Natal
Rio Grande do Norte - Brasil
18 de Abril de 2007.

AGRADECIMENTOS

Primeiramente quero agradecer à minha família, que nunca mediu esforços para me proporcionar de tudo, em especial a oportunidade de estudar. Este trabalho não seria possível sem a força e o encorajamento vindo de vocês.

À minha orientadora Elizabeth Goldbarg, pelo empenho em trabalhar juntamente comigo, sempre dando idéias para colaborar no andamento do trabalho, pela paciência e compreensão nos momentos difíceis.

A todos os colegas de mestrado, que sempre estiveram presentes nos momentos de descontração que foram muito importantes nos tempos em que passamos distantes da família. Em especial a uma amiga que se tornou uma irmã, Ligia Bariani Bagi.

Aos amigos Tatiana Thiele e Fagner Silveira, que me apoiaram muito na fase de conclusão deste trabalho. Igualmente às amigas Fabiany Lamboia, Alyne de Faria e Janaina Gonçalves.

À ANP - Agência Nacional do Petróleo, pelo apoio financeiro.

Uma abordagem através de Algoritmos Transgenéticos para o Problema da Configuração do Traçado de uma Rede de Distribuição de Gás Natural

Autora: Cristine Cunha Schmidt
Orientadora: Prof^ª Dr^ª Elizabeth Ferreira Gouvêa Goldberg
Co-Orientador: Prof^º Dr^º Marco César Goldberg

RESUMO

Palavras-chave: Transgenética, Algoritmos Evolucionários, Redes de Distribuição, Árvore de Steiner, Coleta de Prêmios.

Este trabalho apresenta um algoritmo transgenético híbrido para a solução de um Problema de Configuração de uma Rede de Distribuição de Gás Natural. O problema da configuração dessas redes requer a definição de um traçado por onde os dutos devem ser colocados para atender aos clientes. É estudada neste trabalho uma maneira de conectar os clientes em uma rede com arquitetura em forma de árvore. O objetivo é minimizar o custo de construção da rede, mesmo que para isso alguns clientes que não proporcionam lucros deixem de ser atendidos. Esse problema pode ser formulado computacionalmente através do Problema de Steiner com Prêmios. Este é um problema de otimização combinatória da classe dos NP-Árduos. Este trabalho apresenta um algoritmo heurístico para a solução do problema. A abordagem utilizada é chamada de Algoritmos Transgenéticos, que se enquadram na categoria dos algoritmos evolucionários. Para a geração de soluções iniciais é utilizado um algoritmo primal-dual, e *path-relinking* é usado como intensificador.

An Approach by Transgenetic Algorithms for the Design Configuration of a Natural Gas Distribution Network Problem

Author: Cristine Cunha Schmidt
Advisor: Elizabeth Ferreira Gouvêa Goldberg
Co-Advisor: Marco César Goldberg

ABSTRACT

Key words: Transgenetic, Evolucionary Algorithms, Distribution Networks, Steiner Tree, Prize Collecting.

This work presents a hybrid transgenetic algorithm for the solution of Design Network of a Natural Gas Distribution Problem. The problem of network configuration requires the definition of a layout where the ducts can be laid to attend the customers. This work studied a way to connect the customers in a network with tree architecture. The objective is to minimize the cost of the network construction, even if some not profitable customers leave of being assisted. This problem can be formulated by the Prize Collecting Steiner Tree Problem. It's a NP-hard combinatorial optimization problem. This work presents a heuristic algorithm for the solution of the problem. The used approach is called Transgenetic Algorithm, which is in the category of the evolutionary algorithms. For the generation of initial solutions we use a primal-dual algorithm, and path-relinking as intensifier.

Conteúdo

Lista de Siglas	x
Lista de Tabelas	xi
Lista de Figuras	xii
1 Introdução	1
1.1 Organização do trabalho	4
2 Especificação do problema	6
2.1 O gás natural - uma visão geral	6
2.2 O problema da configuração do traçado	12
2.3 Modelagem computacional	13
2.4 Métodos de solução	16
3 Modelagem do problema da configuração do traçado através do PSP	21
3.1 Definição	21
3.2 Modelagem matemática	23
3.3 Aplicações	25
3.4 Problemas relacionados	25
3.5 Revisão da literatura	27
3.5.1 O surgimento do problema	27
3.5.2 Algoritmos exatos, limites inferiores e estudos poliédricos	28
3.5.3 Algoritmos aproximativos	30
3.5.4 Algoritmos heurísticos	33
3.6 Banco de instâncias	37
4 Transgenética computacional	39
4.1 A evolução das espécies	39
4.2 Teoria da endossimbiose	42
4.3 Transferência horizontal	45
4.4 Algoritmos baseados em endossimbiose	46
4.5 Os algoritmos transgenéticos	47
4.5.1 Regras transgenéticas (RET)	49
4.5.2 O banco de informações genéticas (BIG)	50
4.5.3 Vetores transgenéticos	51
4.6 Algoritmos transgenéticos já desenvolvidos	54

5	Algoritmo transgenético para o PSP	56
5.1	Banco de informações genéticas	60
5.2	População inicial	60
5.3	Vetores transgenéticos	62
5.3.1	O vetor plasmídio	62
5.3.2	O vetor transposon	65
5.4	<i>Path – relinking</i>	67
5.5	Perturbação	69
5.6	Critérios de parada	69
6	Resultados computacionais	70
6.1	Definição de parâmetros	70
6.2	Resultados obtidos	73
7	Considerações finais	83

Lista de Siglas

AG	Algoritmo Genético
AGM	Árvore Geradora Mínima
AGMP	Árvore Geradora Mínima Probabilística
AM	Algoritmo Memético
AMH	Algoritmo Memético Híbrido
ANP	Agência Nacional do Petróleo
AT	Algoritmo Transgenético
ATH	Algoritmo Transgenético Híbrido
BIG	Banco de Informações Genéticas
BT	Busca Tabu
<i>DNA</i>	<i>Deoxyribonucleic Acid</i>
<i>FPCSTP</i>	<i>Fractional Prize Collecting Steiner Tree Problem</i>
GLP	Gás Liquefeito de Petróleo
GN	Gás Natural
GNC	Gás Natural Comprimido
GNL	Gás Natural Liquefeito
GNV	Gás Natural Veicular
<i>GRASP</i>	<i>Greedy Randomized Adaptive Search Procedures</i>
<i>GSEC</i>	<i>Generalized Subtour Elimination Constrains</i>
<i>GPCSTP</i>	<i>Generalized Prize Collecting Steiner Tree Problem</i>
GW	Goemans e Williamson
<i>NW</i>	<i>Net Worth</i>
<i>NWSTP</i>	<i>Node Weighted Steiner Tree Problem</i>
PCG	Problema da Coloração em Grafos
<i>PCSTP</i>	<i>Prize Collecting Steiner Tree Problem</i>
PGM	Partícula Genética Móvel
<i>PLPCSTP</i>	<i>Piecewise Linear Prize Collecting Steiner Tree Problem</i>
PQA	Problema Quadrático de Alocação
PSP	Problema de Steiner com Prêmios
<i>RAM</i>	<i>Random Access Memory</i>
<i>RPCSTP</i>	<i>Rooted Prize Collecting Steiner Tree Problem</i>
<i>SA</i>	<i>Simulated Annealing</i>
SCI	Sistemas de Complexidade Irredutível

<i>SFL</i>	<i>Shuffled Frog Leaping</i>
<i>SPWST</i>	<i>Single Point Weighted Steiner Tree Problem</i>
TC	Transgenética Computacional
UPGN	Unidade de Processamento de Gás Natural
VNS	<i>Variable Neighbourhood Search</i>

Lista de Tabelas

4.1	Alguns procedimentos típicos dos vetores transgenéticos.	53
4.2	Algoritmos transgenéticos já desenvolvidos.	55
6.1	Resultados obtidos para a primeira configuração de parâmetros	70
6.2	Resultados obtidos para a segunda configuração de parâmetros	71
6.3	Resultados obtidos para a terceira configuração de parâmetros	71
6.4	Resultados obtidos para a quarta configuração de parâmetros	71
6.5	Resultados obtidos para a quinta configuração de parâmetros	72
6.6	Resultados obtidos para a sexta configuração de parâmetros	72
6.7	Média, mediana e desvio padrão para as séries K e P.	74
6.8	Média, mediana e desvio padrão para a série C.	75
6.9	Média, mediana e desvio padrão para a série D.	76
6.10	Resultados obtidos para as séries K e P.	78
6.11	Resultados obtidos para a série C.	79
6.12	Resultados obtidos para a série D.	80
6.13	Resultados comparativos.	81
6.14	Demanda estimada dos cliente da rede do bairro Candelária.	81
6.15	Custo dos dutos de três fabricantes.	82

Lista de Figuras

2.1	Infra-estrutura de transporte de gás natural no Brasil.	11
2.2	Rede de distribuição na cidade de Natal.	12
2.3	Tipos de redes de distribuição de gás.	13
2.4	Projeto de rede para o bairro Candelária, em Natal.	16
3.1	Uma grafo e sua solução ótima.	23
4.1	Formação das mitocôndrias.	44
4.2	Processo de transferência horizontal.	45
4.3	Atuação das regras transgenéticas.	50
4.4	Componentes do banco de informações.	51
4.5	Procedimentos que definem os vetores transgenéticos.	52
4.6	Origem da informação dos vetores.	53
5.1	Exemplo de representação através do cromossomo.	57
5.2	Dois plasmídios competindo.	63
5.3	Cromossomo sendo manipulado pelo transposon.	66
6.1	Representação computacional da Fase 2 do projeto de redes da cidade de Natal.	82

Capítulo 1

Introdução

O gás natural - GN é um combustível fóssil com constituição semelhante a do petróleo, e pode ser encontrado no subsolo ou no fundo do mar, em depósitos naturais de rochas sedimentares, associado ou não ao petróleo. Sua origem é da decomposição de plantas e animais, resultado de um processo de milhões anos [r25].

Nos últimos 40 anos, pelas grandes vantagens que oferece quando comparado com outras fontes primárias de energia, o gás natural tem se transformado em uma alternativa energética em diversos países do mundo. Atualmente, é utilizado nos setores da indústria, em residências, no comércio, na geração de energia e nos transportes [r25]. Pode ser utilizado como fonte primária de energia, transformada em energia térmica (geração de calor e/ou frio) e mecânica (geração de torque de motor e/ou eletricidade).

Ele se caracteriza por uma queima eficiente, limpa e competitiva. É mais seguro, de fácil manuseio e com reduzida emissão de gases poluentes. As suas propriedades fazem com que seja um combustível atrativo para utilizações industriais, pois dispensa armazenamento, não necessita de bombagem e preparação para utilização, eliminando, assim, todos os custos associados a estas operações. Além destas vantagens, permite um elevado rendimento térmico, não necessita de tratamentos dos gases resultantes do processo de combustão, prolonga o tempo de vida dos equipamentos e possibilita a aceleração dos ciclos de fabricação.

O gás natural promove no Brasil uma revolução de consumo devido ao aumento da demanda pelo crescente número de indústrias e empresas usuárias. Além de suas vantagens para o meio ambiente, a sua grande disponibilidade como energia facilitará um consumo ainda maior durante os próximos anos.

Para que esta fonte de energia chegue até seus consumidores finais são necessários diversos processos, entre eles o processo de transporte. A melhor opção em termos econômicos para o transporte de gás natural no estado gasoso são os gasodutos [BNO01]. Eles são os responsáveis por levar o gás desde seu ponto de produção até as empresas distribuidoras (públicas ou privadas), que distribuem o gás através de linhas ou redes urbanas de distribuição, as quais são as conexões até o consumidor (residência, comércio ou indústria).

O gás natural somente passou a ser produzido no Brasil na década de 60 com as descobertas de óleo e gás na Bahia, mas a distribuição ficou restrita a áreas próximas a sua produção. Na década de 80, o gás natural foi finalmente disponibilizado para consumo residencial e industrial na região Sudeste, graças à construção da rede de gasodutos que liga a Bacia de Campos ao Rio de Janeiro e São Paulo. Da mesma forma no Nordeste do país, o gasoduto ligando Guamaré a Cabo foi concluído, possibilitando assim a distribuição de gás natural nos estados de Pernambuco, Paraíba e Rio Grande do Norte.

Apesar do esforço em se construir vias de escoamento e transporte da produção de gás, as reservas e a produção brasileira sempre foram pequenas comparadas ao consumo total de energia primária do país. Na década de 90, com a assinatura do acordo Brasil-Bolívia, iniciou-se a construção do gasoduto Gasbol que conecta jazidas bolivianas à região Sudeste e Sul do Brasil. No Nordeste a rede de gasodutos foi ampliada e atualmente atravessa os estados da Bahia, Sergipe, Alagoas, Pernambuco, Paraíba, Rio Grande do Norte e Ceará.

Segundo a ANP - Agência Nacional do Petróleo [r3], hoje a movimentação do volume de gás natural no Brasil é feita por uma malha que congrega 79 gasodutos, num total de 7.624 km de extensão. Mas se pensarmos em termos de todo território nacional (8.511.965 km²), este número ainda é bastante inexpressivo. Ainda segundo a ANP, as reservas provadas de gás no País atingem 306,4 bilhões de m³.

A participação do gás natural na matriz energética do Brasil era de 0,9% em 1981 e saltou para 8,9% em 2004. A meta do governo federal é chegar em 2010 com o gás natural respondendo por 15% da matriz energética nacional [r78]. Para que isso aconteça vários fatores devem ser levados em conta, como investimentos na área de produção; oferta crescente e regular do produto; regulamentação do preço; divulgação do seu baixo poder

de impacto ambiental e investimentos na ampliação da rede de distribuição e transporte do produto [Cas04].

O projeto de uma rede urbana de distribuição de gás natural envolve, entre outros problemas, o da definição de um *layout* para a rede, ou seja, um traçado físico por onde os dutos serão colocados, levando em conta questões econômicas e ambientais. A tubulação é o componente de maior custo do sistema [BNO01], por isso soluções são estudadas com o objetivo de diminuir o consumo deste material.

Este trabalho tem o objetivo de aplicar métodos quantitativos da Pesquisa Operacional e da Otimização Combinatória no desenvolvimento de um algoritmo. Este algoritmo visa constituir uma ferramenta a disposição do projetista para auxiliá-lo na tomada de decisões em projetos de redes urbanas de distribuição de gás natural no que se refere ao traçado da rede, tendo em vista uma análise mais precisa em relação ao custo final do projeto.

O desenvolvimento deste trabalho se justifica pelas vantagens encontradas no emprego do gás natural nos diversos setores, pelo seu baixo impacto ambiental, pelo seu custo competitivo quando comparado com outras fontes de energia, e pelo grande volume de gás existente no Brasil. Ao contrário dos países de primeiro mundo, como os da Europa e os Estados Unidos, que são dotados de extensas e complexas redes urbanas de distribuição de gás, no Brasil a necessidade é ainda a criação de infra-estrutura básica [GGC94]. Portanto, devido a todos os atrativos oferecidos pela utilização do gás natural, e por ser bastante primária a infra-estrutura da rede de transporte no Brasil, motivou-se a realização deste projeto.

Está sendo ampliada na cidade de Natal uma rede urbana de distribuição de gás natural, e há interesse por parte da companhia distribuidora do estado do Rio Grande do Norte em um *software* que possa dar apoio ao projetista no sentido de definir um traçado por onde os dutos deverão ser colocados.

Já existem *softwares* que realizam este trabalho, porém o algoritmo aqui proposto tem uma característica especial, que proporciona vantagens à companhia fornecedora do serviço. Neste trabalho o problema da configuração de redes de distribuição de gás é representado através do problema de Steiner com prêmios (PSP). Este modelo permite conectar um subconjunto de clientes a uma rede de distribuição de gás levando em conta fatores econômicos, onde apenas serão conectados à rede clientes financeiramente atra-

tivos. Por se tratar de um problema altamente combinatório, escolheu-se uma abordagem metaheurística para solucioná-lo. A abordagem utilizada para a solução do problema é chamada de algoritmos transgenéticos (ATs).

Os ATs pertencem à classe dos algoritmos evolucionários e apóiam sua metáfora na endossimbiose e propriedades do fluxo intracelular [GG02]. A endossimbiose 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 [Mor92].

Os ATs, adaptando os conceitos da endossimbiose 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 - 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 [WB05].

Os ATs necessitam de uma população de cromossomos para interagirem com o vetores transgenéticos. Para gerar uma população inicial foi implementado um algoritmo primal-dual com perturbações. O algoritmo aqui proposto pode ser considerado um algoritmo transgenético híbrido, pois ele utiliza, além dos recursos da transgenética computacional, um procedimento de *path-relinking* [Glo96] como intensificador.

Dentre os trabalhos desenvolvidos no grupo de pesquisa ao qual este trabalho pertence, estão diversos trabalhos aplicados à indústria de petróleo e gás natural, como pode ser visto na subseção 4.6.

1.1 Organização do trabalho

O trabalho está organizado da seguinte maneira. O Capítulo 2 apresenta a especificação do problema. É apresentada uma visão geral sobre o gás natural e a indústria

brasileira. É especificado o problema da configuração do traçado de redes de distribuição, sua modelagem e possíveis métodos de solução.

O Capítulo 3 trata do PSP. O problema é definido, são apresentados sua modelagem matemática, suas aplicações e problemas relacionados, além de seu banco de instâncias e uma revisão da literatura do problema, mostrando seu surgimento e os algoritmos propostos para solucioná-lo.

O Capítulo 4 introduz a computação evolucionária. É feita uma descrição de algumas teorias evolucionárias, é apresentada a metáfora dos algoritmos transgenéticos, e é feito um pequeno levantamento da literatura em algoritmos que se encaixam nesse contexto, assim como dos algoritmos transgenéticos desenvolvidos até o presente momento.

O Capítulo 5 descreve o algoritmo transgenético híbrido implementado. No Capítulo 6 são apresentados os resultados computacionais obtidos e é feita uma análise comparativa dos mesmos.

No Capítulo 7 são feitas as considerações finais, e por fim são apresentadas as referências bibliográficas.

Capítulo 2

Especificação do problema

Este capítulo visa definir o problema da configuração do traçado de redes de distribuição de GN. Serão mostrados os tipos de redes possíveis, a modelagem computacional e os métodos de solução mais comuns. Inicialmente serão especificadas as características do GN, em uma visão geral. Serão abordados tópicos como a composição e extração do gás, diferenças entre o GN e outras fontes de energia, vantagens de sua utilização, suas aplicações, e as formas de transporte.

2.1 O gás natural - uma visão geral

O GN é um combustível fóssil formado através do acúmulo e posterior transformação de matéria orgânica sob a superfície terrestre, há milhares de anos. Esta matéria orgânica toma a forma de hidrocarbonetos, como óleo, carvão, xisto e GN. Decompostos em estruturas moleculares mais simples e leves, como o metano, acumulam-se em rochas permeáveis, denominadas reservatórios.

A composição do GN pode variar dependendo do fato de estar associado ou não ao óleo, ou de ter sido ou não processado em unidades industriais [Kir04]. É basicamente constituído de metano (mais de 80%), etano (menos de 10%) e, em menores proporções, propano e hidrocarbonetos de maior peso molecular [r16].

Na natureza, ele é encontrado em acumulações de rochas porosas no subsolo (terrestre ou marinho), em geral acompanhado de petróleo. O GN pode ser classificado em duas categorias: associado e não associado. O gás associado é aquele que, no reservatório, encontra-se em companhia do petróleo, estando dissolvido no óleo ou sob forma de uma capa de gás isto é, uma parte superior da acumulação rochosa, onde a concentração de

gás é superior à concentração de outros fluidos como água e óleo. O gás não associado é aquele que, no reservatório, está livre do óleo ou este se encontra em concentrações muito baixas. Na acumulação rochosa porosa, a concentração de gás é predominante, permitindo a produção basicamente de gás. [r26]

Existem algumas características que destacam as vantagens do uso do GN, e o diferenciam de outros combustíveis:

- O GN não é tóxico;
- Sua densidade relativa é igual a 0,6. Isso significa que ele é mais leve que o ar e, em caso de vazamento, se dissipa na atmosfera naturalmente;
- O gás é incolor e inodoro em seu estado natural. Um cheiro característico (mistura de mercaptana) é inserido em sua composição para facilitar a identificação de eventuais vazamentos;
- O GN é inflamável apenas quando atinge temperaturas superiores a 622°C, o que dificulta qualquer tipo de queima acidental;
- O GN não é explosivo, apesar de ser inflamável. Se houver combustão, ele manterá a chama sem explosões até que seu suprimento se esgote;
- De todos os combustíveis utilizados atualmente, o GN é um dos menos poluentes, dispensando, por isso, equipamentos de filtragem dos gases combustíveis.

O aproveitamento do gás é feito através da perfuração de poços. A exploração é a etapa inicial do processo e consiste em duas fases: a pesquisa, onde é feito o reconhecimento e o estudo das estruturas propícias ao acúmulo de petróleo e/ou GN, e a perfuração do poço, para comprovar a existência desses produtos em nível comercial. Após a extração, ao chegar à superfície, o gás é processado para remoção de impurezas, como água e outros gases. A seguir, ele é transportado para as zonas de consumo e refino.

Na etapa de processamento, o gás segue para unidades industriais, conhecidas como UPGN - Unidades de Processamento de Gás Natural, onde ele será desidratado (isto é, será retirado o vapor d'água) e fracionado, gerando as seguintes correntes: metano e etano (que formam o gás processado ou residual); propano e butano (que formam o GLP - gás

liquefeito de petróleo ou gás de cozinha); e um produto na faixa da gasolina, denominado C5+ ou gasolina natural.

No estado líquido (como GNL - gás natural liquefeito), pode ser transportado por meio de navios, barcas e caminhões criogênicos, a -160°C , e seu volume é reduzido em cerca de 600 vezes, facilitando o armazenamento. Nesse caso, para ser utilizado, o gás deve ser revaporizado em equipamentos apropriados [r16] [Kir04].

No estado gasoso, o transporte do GN é feito por meio de gasodutos ou, em casos muito específicos, em cilindros de alta pressão (como GNC - gás natural comprimido).

Nos gasodutos, a força que faz com que o transporte de gás aconteça é a pressão, que gradualmente se dissipa e, sendo assim, uma série de estações de compressão são posicionadas ao longo do caminho do gasoduto. Cada estação tem um grande número de compressores que aumentam a pressão do gás, a fim de "empurrá-lo" até a próxima estação. Nos dutos de transporte de longa distância, as pressões usuais podem atingir de 100 a 150 kg/cm^2 logo após a estação de compressão, caindo, ao longo do duto, até cerca de 30 a 40 kg/cm^2 , quando haverá uma outra estação de compressão. Este ciclo pode se repetir várias vezes, permitindo atingir distâncias praticamente ilimitadas [r16].

Além dos gasodutos de transporte, existem os de transferência e de distribuição de gás no estado gasoso. Os gasodutos de transferência são de uso particular do proprietário ou explorador das facilidades, e conduzem a matéria-prima até o local de processamento ou utilização.

De forma semelhante, os gasodutos de distribuição (ou redes de distribuição) levam o gás canalizado recebido das transportadoras até os usuários finais. A distribuição é a etapa final do sistema de fornecimento. É o momento em que o gás chega ao consumidor para uso industrial, automotivo, comercial ou residencial. Nessa fase, o gás já deve atender a padrões de especificação e, praticamente, isento de contaminantes, para não causar problemas aos equipamentos nos quais será utilizado como combustível ou matéria-prima. Quando necessário, deverá também estar odorizado, para ser detectado facilmente em caso de vazamentos [Kir04]. As redes de distribuição transportam volumes menores de GN a menores pressões, com tubulações de diâmetros menores que as dos gasodutos. A pressão é reduzida para 5 a 6 kg/cm^2 nos ramais principais e, nas unidades de consumo, para 15 a 30 cm de coluna d'água [r16].

Por ser mais abundante, com reservas superiores às do petróleo, é mais barato. Em longo prazo, sua utilização reflete-se no barateamento dos custos de produção, com efeito direto nos preços para o consumidor final.

O GN pode substituir o gás de botijão (GLP) nas residências. Pode ser utilizado em sistemas de calefação e de aquecimento de água, em aparelhos de ar condicionado, geladeiras e fornos de microondas. Restaurantes, hotéis e bares utilizam este combustível comercialmente, freqüentemente nas cozinhas.

A substituição do GN como combustível (GNV) no transporte significa uma considerável redução na poluição das grandes cidades, além de baratear, em longo prazo, a locomoção de produtos e pessoas. Por ser mais limpo, com uma queima mais eficiente, reduz os gastos com manutenção e aumenta a vida útil dos veículos.

Na indústria, o GN propicia aumento na qualidade final dos produtos. Não necessita de aquecimento antes da combustão, possibilita o controle preciso da temperatura e assegura queima completa. Por ser mais limpo, os custos com equipamentos antipoluentes são eliminados. Por ser entregue via dutos, elimina os gastos com frete, assim como os custos de estocagem associados a outros combustíveis.

A geração de energia elétrica a partir de GN é feita pela queima do gás combustível em turbinas a gás. Entre as vantagens adicionais da geração termelétrica a GN estão o prazo relativamente curto de maturação do empreendimento e a flexibilidade para o atendimento de cargas de ponta [r84]. Estão sendo construídas 56 usinas termelétricas no Brasil, sendo muitas delas abastecidas pelo GN.

Com o objetivo de atender a necessidade de mudança da matriz energética brasileira, contribuindo para o aumento de consumo do GN, em 1992 firmou-se um acordo para a construção do gasoduto Brasil-Bolívia (também conhecido como Gasbol). Entre as possíveis alternativas de importação analisadas pela Petrobrás, o GN boliviano mostrou-se a mais viável: abundância nas reservas do país, alto poder de queima do gás e proximidade com o Brasil.

O gasoduto começou a ser construído em 1997, iniciando sua operação em 1999. Ele transporta o GN proveniente de Rio Grande, na Bolívia, até Canoas, na Grande Porto Alegre. São 3.150 Km de extensão, sendo 2.593 Km em solo brasileiro. A realização desta obra aumentou em quase 60% a malha até então existente, gerando crescimento

econômico em grande parte do país. Estima-se que o gasoduto estará plenamente operativo em 2010, e o objetivo é de que o GN chegue a 15% de todo o consumo energético brasileiro.

A Figura 2.1 indica as principais conexões entre os países da América do Sul. São mostrados gasodutos em operação, em construção, em estudo e reservas de gás. Para viabilizar um incremento da posição do GN no Brasil, a Petrobrás desenvolveu as seguintes diretrizes: interligar toda a região Nordeste, desde a Bahia até o Ceará, e ligar as regiões Sudeste e Sul. A interligação dos estados do Nordeste foi sendo construída passo a passo, sendo que os trechos Guamaré (RN) - Pecém (CE) e Pilar (AL) - Cabo (PE) entraram em operação recentemente. Com isto, as reservas da região podem ser mobilizadas para atender a demanda. No Sudeste e Sul, foi necessária a ligação dutoviária ao longo da região costeira, onde há a maior concentração humana e, portanto, maior consumo potencial. A interligação do Nordeste com a Bacia de Campos, integrando toda a região litorânea, será facilitada pela produção de gás na área norte da Bacia de Campos.

A Potigás - Companhia Potiguar de Gás [r11] é responsável exclusiva pela distribuição de gás canalizado no Rio Grande do Norte. O GN distribuído e comercializado é oriundo dos campos de produção de gás do Rio Grande do Norte, processado na UPGN da Petrobrás em Guamaré e transportado através dos Gasodutos do Nordeste - NORDESTÃO e Guamaré - Pecém - GASFOR. A Potigás também está ligada aos campos de GN de Alagoas, Sergipe e Bahia através do Gasoduto Pilar - Cabo - GASALPE. A distribuição do GN pela Potigás é realizada através de uma rede de gasodutos de 180 km. A sua rede de distribuição atende consumidores industriais e comerciais em oito municípios do Estado.

A Figura 2.2 mostra a rede de dutos instalada pela Potigás na região da grande Natal. Como o mercado do Rio Grande do Norte apresenta um grande potencial de crescimento, a Potigás investiu em alguns projetos, nos quais destacam-se a ampliação da rede de distribuição em Natal e o fornecimento de GN para comércios e domicílios. Foram concluídas em novembro de 2006, as obras de duplicação do gasoduto tronco de atendimento a Natal. O projeto prevê a ampliação da capacidade de suprimento a malha de distribuição de GN da Grande Natal de 300 mil m³/dia para 700 mil m³/dia.

Assim como no caso do Rio Grande do Norte, a implantação de gasodutos no Brasil impulsiona a instalação de sistemas de transporte para a distribuição urbana do gás. A



Figura 2.1: Infra-estrutura de transporte de gás natural no Brasil.

Petrobrás já estuda ampliar o Gasoduto Brasil-Bolívia para atender as necessidades do mercado brasileiro de gás em 2009 ou 2010.

Entre redes de transmissão e distribuição, o Brasil tem cerca de 17 mil km de tubos assentados, o que limita a chegada do combustível a novos mercados. Somente em redes de transporte, o Brasil tem 8 mil km, enquanto isso, a França tem 34 mil km, o Canadá 100 mil km e os Estados Unidos 450 mil km. Há boas perspectivas para o crescimento do mercado do GN no Brasil desde que alguns entraves sejam removidos. Mais que isso,



Fonte: Potigas

Figura 2.2: Rede de distribuição na cidade de Natal.

é necessário que o crescimento do mercado do GN seja inserido realmente na política estratégica para o setor de energia do país para garantir o crescimento deste combustível na matriz energética brasileira.

2.2 O problema da configuração do traçado

O projeto de uma rede distribuição de gás envolve a definição de um *layout* para a rede, ou seja, um traçado físico por onde a rede deverá passar, levando-se em conta questões econômicas e ambientais, entre outras. Este traçado é um pré-requisito para a escolha dos tipos e dos diâmetros dos dutos a serem utilizados, de forma que as demandas dos usuários da rede sejam atendidas, ou seja, uma certa pressão mínima deve ser disponibilizada em determinados pontos da rede [Cas04].

Podem existir diversos tipos de solução para esse problema, dependendo do tipo de rede que é requerida. O fator econômico tem um papel significativo na escolha da solução

final. A seleção final adotada está muitas vezes a cargo do projetista e não exclusivamente dependente do traçado resultante.

O problema da configuração do traçado pode ser modelado através de grafos, e o objetivo é definir uma estrutura de custo mínimo em relação ao comprimento dos trechos de tubulação, considerando as restrições referentes à implantação [GGC94]. Dependendo do tipo de rede a ser construída, são empregados modelos para a formalização e resolução do problema.

Existem basicamente três tipos de redes de distribuição de gás, como pode ser visto na Figura 2.3. A rede mais esparsa é denominada rede ramificada e basicamente objetiva que o cliente seja conectado a rede (Figura 2.3a). Redes mais complexas permitem a formação de circuitos alternativos para o atendimento dos clientes. A multiplicidade de circuitos de fornecimento tem o objetivo de garantir a confiabilidade e a continuidade dos serviços de distribuição ainda que sob condições de manutenção ou de interrupções imprevistas em determinadas rotas (Figura 2.3b e Figura 2.3c).

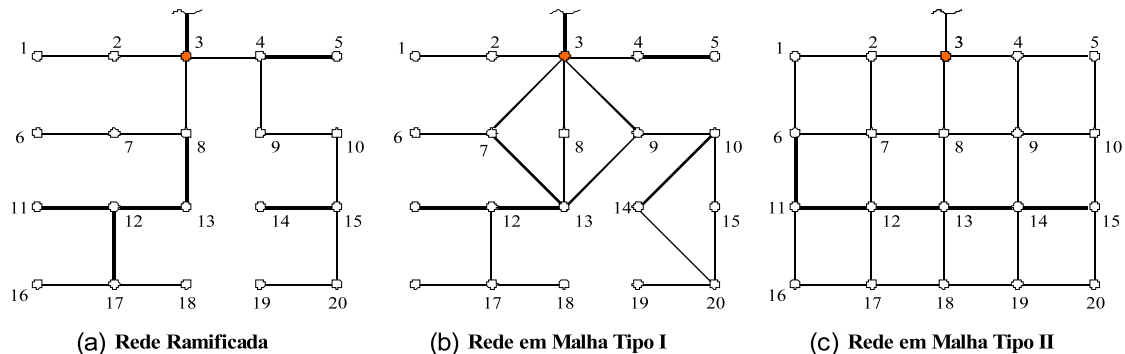


Figura 2.3: Tipos de redes de distribuição de gás.

Neste trabalho serão consideradas apenas redes ramificadas, onde existe um único caminho ligando um nó v a qualquer outro nó w da rede. Esse tipo de rede apresenta uma economia em relação à tubulação utilizada em sua construção, devido à menor quantidade de material empregado.

2.3 Modelagem computacional

Segundo Cormen et al. [CLRS01], um grafo $G = (V, E)$ é um conjunto de pontos V , chamados vértices ou nós, conectados por um conjunto de linhas E , chamadas de

arestas ou arcos. Dependendo da aplicação, as arestas podem ou não ter direção, pode ser permitido ou não arestas ligarem um vértice a ele próprio e vértices e/ou arestas podem ter um peso numérico associado.

Um circuito num grafo é um caminho onde os nós inicial e final são os mesmos. Um grafo é dito conexo se, para ir de um nó v do grafo a qualquer outro nó w , existe pelo menos um caminho de v a w .

Uma árvore T é um grafo conexo que não contém circuitos. Uma árvore geradora para um grafo G é um subgrafo que é uma árvore e que contém todos os nós de G . Assim, em uma árvore, quaisquer dois nós são unidos por um único caminho. Além disso, o número de arestas é igual ao número de nós menos 1. O custo de um grafo é a soma dos custos das arestas do grafo. Uma árvore geradora mínima (AGM) é uma árvore geradora que possui custo mínimo. O problema de determinar uma árvore geradora de custo mínimo é conhecido como o problema da árvore geradora mínima.

O problema da configuração do traçado para redes do tipo ramificadas pode ser representado pelo problema da AGM. Devem ser considerados um conjunto de nós a serem atendidos por uma empresa distribuidora de GN, e um conjunto de arestas que representam as ligações entre os nós. O objetivo é ligar todos os nós do grafo (os clientes) através de um subconjunto de arestas (as ruas por onde os dutos serão colocados) de custo mínimo, considerando como custo a soma dos pesos das arestas (comprimentos dos dutos).

Outro modelo intuitivo para formalizar o problema da configuração do traçado de redes ramificadas é um problema clássico da otimização combinatória: o problema da árvore de Steiner [GL00][GGC94][Wer01]. Em termos de grafos, o problema consiste no seguinte: seja $G = (V, E)$ um grafo onde $V = \{v_1, \dots, v_n\}$ representa os possíveis pontos de passagem da rede de distribuição, pontos provavelmente associados a uma planta urbana, e $E = \{e_1, \dots, e_m\}$ representa as vias de ligação entre os pontos de V . O conjunto de pontos é composto por três tipos de nós: nós de demanda, nó fonte e nós de passagem. Às arestas do conjunto E estão associados custos $c_i, i = \{1, \dots, m\}$, que quantificam a distância entre os nós. O objetivo é encontrar uma configuração em forma de árvore, com custo mínimo, que conecte os nós de demanda ao nó fonte.

Existe um problema que é uma generalização do problema da árvore de Steiner, o chamado problema de Steiner com prêmios - PSP, problema da árvore de Steiner com

coleta de prêmios, ou *prize collecting Steiner tree problem*. Esse problema considera prêmios $\pi_i, i = \{1, \dots, n\}$, associados aos nós de demanda do grafo, além dos custos associados às arestas. O objetivo é otimizar o traçado de uma rede ramificada, em forma de árvore, de modo a minimizar os custos do traçado enquanto maximiza o lucro obtido em atender à demanda da rede.

Pode-se imaginar um cenário em que uma empresa deseja construir uma rede para fazer distribuição de gás. É conhecido o conjunto de clientes que têm interesse em receber o serviço. A empresa pode estimar qual será o ganho em atender cada cliente, e qual será o custo em colocar dutos nas ruas para atender aos mesmos. Os clientes são vistos como nós de demanda de um grafo, onde o valor da demanda (ou prêmio) é o lucro estimado pela empresa em atender ao cliente. Intersecções de ruas são vistas como nós de passagem do grafo, cuja demanda é zero. As ruas por onde os dutos vão ser colocados são vistos como as arestas do grafo, cujos pesos são os custos estimados dos dutos.

Um exemplo pode ser visto na Figura 2.4, que representa o projeto de uma rede de distribuição de gás para o bairro Candelária, na cidade de Natal. A empresa distribuidora de GN da cidade, a Potigás [r11], forneceu os dados da Fase 2 deste projeto. Cada ponto em preto no mapa representa um possível cliente, e tem uma demanda de gás estimada. Estão implícitos no mapa os pontos de passagem, que possuem demanda zero. A Fase 2 deste projeto foi solucionada através do algoritmo aqui proposto, e os resultados se encontram no Capítulo 6.

Esta abordagem proporciona uma vantagem para o fornecedor de serviços. Esta vantagem consiste na possibilidade de os clientes poderem ser selecionados pela companhia de acordo com o quanto ela irá ganhar em adicioná-los à rede. Portanto, só serão conectados à rede clientes financeiramente atrativos. O objetivo é traçar uma rede conectando um subconjunto de clientes, maximizando o lucro enquanto minimiza os custos de construção da rede. O Capítulo 3 apresenta uma definição detalhada deste problema, assim como sua modelagem matemática, aplicações e variações.

Pode-se dizer que o problema clássico da árvore de Steiner é um caso particular do PSP, onde todos os nós de demanda têm prêmio infinito, e os nós de passagem têm prêmio nulo. Karp [Kar76] definiu o problema da árvore de Steiner como sendo NP-Árduo, cuja dificuldade de solução computacional cresce exponencialmente em relação ao número

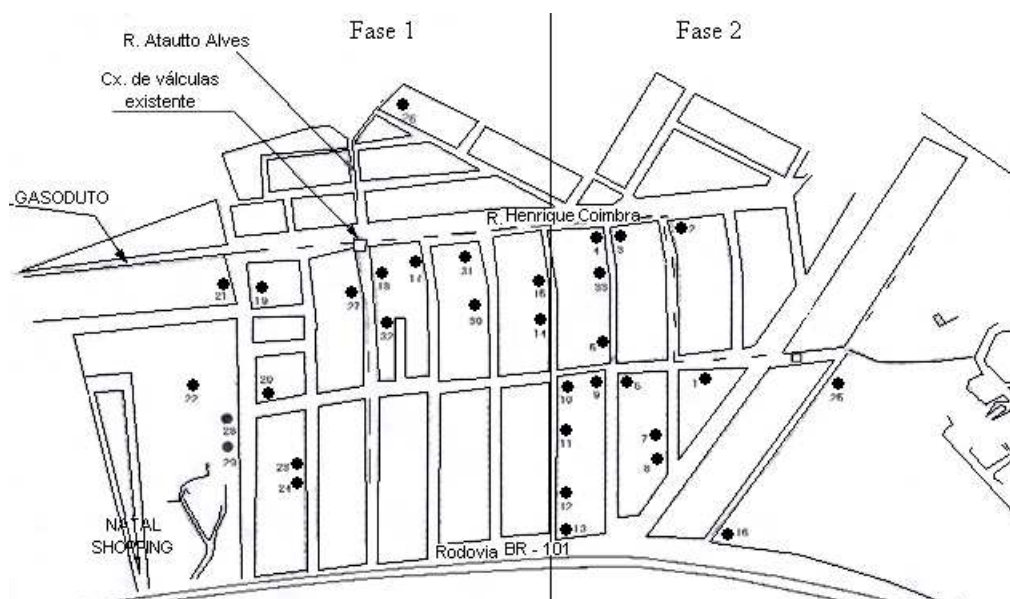


Figura 2.4: Projeto de rede para o bairro Candelária, em Natal.

de nós existente na rede. Sendo o PSP um caso geral deste problema, ele é também da classe dos NP-Árduos [CRR01]. Devido à complexidade de tais problemas, algoritmos que geram soluções aproximadas são bastante aplicados para resolvê-los.

2.4 Métodos de solução

Ao se pensar em um problema altamente combinatório, uma opção para solucioná-lo seria analisar todas as combinações possíveis para conhecer a melhor. Se o problema possui um universo de dados pequeno, realmente esta é uma maneira correta e viável de se buscar a melhor solução. Porém, os problemas reais normalmente possuem um número de combinações muito extenso, o que torna inviável a análise de todas as combinações, uma vez que o tempo computacional exigido se torna impraticável.

As heurísticas procuram encontrar soluções ótimas, ou próximas da otimalidade, em um tempo computacional razoável, sem, no entanto, conseguir definir se esta é a solução ótima, nem quão próxima ela está da solução ótima [CS03]. As heurísticas, entretanto, geralmente são desenvolvidas especificamente para um problema particular, não sendo eficientes na resolução de uma classe mais ampla de problemas.

Uma reunião de conceitos das áreas de Otimização e Inteligência Artificial deu início à criação de procedimentos heurísticos com uma certa estrutura teórica e com caráter mais

geral, sem prejudicar a principal característica destes, que é a flexibilidade. Estes procedimentos são conhecidos como métodos "inteligentemente flexíveis", ou "metaheurísticas". Dentre os procedimentos enquadrados como metaheurísticas que surgiram ao longo das últimas décadas, destacam-se os Algoritmos Genéticos (AGs) [Gol89], *Simulated Annealing* (SA) [KGV83], Busca Tabu (BT) [Glo86], *GRASP* [FR95], VNS [MH97], Colônia de Formigas [DC99], entre outros.

Assim como muitos problemas de otimização de interesse prático, o problema da configuração do traçado de redes de distribuição é de elevado nível de complexidade, ou seja, é de difícil tratamento computacional. Portanto, uma abordagem prática para resolver problemas deste tipo é empregar algoritmos heurísticos que podem encontrar soluções ótimas ou aproximadas em um tempo razoável de execução em um computador.

Como o problema da configuração de redes de distribuição de GN é semelhante ao problema da configuração de redes de distribuição de água, telecomunicações, energia elétrica, entre outros, serão relatados aqui algoritmos que foram propostos na literatura para solucionar o problema com diversas aplicações.

Kim e Gen [KG99] propuseram um AG para solucionar o problema da otimização de redes bi-critério, com aplicação em redes de telecomunicação. Eles utilizaram como base árvores geradoras, e os objetivos são a minimização do custo das conexões e do atraso médio das mensagens, considerando a confiabilidade da rede. Amasifen et al. [ARM05] utilizam AGs para solucionar o problema, com aplicação em redes de distribuição de energia elétrica. As configurações iniciais da população são feitas através de heurísticas baseadas na teoria de AGM, utilizando os algoritmos de Kruskal [Kru56] e Prim [Pri57]. Abuali et al. [ASW94] modelam o problema da configuração do traçado como um problema de AGM probabilístico (AGMP), e utilizam AGs para obter soluções para o problema. O problema da AGMP é uma generalização do problema clássico da AGM e é considerado um problema mais realístico do que este. Ele considera circunstâncias onde nem todos os nós estejam presentes, mas podem estar presentes com determinadas probabilidades.

Dandy et al. [DSM96] propuseram um AG para o problema da otimização de redes de distribuição de água. O algoritmo introduz um expoente na função de aptidão que aumenta em magnitude conforme o AG executa. É introduzido um novo operador de mutação. São apresentados resultados que comparam o desempenho de um AG tradicional com o AG

proposto para uma rede da cidade de Nova Iorque. Os resultados do estudo de caso indicam que o AG proposto é significativamente melhor do que AGs tradicionais. Dengiz et al. [DAS97] apresentam um AG com operadores especializados de codificação, inicialização e busca local para otimizar o projeto de redes de comunicação. Este problema NP-Árduo frequentemente tem muitas restrições, assim, uma inicialização aleatória e operadores genéticos padrão geralmente resultam em redes inviáveis. Outra complicação é que a função de aptidão envolve o cálculo da reabilitação de todos terminais da rede, o que é computacionalmente muito caro. Então, é necessário que a busca avalie a necessidade de explorar completamente a fronteira entre redes viáveis e inviáveis, juntamente com o cálculo da função de aptidão apenas nas redes mais promissoras. O resultado do algoritmo é comparado com resultados ótimos encontrados pelo método *branch-and-bound* e também com um AG com operadores de busca local em uma série de 79 instâncias.

Eusuff e Lansey [EL03] desenvolveram um algoritmo baseado na meta-heurística *Shuffled Frog Leaping - SFL*. O algoritmo visa determinar tamanhos discretos ótimos de uma tubulação para novas redes de distribuição de água e para expansões de redes já existentes. *SFL* é baseado em população, e sua metáfora de busca cooperativa é inspirada na memética natural. O algoritmo usa a evolução memética para introduzir informações de um indivíduo em outro, em uma busca local. A busca local é similar à utilizada em algoritmos de otimização por nuvem de partículas. Uma estratégia permite a troca de informação entre buscas locais em movimento para um ótimo global.

Alperovits e Shamir [AS77] propuseram um método de gradiente de programação linear para gerar sistemas de distribuição de água. O sistema é uma rede de encanamento que fornece demandas aos consumidores e pode conter bombas, válvulas, e reservatórios. As variáveis de decisão incluem parâmetros de projeto como diâmetros da tubulação, capacidades das bombas, e elevações do reservatório, e parâmetros operacionais como as bombas a serem operadas e os ajustes de válvula para cada uma das condições de carregamento. A função objetivo, a ser minimizada, reflete o capital do custo global mais o valor atual de custos operacionais.

Goulter e Bouchart [GB90] propõem uma metodologia para incorporar considerações de confiabilidade diretamente em redes de distribuição de água com ciclos a custo mínimo. A essência da metodologia é a medida de confiabilidade, e a possibilidade de se fazer

mudanças no sistema de distribuição se a confiabilidade encontrada for insatisfatória.

Uma variedade de esquemas de otimização local foram desenvolvidos para problemas de projetos de redes de distribuição de água. Eiger et al. [ESBT94] apresentam uma aproximação por otimização global para uma formulação baseada em ciclos e caminhos, usando programação linear semi-infinita para derivar limites inferiores. Em contraste, Sherali e Smith [SS97] empregam uma formulação baseada em arcos que é linear com exceção de certas restrições. Os limites inferiores são derivados de uma Técnica de Reformulação-Linearização satisfatória que constrói um relaxamento de programação linear para o problema, e isto é embutido em um algoritmo *branch-and-bound*. A convergência para uma solução ótima é induzida através de coordenação deste processo com um esquema de particionamento apropriado.

O trabalho de Mariano e Morales [MM99] soluciona o problema da configuração do traçado deste tipo de rede através de um algoritmo de colônia de formigas, com aplicação em distribuição de água para irrigação. A configuração do traçado é um dos objetivos do problema tratado por eles, que é o problema do projeto de redes de distribuição de água não linear com múltiplos objetivos. O algoritmo trabalha com famílias de formigas. Cada família procura por uma solução que depende da solução das outras famílias criando um mecanismo de negociação e procurando por soluções de acordo com todos os objetivos envolvidos. Para otimizar o traçado, uma família é responsável por encontrar uma disposição mínima da rede que conecta todos os nós que fornecem água a cada lote de demanda. Para isso, é utilizado um valor heurístico para avaliar o quão bom é fazer a conexão entre dois nós x e y .

Rodrigues e Silva [RS04] propõem um algoritmo onde a rede pode ser modelada como um problema de Steiner. Propõe-se a resolução em duas etapas, onde, na primeira etapa, executa-se o pré-processamento do problema, reduzindo a complexidade computacional do mesmo, e, numa segunda etapa, utiliza-se a Busca Tabu para a geração de soluções deste problema. O algoritmo de Souza [Sou04] define o problema de configuração do traçado como o problema de Steiner clássico, e utiliza um algoritmo transgenético para solucioná-lo. Este trabalho obteve bons resultados, e o algoritmo está imbutido no software POMDIGas [GCG04].

Klau et al. [KLM⁺03] modelam o problema da configuração do traçado através do

problema da árvore de Steiner fracionário. Desta maneira, o objetivo é maximizar a relação da soma dos lucros e a soma dos custos (fixo e variável), dado que isto corresponde ao fator de "retorno do investimento". Na formulação linear do problema, o lucro total, ou seja, a soma dos prêmios dos clientes conectados à rede menos a soma dos custos deve ser maximizada.

O presente trabalho apresenta uma abordagem metaheurística para resolver o problema da configuração do traçado, modelando-o através do PSP. Existem diversos algoritmos na literatura para solucionar este problema específico, utilizando diferentes métodos. Estes algoritmos estão reportados na revisão da literatura do problema, apresentada na sub-seção 3.5. A abordagem utilizada é denominada Trangenética Computacional (TC). O algoritmo desenvolvido nesse trabalho é um algoritmo transgenético híbrido, pois é utilizado junto com ele outra abordagem heurística. Para geração da população inicial é utilizada uma implementação de um algoritmo primal-dual, e para se obter intensificação no processo de busca é utilizado um algoritmo *path-relinking*. A abordagem utilizada mostrou-se competitiva, e os resultados são analisados no Capítulo 6.

Capítulo 3

Modelagem do problema da configuração do traçado através do PSP

3.1 Definição

Seja $G = (V, E)$ um grafo conexo não direcionado, onde $V = \{v_1, \dots, v_n\}$ é o conjunto de nós, e $E = \{e_1, \dots, e_m\}$ é o conjunto de arestas. Seja uma função custo não negativa $c : E \rightarrow \mathbb{R}_+$ associada às suas arestas e uma função prêmio não negativa $\pi : V \rightarrow \mathbb{R}_+$ associada aos seus nós. O objetivo do problema da árvore de Steiner com coleta de prêmios é encontrar uma sub-árvore T de G que seja ótima, ou seja, que minimize a soma dos custos de suas arestas somada com os prêmios dos nós que não estão na árvore (minimização de custos), ou que maximize a diferença entre a soma prêmios dos nós que estão na árvore e a soma do custo das arestas que estão na árvore (maximização do lucro).

Pode-se distinguir os vértices de demanda dos demais, através da definição do conjunto X , assumindo que $X \neq \emptyset$. Se o subconjunto de nós X é conhecido, tem-se o problema clássico da árvore de Steiner, que consiste em encontrar a AGM de G expandindo todos os nós de X . Este é um caso particular do PSP em que todos nós terminais tem prêmios infinitos e os outros tem prêmios nulos.

Existem quatro abordagens, com funções objetivo diferentes, para solucionar o PSP. Elas foram inicialmente estudadas por Goemans e Williamson em [GW96], e simplificam os conceitos de otimização:

- O problema de minimização de Goemans e Williamson (GW): Procura uma sub-árvore $T' = (V', E')$ de G que minimize o custo das arestas na árvore somado com

os prêmios dos vértices que não estão na árvore, isto é, minimizar

$$GW(T') = \sum_{e \in E'} c(e) + \sum_{v \notin V'} \pi(v) \quad (3.1)$$

- O problema de maximização *Net Worth* (*NW*): Procura uma sub-árvore $T' = (V', E')$ de G que maximize os prêmios dos nós que estão na árvore menos o custo das arestas que estão na árvore, ou seja, maximizar

$$NW(T') = \sum_{v \in V'} \pi(v) - \sum_{e \in E'} c(e) \quad (3.2)$$

- O problema das Cotas (*Quote*): dado uma cota de prêmios $Q > 0$, encontrar uma sub-árvore T' que minimize a soma dos custos das arestas que estão na árvore, sendo que a soma dos prêmios dos vértices que estão na árvore deve ser maior ou igual ao valor da cota de prêmio Q . Ou seja, minimizar

$$\sum_{e \in E'} c(e) \text{ sujeito a } \sum_{v \in V'} \pi(v) \geq Q \quad (3.3)$$

- O problema do Orçamento (*Budget*): dado um orçamento de arestas $B > 0$, encontrar uma sub-árvore T' que maximize a soma dos prêmios dos nós que estão na árvore, sendo que a soma dos custos das arestas que estão na árvore deve ser menor ou igual ao valor do orçamento de arestas B . Ou seja, maximizar

$$\sum_{v \in V'} \pi(v) \text{ sujeito a } \sum_{e \in E'} c(e) \leq B \quad (3.4)$$

A abordagem de minimização de custos GW (Equação 3.1) é a mais comumente utilizada. Então, para que seja possível realizar comparações com outros algoritmos, esta função objetivo foi adotada também neste trabalho. Dando enfoque a esta abordagem, a Figura 3.1 mostra um exemplo, onde tem-se um grafo e a solução ótima correspondente. Os nós são identificados por um número e seu valor de demanda (prêmio) está representado ao seu lado. Pode-se notar que os nós em cinza representam a demanda da rede, enquanto os nós em branco são os nós de passagem, cuja demanda é nula. As arestas são representadas pelas linhas (tracejadas ou contínuas), acompanhadas por um valor inteiro, que é o peso da aresta.

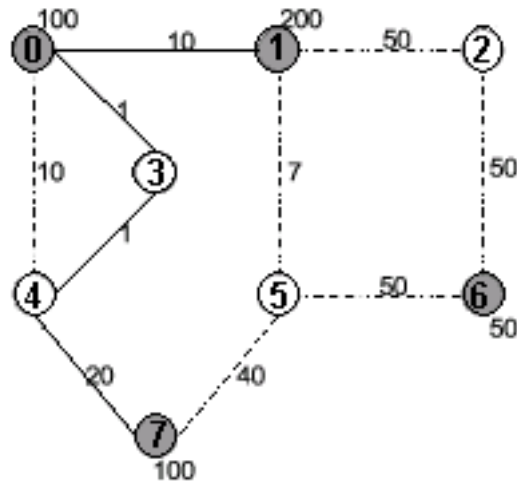


Figura 3.1: Um grafo e sua solução ótima.

As linhas tracejadas representam as arestas que não fazem parte da solução ótima, enquanto as linhas contínuas representam o traçado que liga os nós de demanda em uma rede em forma de árvore. Segundo a abordagem de minimização de custos GW, pode-se calcular o valor da solução da seguinte maneira. A soma dos pesos das arestas que fazem parte da solução equivale a $(20 + 1 + 1 + 10) = 32$, e a soma dos prêmios dos nós de demanda que não fazem parte da solução equivale a 50 (apenas um nó de demanda ficou fora da solução, o nó 6). Então o valor da solução é $(32 + 50) = 82$.

3.2 Modelagem matemática

Existem várias modelagens matemáticas para o PSP ([Lju04b][LWP⁺04][Min00][LR00]). Será mostrada aqui uma delas, que é baseada nas restrições de eliminação de sub-rotas generalizadas (*Generalized Subtour Elimination Constraints - GSEC*) [LR00], ou seja, é baseada na descrição da estrutura de árvore da solução.

Esta modelagem segue de uma formulação estendida do problema de Steiner em grafos, introduzida independentemente por vários autores. Considerando as variáveis x e y :

$$x_e = \begin{cases} 1, & \text{se } e \in E_T \\ 0, & \text{caso contrário} \end{cases} \quad \forall e \in E \quad (3.5)$$

$$y_v = \begin{cases} 1 & \text{se } v \in V_T \\ 0 & \text{caso contrário} \end{cases} \quad \forall v \in V \quad (3.6)$$

$$\min \sum_{e \in E} c_e x_e + \sum_{v \in V} \pi_v (1 - y_v) \quad (3.7)$$

sujeito a:

$$\sum_{e \in E} x_e = \sum_{v \in V} y_v - 1 \quad (3.8)$$

$$\sum_{e \in S} x_e \leq \sum_{v \in S\{k\}} y_v \quad \forall S \subseteq V, |S| \geq 2, \forall k \in S \quad (3.9)$$

$$0 \leq x_e \leq 1 \quad \forall e \in E \quad (3.10)$$

$$0 \leq y_v \leq 1 \quad \forall v \in V \quad (3.11)$$

$$y_v \in \{0, 1\} \quad \forall v \in V \quad (3.12)$$

Onde:

- x é um vetor de incidência que representa as arestas. Quando $x_e = 1$ significa que a aresta e pertence à solução, e quando $x_e = 0$ significa que a aresta e não pertence à solução.
- y é um vetor de incidência que representa os nós. Quando $y_v = 1$ significa que o nó v pertence à solução, e quando $y_v = 0$ significa que o nó v não pertence à solução.
- c é uma função de custo, onde c_e é o custo da aresta e .
- π é uma função de prêmio, onde π_v é o prêmio do nó v .
- S é um subconjunto de nós do conjunto $V(S \subseteq V)$.

As restrições 3.8 e 3.9 descrevem a estrutura de árvore da solução. A restrição 3.8 exclui a árvore vazia do conjunto de soluções viáveis, dizendo que o total de arestas da solução deve ser igual ao valor do total de nós menos um. A restrição 3.9 garante que a solução não tem ciclos, ou seja, é uma árvore. As demais restrições caracterizam que os vetores x e y representam os valores 0 e 1 (são vetores de incidência).

3.3 Aplicações

O PSP tem características que se encaixam muito bem em aplicações de distribuição de facilidades ou serviços. O presente trabalho propõe o uso do PSP para modelar o problema da configuração do traçado de redes de distribuição de gás natural, mas são encontradas na literatura outras aplicações.

Canuto et al. [CRR01] e Johnson et al. [JMP00] dão ênfase na aplicação do problema em projetos de rede de acesso local de telecomunicações. Assim como em uma rede de distribuição de gás, o objetivo é equilibrar o potencial de lucro que pode ser obtido oferecendo um serviço a clientes, e o custo de construir a rede. Nesta aplicação, uma rede de fibra óptica deve ser construída. No grafo associado, arestas são segmentos de ruas onde a fibra pode ser colocada, e os custos das arestas são os custos associados em colocar a fibra ao longo dos segmentos das ruas. Os nós do grafo são interseções de ruas e locais onde se encontram clientes. Os prêmios dos nós são estimativas do potencial de lucro a ser obtido providenciando o serviço ao cliente naquele nó.

Outros autores citam a utilização do modelo para finalidades diferentes, mas com as mesmas características, tais como a construção de redes de TV a cabo, redes de distribuição de água, óleo, e energia elétrica [KLM⁺04] [HS06].

Jain e Hajiaghayi [JH06] citam a existência de problemas de coleta de prêmios em teoria dos jogos, especificamente no contexto do compartilhamento de custos, onde o custo de um agente não pode ser maior do que a utilidade do mesmo.

3.4 Problemas relacionados

- *Rooted prize-collecting Steiner tree problem (RPCSTP)*

Na prática, podem existir restrições adicionais em um projeto. O problema do

planejamento de redes de distribuição pode requerer que uma fonte esteja conectada à rede. Em geral, o problema da árvore de Steiner com coleta de prêmios com raiz (*RPCSTP*) é definido como um variante do PSP, onde um vértice adicional $v_0 \in V$, com prêmio bastante alto, é adicionado ao grafo, representando um depósito ou repositório, que deve ser parte de toda solução viável T [Lju04b].

- *Fractional prize collecting Steiner tree problem (FPCSTP)*

Esta variação do PSP é útil no caso do projeto de redes de distribuição, pois as companhias geralmente não estão interessadas em maximizar o lucro absoluto de um projeto, mas sim em maximizar o retorno dos investimentos. O problema correspondente, problema da árvore de Steiner com coleta de prêmios fracionário (FPCSTP) [KLM⁺03], consiste em encontrar uma sub-árvore T de G , onde $c_0 > 0$ representa o custo fixo do projeto, que maximize a razão do lucro sobre os custos:

$$\max \frac{\sum_{v \in V_T} \pi(v)}{c_0 + \sum_{e \in E_T} c(e)} \quad (3.13)$$

- *Piecewise linear prize collecting Steiner tree problem (PWPCST)*

Esta é outra variação interessante do PSP para o projeto de redes de distribuição. Em aplicações práticas é freqüente o caso em que os custos fixos da implementação de um duto, por exemplo, não sejam linearmente dependentes do seu tamanho, e devem ser levados em consideração os custos não lineares dependentes do fluxo de cada duto em particular. Este problema pode ser modelado através do problema da árvore de Steiner com coleta de prêmios dependente de um conjunto de variáveis (PWPCSTP) [Lju04b].

- *Generalized prize collecting Steiner tree problem (GPCSTP)*

Segundo Jain e Hajiaghayi [JH06], neste problema é dado um grafo $G = (V, E)$, um conjunto de pares $P = \{(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)\}$, uma função de custo não negativa c (onde c_{ij} é o custo de conectar o par (i, j)), e uma função de penalidade não negativa π (onde π_{ij} é uma penalidade para o par (i, j)). O objetivo é encontrar uma maneira, a custo mínimo, de conectar um conjunto de arestas e pagar uma

penalidade para os pares de nós que não estão conectados através deste conjunto de arestas.

Assume-se que $P = V \times V$ (todos os pares não ordenados (i, j) , onde $i \neq j$), dado que a penalidade de um par que não precisa estar conectado será zero. Esta é uma generalização do PSP, e no caso em que todas as penalidades são infinitas, tem-se o clássico problema da árvore de Steiner generalizado.

3.5 Revisão da literatura

Esta seção tem o objetivo de mostrar o histórico do PSP. Primeiramente é apresentado o surgimento do problema, assim como os primeiros algoritmos que foram propostos para sua solução (seção 4.1). Mais tarde são citados alguns dos principais algoritmos já propostos para a solução do PSP encontrados na literatura até os dias de hoje.

Segundo Cormen et al. [CLRS01], os algoritmos podem ser divididos em três classes: algoritmos exatos, aproximativos e heurísticos. Os algoritmos exatos garantem o encontro da solução ótima. Os algoritmos aproximativos e os algoritmos heurísticos buscam, de maneira viável, encontrar soluções próximas da ótima. Porém, os algoritmos aproximativos garantem um fator de aproximação para a solução, enquanto os heurísticos não.

Na seção 4.2 são citados os algoritmos exatos, algoritmos que encontram limites inferiores, e algoritmos baseados em estudos da estrutura do problema. A seção 4.3 mostra os algoritmos aproximativos, e descreve mais detalhadamente o algoritmo utilizado no presente trabalho para geração de uma população inicial. Finalmente, a seção 4.4 mostra os dois algoritmos baseados em metaheurísticas encontrados na literatura.

3.5.1 O surgimento do problema

Segev, em 1987 [Seg87], foi o primeiro a considerar o problema da árvore de Steiner com pesos nos nós além dos pesos nas arestas (*Node Weighted Steiner Tree Problem - NWSTP*). Ele notou que quando os pesos dos nós são não negativos e a raiz de uma árvore solução é conhecida, o problema pode ser transformado no problema da árvore de Steiner direcionado. Segev também mostrou que um caso especial do problema, chamado de problema da árvore de Steiner com um único nó com peso (*Single Point Weighted Steiner*

Tree problem - SPWST), é NP-Completo. Neste problema é dado um nó especial a ser incluído em uma solução, e os pesos no resto dos nós são negativos. Ele desenvolveu diversas heurísticas para o problema. Neste modelo, pesos negativos nos nós representam lucros, enquanto pesos não negativos nas arestas representam o custo correspondente para se coletar os lucros.

O termo "coleta de prêmios" foi introduzido em 1989 por Balas [Bal89] no contexto do problema do caixeiro viajante. Ele usou um modelo complementar ao de Segev, sendo os pesos dos nós não negativos, que poderiam ser vistos como penalidades ao não se incluir estes nós em uma solução. Portanto, ao invés de subtrair os pesos dos nós da solução do somatório dos custos das arestas da solução, Balas somava os pesos dos nós que não estavam na solução ao custo das arestas da solução. A partir daí diversos autores introduziram algoritmos para solucionar o problema e suas variações, como é visto a seguir.

3.5.2 Algoritmos exatos, limites inferiores e estudos poliédricos

Os Algoritmos exatos sempre encontram a solução ótima para o problema (se ela existe). Esta solução deve satisfazer de forma ótima a função objetivo correspondente ao problema em questão, respeitando todas as restrições que se aplicam à resolução do mesmo. Entre os métodos exatos são encontrados o *branch-and-bound*, a programação dinâmica, métodos baseados em relaxação Lagrangeana, e métodos baseados em programação linear e inteira, como *branch-and-cut*, *branch-and-price*, e *branch-and-cut-and-price* [PR05].

Nemhauser e Wolsey [NW88] definem uma relaxação do problema P como sendo um problema de otimização, RP , que possui as seguintes propriedades: (i) o conjunto de soluções viáveis de P é um subconjunto das soluções viáveis de RP , e (ii) o valor da função objetivo do problema P não é menor que o correspondente valor de RP , isto é, $v(RP) \leq v(P)$ (para problemas de minimização).

Relaxações de um problema de programação inteira permitem que se obtenham limites da solução ótima do mesmo [EG02]. O limite superior (*upper bound*) de um problema é a quantidade de recursos que é suficiente para resolvê-lo, e o limite inferior (*lower bound*) é a quantidade mínima de recursos necessários para resolvê-lo. É interessante se estabelecer

o maior limite inferior e o menor limite superior possível. Se os dois limites forem iguais, é conhecida exatamente a quantidade de recursos que é tanto necessária quanto suficiente para resolver o problema, ou seja, este é um algoritmo ótimo, sendo que a quantidade de recursos utilizada por qualquer outro algoritmo para a tarefa será maior, ou no melhor caso igual a deste algoritmo.

No entanto, nem sempre é possível construir algoritmos ótimos. Portanto, com a obtenção de limites inferiores é possível saber o quanto de melhora pode-se atingir na busca de um algoritmo para o problema em questão [AHU74]. Se houver uma lacuna entre o algoritmo mais eficiente e o limite inferior conhecido para o problema, está aberta a possibilidade de existirem melhorias.

Em 1987, Segev [Seg87] apresentou o primeiro algoritmo para obter limites para o SPWST. Em seu artigo ele apresentou formulações de multi-comodidade e comodidade simples de fluxo para o SPWST, e mais tarde desenvolveu algoritmos baseados em relaxações Lagrangeanas que foram embutidas em um algoritmo *branch-and-bound*. Ele também propôs heurísticas para gerar soluções iniciais viáveis para o problema.

Em 1991, Fischetti [Fis91] estudou a estrutura facial de uma generalização de um problema chamado problema da arborescência de Steiner, ou árvore de Steiner direcionada. Ele disse que o NWSTP, e conseqüentemente o PSP, pode ser transformado no problema da arborescência de Steiner. Detalhes sobre esta transformação podem ser vistos em [LWP⁺06]. Em 1994 Goemans [Goe94] publicou um estudo teórico sobre a estrutura poliédrica do NWSTP. Engevall et al., em 1998 [EGLV98], propuseram uma nova formulação para o NWSTP, baseada na formulação do problema da AGM. Para encontrar bons limites inferiores, eles usaram uma heurística Lagrangeana e procedimentos de subgradiente baseados na formulação do problema da AGM. Os autores apresentaram resultados computacionais mostrando que seu algoritmo superava o de Segev [Seg87].

Em 2000, Lucena e Resende [LR00] apresentaram um algoritmo cutting plane para resolver o PSP baseado em uma formulação com restrições de eliminação de sub-rotas generalizadas (seção 2.2). Eles utilizaram um pré-processamento, onde o grafo é reduzido [Uch06]. As instâncias das séries K, P, C e D foram utilizadas para os testes. Eles obtiveram sucesso, porém não atingiram o ótimo em todas as instâncias.

Em 2003, Cunha et al. [CLMR03], usando a abordagem proposta por Beasley

[Bea89], formularam o PSP como um problema de floresta mínima restrita, que é facilmente solucionado por um procedimento de relaxação Lagrangeana. Eles encontraram novos limites inferiores, e apresentaram resultados computacionais para as instâncias citadas acima.

Em 2004, Ljubic et al. [LWP⁺06] propuseram uma estrutura algorítmica para a solução exata do PSP. Apresentaram a formulação e implementação de um algoritmo *branch-and-cut* baseado em um modelo de grafo direcionado, e usaram uma combinação de diversos métodos encontrados no estado da arte para o problema da árvore de Steiner. O algoritmo é mais rápido do que o de Lucena e Resende [LR00], alcançou a solução ótima para todas as instâncias, e ainda foi proposta uma nova classe de instâncias.

3.5.3 Algoritmos aproximativos

Os algoritmos aproximativos são uma possibilidade viável de se encontrar soluções para problemas exponenciais onde os métodos exatos são impraticáveis. Estes algoritmos não necessariamente produzem uma solução ótima, mas soluções que estão dentro de um certo fator de aproximação da mesma.

Segundo Cormen et al. [CLRS01], estes algoritmos sempre se aproximam da solução ótima, e a relação de aproximação de um algoritmo é $\rho(n)$ se, para qualquer entrada de tamanho n , o custo C da solução produzida pelo algoritmo de aproximação está dentro de um fator $\rho(i)$ do custo C^* de uma solução ótima:

$$\max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \leq \rho(n) \quad (3.14)$$

Surgiu em 1993 o primeiro algoritmo aproximativo para o problema da árvore de Steiner com coleta de prêmios, e também para o problema do caixeiro viajante com coleta de prêmios. Este algoritmo foi desenvolvido por Bienstock et al. [BGSLW93], com garantias de aproximação 3 e $5/2$, respectivamente.

Em 1996, Goemans e Williamson [GW96] melhoraram o fator de aproximação para 2, para os dois problemas. O algoritmo é baseado em um esquema primal-dual, sua complexidade é $O(n^2 \log n)$, onde n é o número de nós do grafo, e encontra soluções dentro de um fator de $2 - \frac{1}{n-1}$ do ótimo. Para resolver o PSP onde não existe um nó raiz previamente definido, o algoritmo testa cada nó como uma possível raiz. Assim, o tempo total no pior

caso é $O(n^3 \log n)$. Por simplicidade, este algoritmo será chamado de GW nas próximas citações.

Em 2000, Johnson, Minkoff e Phillips [JMP00] descreveram uma implementação 2-aproximação do algoritmo primal-dual GW [GW96]. Em sua implementação, eles fizeram uma melhora na segunda fase do algoritmo, a chamada fase de poda, e também propuseram modificações na primeira fase do algoritmo, a fase de crescimento, para tornar o algoritmo independente da escolha do nó raiz. O algoritmo ficou mais rápido e os resultados são pelo menos tão bons quanto os do algoritmo anterior, e na maioria das vezes significativamente melhor. O resultado foi um algoritmo que executa em tempo $O(n^2 \log n)$ com um fator de aproximação 2. Este foi o algoritmo utilizado no presente trabalho para geração da população inicial, por isso ele será explicado mais detalhadamente na subseção a seguir. Em sua tese, Minkoff [Min00] realizou testes exaustivos comparando este algoritmo com o de Goemans e Williamson, utilizando dois conjuntos de instâncias grandes, um gerado aleatoriamente e outro obtido através de mapas de ruas.

Em 2003, Feofiloff et al. [FFFP03] apresentaram uma prova revisada do algoritmo de Johnson et al [JMP00]. Os autores também propuseram uma modificação do algoritmo GW [GW96] baseada em uma formulação de programação linear um pouco diferente. O novo algoritmo tem seu tempo de execução em $O(n^2 \log n)$ e encontra solução a uma razão de $2 - \frac{2}{n}$ do ótimo.

Algoritmo primal-dual de Johnson et al.

Este algoritmo primal-dual é uma versão modificada do algoritmo GW [GW96] para solucionar o mesmo problema, porém sem considerar um nó raiz fixo. O algoritmo é composto de duas fases, uma fase de crescimento e uma fase de poda. Este algoritmo é chamado de *GW-Unrooted-Growth*, ou seja, ele é uma adaptação daquele algoritmo com a fase de crescimento independente da escolha do nó raiz. Ele encontra a melhor sub-árvore sobre todas as possíveis escolhas de raiz, sem aumentar o tempo de execução do algoritmo.

O algoritmo recebe um grafo $G = (V, E)$, uma função dos custos c de E em \mathbb{Q}_{\geq} e uma função dos prêmios π de V em \mathbb{Q}_{\geq} e retorna uma árvore T em G tal que

$$c(E_T) + \pi(\bar{V}_T) \leq 2opt(PSP(G, c, \pi)) \quad (3.15)$$

Na fase de crescimento, uma solução dual viável é construída e uma solução primal correspondente satisfazendo as condições complementares é encontrada. A solução primal então obtida induz uma floresta F e um conjunto de nós X que não estão nesta floresta. Nesta fase o algoritmo mantém um conjunto de componentes particionadas como ativos e inativos. Cada componente tem um excedente não negativo associado, e inicialmente cada nó é seu próprio componente e o excedente é o valor de seu prêmio.

O algoritmo GW trata o nó raiz de uma maneira especial durante a fase de crescimento, sendo que o componente que contém a raiz (componente raiz) é sempre mantido como inativo, não importando qual é o seu excedente. Um componente é considerado ativo se, e somente se, ele tem o excedente positivo e não contém o nó raiz. Cada aresta ativa tem um déficit, cujo valor inicialmente é o custo da aresta. Já no algoritmo de Johnson et al. é permitido que o componente raiz seja ativo e creça assim como qualquer outro componente enquanto ele tiver excedente positivo. Ele será desativado apenas quando ele já não tiver mais excedente positivo, e poderá se tornar ativo novamente quando unido a algum outro componente com excedente suficiente.

Na fase de poda, o algoritmo GW exclui, além de todos os componentes que não contém a raiz, algumas arestas e nós do componente raiz para diminuir o valor da função objetivo sem desrespeitar as restrições de viabilidade. Este procedimento tem complexidade $O(n^2)$. A fase de poda do algoritmo de Johnson et al. executa em $O(n)$, é significativamente mais rápida e produz soluções que são ao menos tão boas quanto as do algoritmo GW.

O algoritmo de poda de Johnson et al. é conhecido como *Strong Pruning*, e pode ser visto no seguinte procedimento recursivo, inicialmente aplicado ao nó raiz. O valor $nw(v)$ de cada nó v é definido inicialmente como o valor de seu próprio prêmio π_v . O procedimento não precisa de nenhuma informação da fase de crescimento, além da subárvore resultante da mesma.

Algoritmo 1: *StrongPrune*(v)

Entrada: Árvore T e nó raiz v **Saída:** Árvore T'

```

1 para todo filho  $u$  de  $v$  faça
2   StrongPrune( $u$ )
3   se ( $c(u, v) \geq nw(u)$ ) então
4     | Remova de  $T'$  a aresta  $(u, v)$  e a subárvore com raiz em  $u$ 
5   fim
6   senão
7     |  $nw(v) \leftarrow nw(v) + nw(u) - c(v, u)$ 
8   fim
9 fim

```

3.5.4 Algoritmos heurísticos

Os algoritmos heurísticos procuram encontrar boas soluções em um tempo computacional razoável, sem, no entanto, conseguir definir se a solução encontrada é a solução ótima, nem quão próxima ela está da solução ótima. As metaheurísticas são constituídas em cada passo de uma heurística subordinada, a qual deve ser modelada para cada problema específico [Rib96]. Elas são de caráter geral e têm condições de escapar de ótimos locais.

São encontrados na literatura algoritmos heurísticos para algumas variações do PSP, como em [HS06] [Lev03] e [Lju04a]. Porém, apenas duas abordagens metaheurísticas foram publicadas para a solução do PSP em sua versão de minimização de custos - Equação 3.1, como é tratado neste trabalho. Elas são explicadas a seguir.

GRASP híbrido

Canuto et al., em 2000 [CRR01], propuseram um abordagem heurística para a solução do PSP. Trata-se de um algoritmo multi-start com busca local e perturbações, que pode ser chamado de *GRASP* (*greedy randomized adaptive search procedure*) [FR95] híbrido. *GRASP* é uma metaheurística *multi-start* que aplica busca local a soluções iniciais geradas por um procedimento construtivo guloso e aleatório. O algoritmo é híbrido pois são utilizados também *path-relinking* [Glo96] e *VNS* (*Variable Neighbourhood Search*) [MH97] como intensificador e pós-otimizador, respectivamente.

O algoritmo utiliza soluções iniciais geradas pelo algoritmo primal-dual de Goemans

e Williamson [GW96], onde são utilizadas perturbações nos prêmios dos nós, para que a cada iteração o algoritmo produza uma solução diferente. As perturbações são de dois tipos:

- Perturbação por eliminação: para forçar a diversificação, o algoritmo é induzido a construir novas soluções sem alguns dos nós que apareciam na solução obtida na iteração anterior. Isto é feito através da troca dos prêmios de alguns nós persistentes para zero, nós estes que estavam na solução gerada pelo algoritmo GW e permaneceram até o final da busca local na iteração anterior. É considerado um parâmetro α , e são trocados para zero os prêmios associados a um conjunto escolhido aleatoriamente contendo $\alpha\%$ dos nós persistentes observados na iteração anterior.
- Perturbação por troca de prêmios: são introduzidos pequenos ruídos nos prêmios. Para cada nó $i \in V$ é gerado um fator β no intervalo $[1 - a, 1 + a]$, onde a é um parâmetro de implementação, e o prêmio associado ao nó i é substituído por $\pi_i \leftarrow \pi_i \times \beta$.

O algoritmo tem como critério de parada um número máximo de iterações. Dentro desse laço, é chamado o algoritmo primal-dual e, em seguida, o algoritmo de busca local, a solução obtida é inserida no *pool* de soluções elite, se ela atende às condições pré-estabelecidas, e então é aplicado o *path-relinking*. A melhor solução é atualizada se necessário. Ao final de um laço, é feita uma chamada ao procedimento que realiza as perturbações, para que na próxima iteração a solução gerada pelo algoritmo primal-dual seja diferente da anterior. Depois de executadas todas as iterações, é aplicado o algoritmo VNS na melhor solução obtida.

A vizinhança de uma solução na fase de busca local é definida por todas as árvores geradoras mínimas cujo conjunto de nós difere da solução atual por exatamente um nó. Ou seja, duas soluções são vizinhas se elas têm um nó de diferença quando comparadas. O algoritmo utilizado para a construção da AGM foi o algoritmo de Kruskal [Kru56].

A cada iteração é aplicado *path-relinking*. Esta é uma técnica proposta por Glover [Glo96] como uma maneira de integrar intensificação e diversificação em busca tabu. Esta técnica gera novas soluções explorando a trajetória que conecta soluções de boa

qualidade. Partindo de uma solução chamada de solução de partida, um caminho no espaço de soluções que leve a uma solução guia é gerado e explorado em busca de soluções melhores. A técnica é chamada de *path-relinking* porque, em busca tabu, quaisquer duas soluções estão conectadas por uma série de movimentos, o que não acontece em soluções *GRASP*.

Para adaptar a técnica ao *GRASP*, Laguna e Marti [LM99] propõe uma reformulação. Nela, o processo de religamento realiza os movimentos necessários para transformar a solução de partida na solução guia. A solução guia é escolhida aleatoriamente do conjunto de elite. Seja Δ o conjunto de movimentos necessários para transformar a solução de partida na solução guia. A proposta consiste de, repetidamente, escolher o melhor movimento em Δ e aplicá-lo à solução de partida, até que a solução guia seja alcançada. O objetivo é verificar se há alguma solução intermediária, durante a "transformação", que seja melhor que as soluções já conhecidas.

Por fim, é aplicado um algoritmo VNS na melhor solução encontrada nas iterações *GRASP*, onde é considerada uma vizinhança de ordem k . Ou seja, a vizinhança de uma solução é formada pelo conjunto de árvores geradoras mínimas cujo conjunto de nós difere dela por exatamente k nós.

São relatados resultados computacionais para as instâncias das séries K, P, C e D, e pode-se notar a qualidade das soluções e as vantagens no tempo de execução do algoritmo quando comparado ao algoritmo cutting planes de Lucena e Resende [LR00].

Algoritmo memético híbrido

Klau et al. [KLM⁺04], em 2005, desenvolveram uma combinação de um algoritmo memético com um algoritmo exato. A proposta consiste de três partes: uma fase de pré-processamento extensiva que reduz consideravelmente o grafo de entrada; a parte central que é o algoritmo memético baseado em um algoritmo evolucionário e uma sub-rotina exata para o problema em árvores; e a terceira parte consiste em uma pós-otimização resolvendo uma relaxação de um modelo de programação linear inteira (*branch-and-cut*).

É utilizada uma heurística composta de 5 passos para a geração da população inicial, onde é utilizada uma rede de distâncias, que contém o caminho mais curto entre todos os nós do grafo (ver seção 5.1):

1. Selecionar aleatoriamente um subconjunto de nós de demanda;
2. Construir a AGM do grafo induzido por este conjunto;
3. Substituir cada aresta desta árvore pelo caminho mais curto correspondente na rede de distâncias calculada previamente;
4. Encontrar a AGM do grafo induzido por este novo conjunto de nós;
5. Aplicar um algoritmo exato para tornar esta sub-árvore ótima.

São realizados cruzamentos e mutações segundo a abordagem evolucionária [MN92]. Cada solução gerada na população inicial, e cada solução derivada de cruzamentos e possíveis mutações é podada de maneira ótima com respeito às suas sub-árvores, usando um algoritmo exato de melhoramento local.

A cada iteração é feita a seleção de duas soluções pais através de torneio. Essas duas soluções são recombinadas, é aplicada mutação com probabilidade $p_{mut} \in [0, 1]$, e é realizada uma poda através do algoritmo de melhoramento local. Cada solução gerada substitui o pior indivíduo da população e, para manter a diversidade, soluções já existentes na população são descartadas.

Os cruzamentos são realizados com o objetivo de transmitir como herança as propriedades estruturais das soluções pais. Se as duas soluções a serem recombinadas têm pelo menos um vértice em comum, é construída a AGM sobre a união de seus conjuntos de arestas. Se as soluções a serem recombinadas são disjuntas, são escolhidos aleatoriamente dois nós que não estejam em nenhuma das duas soluções, é recuperado o caminho mais curto entre eles (através da rede de distâncias já calculada) e são adicionadas para cada nó v do caminho, todas as arestas pertencentes ao *cluster* de $N(base(v))$. Finalmente é construída uma árvore geradora aleatória sobre estas arestas e é aplicado o algoritmo de melhoramento local. Os clusters são grupos formados através de um procedimento, onde, para cada nó de demanda z é definido um cluster $N(z)$:

$$N(z) := \{v \in V' \mid \forall c \in R' : d'_{vz} \leq d'_{vc}\} \cup \{z\} \quad (3.16)$$

Onde d'_{vz} é o caminho mais curto entre v e z . Portanto, cada nó sem demanda v é designado ao cluster de seu vértice de demanda mais próximo $z = base(v)$.

Como pós-otimizador é utilizado um algoritmo *branch-and-cut*. O resultado da fase de pós-otimização do algoritmo é ignorado na comparação realizada neste trabalho, pois será feita uma comparação apenas com algoritmos heurísticos. Esta proposta se mostrou mais rápida do que a de Canuto et al. [CRR01], porém a qualidade das soluções resultantes do algoritmo memético se mostrou inferior, como pode ser visto no Capítulo 6.

3.6 Banco de instâncias

Os problemas clássicos da Otimização Combinatória possuem bancos de instâncias. Estes bancos são criados para que os algoritmos propostos para solucionar tais problemas possam utilizá-los em seus experimentos e assim realizar testes de desempenho, podendo comparar os resultados entre si.

Os algoritmos que modelam o problema da otimização do traçado através do PSP encontrados na literatura utilizam a mesma base de dados para realizar seus experimentos. O banco de instâncias para o PSP está disponível eletronicamente em <http://www.research.att.com/~mgcr/data/index.html>. Este banco contém basicamente quatro grupos de problemas:

- Grupo K: grafos de 100 a 400 nós e 319 a 1507 arestas. São grafos geométricos gerados de maneira aleatórias mas projetados para terem uma estrutura similar a mapas de ruas.
- Grupo P: grafos de 100 a 400 nós, e 284 a 1144 arestas. São grafos não estruturados e projetadas para que seus nós tenham grau e prêmios constantes.
- Grupo C: grafos de 500 nós e de 625 a 12500 arestas. São 40 grafos derivadas de instâncias do problema da árvore de Steiner disponíveis na *OR-Library* [Bea90]. Para cada uma das 20 instâncias da série original C dois conjuntos, A e B, foram gerados atribuindo-se prêmio zero a nós não terminais, e um valor de prêmio aleatório no intervalo $[1, \text{max_prize}]$ para os nós não terminais. Para o conjunto A, $\text{max_prize} = 10$; para o conjunto B, $\text{max_prize} = 100$.
- Grupo D: grafos de 1000 nós e de 1250 a 25000 arestas. São 40 grafos derivadas de instâncias do problema da árvore de Steiner, da mesma forma que o grupo C.

Os grupos K e P foram criados por Johnson et al. [JMP00], e mais detalhes de como essas instâncias foram geradas podem ser vistos em [Min00]. Os grupos C e D foram criados por Canuto et al. [CRR01].

Capítulo 4

Transgenética computacional

A transgenética computacional faz parte do amplo contexto da computação evolucionária. Por isso, este capítulo visa apresentar as teorias nas quais a computação evolucionária se baseia, e mostrar as diferenças encontradas nas abordagens mais conhecidas e suas definições.

A seção 4.1 aborda as teorias clássicas da evolução das espécies: a teoria de Darwin, de Lamarck, de Behe, a teoria Neutra e o Neodarwianismo. As seções 4.2 e 4.3 apresentam as bases da transgenética computacional: a teoria da endosimbiose e a transferência horizontal. Na seção 4.4 é feito um levantamento do estado da arte em algoritmos baseados neste contexto. A transgenética computacional ganha ênfase na subseção 4.5, onde são definidos os seus algoritmos.

4.1 A evolução das espécies

A questão da evolução das espécies é até hoje motivo de discussões e existem opiniões distintas sobre o assunto. Porém existem duas teorias bastante conhecidas, a de Jean Baptiste Lamarck e a de Charles Darwin [?]. Lamarck defendia a teoria da herança dos caracteres adquiridos, enquanto Darwin dizia que a evolução se dava pela seleção natural, através da luta pela existência.

Lamarck escreveu duas obras defendendo sua teoria: "Filosofia Zoológica" e "História Natural dos Invertebrados". Segundo Lamarck, o ambiente em que vivem e o modo de vida dos animais influiriam neles de modo a adaptá-los cada vez mais e melhor às novas condições. As mudanças adquiridas na vida de um animal seriam transmitidas aos seus descendentes. Esta é a lei dos caracteres adquiridos.

Ao contrário do que afirmava Lamarck, para Darwin, a causa da transformação de uma espécie em outra seria inteiramente extrínseca ao seu organismo. A luta pela sobrevivência é que seria o verdadeiro motor da evolução, permitindo que continuassem a existir apenas os indivíduos mais aptos. Esta é a teoria que ele apresentou em seu famoso livro "On the Origin of Species by Means of Natural Selection" [1]. O termo "Darwinismo" é usado para designar vários processos relacionados com as ideias de Darwin. O eixo da proposta de Darwin desenvolveu-se segundo os seguintes argumentos:

- Os organismos têm uma fertilidade tão grande que o número de indivíduos numa população tende a aumentar de maneira exponencial, caso todos os indivíduos consigam reproduzirem-se com sucesso;
- As populações normalmente apresentam um tamanho estável, flutuando dentro de certos limites;
- O aumento da disponibilidade dos recursos naturais não acompanha o crescimento populacional;
- As populações de organismos apresentam variabilidade;
- Parte dessa variação pode ser transmitida aos descendentes;
- Parte da variação ocorre em caracteres que afetam as chances de sobrevivência e reprodução dos organismos.

A proposta de Lamarck logo caiu em descrédito, e a de Darwin vem sendo motivo de um debate que se arrasta há muito tempo. De uma forma geral, a teoria de Darwin tornou-se a teoria mais aceita no meio acadêmico, especialmente depois da reforma conceitual introduzida na década de 50, e que renomeou a abordagem como Teoria Sintética ou Neodarwinismo. Da forma como foi concebida a chamada evolução darwiniana exige que as variações individuais e funcionais sejam sempre de pequena amplitude (gradualismo). É bem conhecida a frase utilizada pelos autores evolucionários que afirma: "Natura non facit saltum - a natureza não dá saltos".

Existem algumas alternativas ao chamado "Darwinismo radical". São várias as linhas de raciocínio que questionam a suficiência da teoria de Darwin para explicar a evolução da vida. Serão citadas aqui as teorias de Behe [Beh96] e de Kimura [Kim68].

A primeira nega a suficiência do mecanismo do gradualismo para explicar todos os processos de especiação - emergência de espécies diferentes. A chamada "evolução" decorrente do suave acúmulo de adaptações tem um problema intrínseco: ao ter seu guiamento dado pelo processo de seleção natural, obriga que o acúmulo das adaptações seja sempre útil. Ou seja, a modificação estrutural de um organismo teria que seguir um caminho em que cada modificação fosse útil. Behe [Beh96] aponta um erro no raciocínio darwiniano que, segundo ele, seria absurdamente reducionista. O autor destaca o chamado paradoxo da formação de sistemas de complexidade irreduzível (SCI). Estes sistemas têm as seguintes características:

- São compostos por um conjunto de mecanismos específicos e que devem funcionar de modo simultâneo e integrado;
- A função do SCI só pode ser alcançada quando cada componente apresenta um bom funcionamento. A regulação de um SCI só pode ser feita com o sistema operando em sua versão final;
- O SCI atende plenamente ao seu objetivo proposto, mas os componentes, em si, não obrigatoriamente desempenham funções de natureza semelhante às desempenhadas pelo SCI.

Behe mostra que os SCIs são abundantes na natureza. Esses problemas ficam mais visíveis quando os mecanismos de evolução devem ser interpretados e compreendidos ao nível molecular. Neste contexto, a proposta Darwiniana comporta-se como uma verdadeira "caixa preta"[Beh96].

O problema com a formação dos SCIs é que eles são sistemas que representam um alto investimento biológico e são incapazes de exibirem utilidade antes de serem completamente montados. A constituição de um SCI acumularia o peso de uma inutilidade crescente até o momento final de seu funcionamento. A formação de um SCI exigiria a admissão de um processo de redução contínua da adequação das gerações de indivíduos portadores daquelas mutações acumuladas, até o momento de seu funcionamento (ocasião em que um enorme salto positivo na adequação seria sentido).

Além disso, o SCI supostamente deveria dispensar - ou mesmo contrariar - o guiamento proposto pela seleção natural. Como associado a cada SCI deve corresponder um

conjunto de genes que lhe dá elaboração dentro do organismo, o darwinismo deveria admitir o acúmulo de mutações inúteis e, finalmente, o salto evolucionário correspondente à conclusão do SCI.

Outra linha de raciocínio é a Teoria Neutra, que foi proposta por Motoo Kimura em meados da década de 60 [Kim68]. Ela assume que a maioria das variações encontradas em seqüências de DNA e proteínas tanto dentro como entre diferentes espécies é neutra com relação à seleção. Dessa forma, Kimura sugere que a maior parte das variações genéticas em nível molecular seria seletivamente neutra e sem qualquer valor adaptativo.

Por se mostrarem necessários, os saltos evolucionários despertaram diversas teorias. Uma das possibilidades poderia estar associada ao compartilhamento de genes entre diferentes espécies. Outra possibilidade, ainda mais radical, poderia estar associada à fusão de indivíduos simbiotes - caso especialmente visível no contexto dos microorganismos [MS89].

O "pluralismo" procura uma proposta que soma o processo de seleção natural com os processos de desenvolvimento. O pluralismo imagina a evolução como uma solução de compromisso entre a contingência e seu conteúdo completamente aleatório, e o determinismo imposto pela estrutura da informação genética já existente, bem como pela ação do meio ambiente. Para o pluralismo nem toda mudança evolutiva representaria a sobrevivência do mais apto, como preconiza o neodarwinianismo. Contudo, o processo evolucionário exige ainda a consideração de elementos mais complexos do que o anteriormente exposto.

4.2 Teoria da endosimbiose

A simbiose é um fenômeno amplamente disseminado entre os seres vivos, e ressalta características notáveis em termos de contribuições potenciais aos processos evolutivos. É uma relação mutuamente vantajosa entre dois ou mais organismos vivos de espécies diferentes. Na relação simbiótica, os organismos agem ativamente em conjunto para proveito mútuo, o que pode acarretar em especializações funcionais de cada espécie envolvida [r78]. Ela se apresenta como um paradigma de interações cooperativas, em certa oposição ao panorama dominante da competição e sobrevivência do mais forte. A simbiose en-

contra um amplo raio de atuação, envolvendo desde o nível celular até as complexas sociedades de animais superiores.

A endossimbiose, contudo, representa um acréscimo principalmente ao quadro da evolução celular. Um endossimbionte é qualquer organismo que vive no interior do corpo ou das células de outro organismo, realizando uma relação ecológica designada como endossimbiose [r78]. Por exemplo, algumas bactérias, fixadoras de azoto vivem em nódulos, nas raízes de algumas leguminosas; alguns corais, em recifes, hospedam algas unicelulares; vários insetos contêm bactérias endossimbiontes; entre outros exemplos possíveis.

A Teoria da Endossimbiose foi criada por Lynn Margulis [Mar91] e propõe que organelas ou organóides, que compõem as eucélulas (células eucariontes) tenham surgido como consequência de uma associação simbiótica estável entre organismos. Mais especificamente, esta teoria postula que os cloroplastos e as mitocôndrias (organelos celulares) dos organismos eucariontes (com um verdadeiro núcleo celular) têm origem num procarionte autotrófico - provavelmente um antepassado das cianobactérias atuais - que viveu em simbiose dentro de outro organismo, também unicelular, mas provavelmente de maiores dimensões, obtendo assim proteção e fornecendo ao hospedeiro a energia fornecida pela fotossíntese.

Como um primeiro resultado palpável, a teoria da endossimbiose foi capaz de prover uma explicação bastante razoável para a origem de algumas organelas intracelulares, em especial mitocôndrias e plastídeos (cloroplastos). A Figura 4.1 exibe o suposto processo de incorporação de uma bactéria aeróbica procariótica ao corpo de uma célula primitiva. Segundo Margulis [Mar98], a formação das mitocôndrias teria resultado de uma fusão simbiótica.

Margulis ressalta que a evolução endosimbiótica possui um componente "lamarckista" ao viabilizar a herança de conjuntos genômicos adquiridos. Outros exemplos de processos de evolução endosimbiótica são encontrados entre alguns vermes que abrigam em seus tecidos (extracelularmente) microorganismos fotossintetizadores (por exemplo, a *Convoluta roscoffensis*, que abriga grandes populações de algas do gênero *Platymonas*). Algumas células eucarióticas possuem cianobactérias verdadeiras no interior de seus citoplasmas, como é o caso da espécie *Cyanophora paradoxa* [ABL⁺97]. Propostas mais extremas da teoria endosimbiótica postulam a origem de centrossomos e cinetoplastos

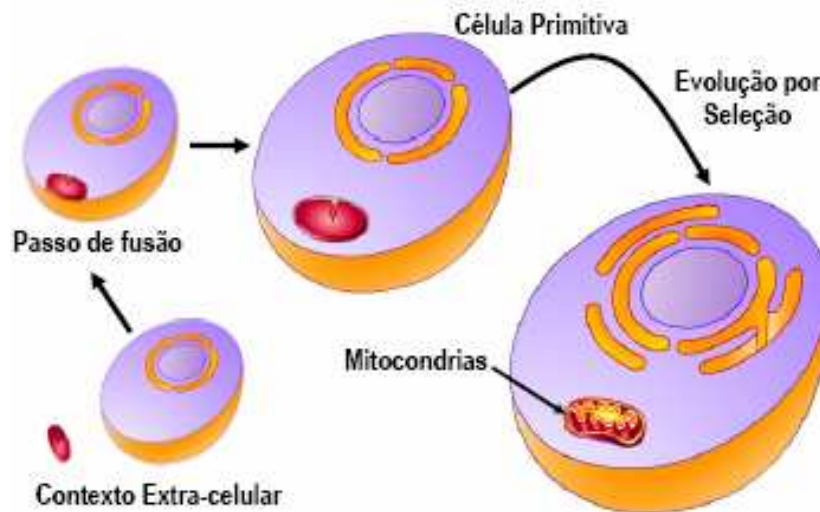


Figura 4.1: Formação das mitocôndrias.

também como produto de associações de eucariotos com bactérias.

A participação de eventos de simbiose-endosimbiose e a transferência genética horizontal como contribuição para o aparecimento de novidades evolutivas [Mar91] tem tido um estímulo experimental-teórico mais recente com descobertas como a presença de mais de 200 genes em seres humanos de possível origem bacteriana (e ausentes de seres muito mais próximos a nós, como fungos, moscas, ou vermes), sugerindo uma transferência horizontal relativamente mais recente [Lan01].

Na endosimbiose, a competição não representa a única via promotora do aperfeiçoamento, tão pouco a sobrevivência e reprodução dos mais aptos. A abordagem enfatiza a cooperação que pode ser traduzida por ações coordenadas e troca de informações genéticas. Margulis, ao definir o princípio da endosimbiose, afirmou que "a evolução do genoma no longo prazo é muito mais influenciada por interações extra-intracelulares do que pela mutação ou seleção natural".

A teoria de Margulis encontrou forte oposição dos defensores do darwinismo. A teoria da evolução endosimbiótica foi rejeitada inicialmente por falta de comprovação científica, contudo atualmente representa uma abordagem bem aceita.

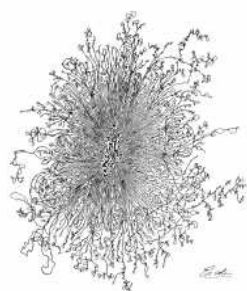
A compreensão da evolução microbial tem evoluído fortemente nos últimos anos principalmente com a análise de genomas animais. Na década de 70 pensava-se que as bactérias eram simples máquinas de reprodução. Hoje se aceita que as populações microbiais possuem capacidade de cooperar e coordenar ações, bem como habilidades de recuperar

informações embutidas no contexto extracelular. Além disso, essas pequenas criaturas podem realizar trocas diretas de informações. A habilidade das bactérias se comunicarem e coordenarem ações através de sinais moleculares recebeu o nome de "Quorum Sensing".

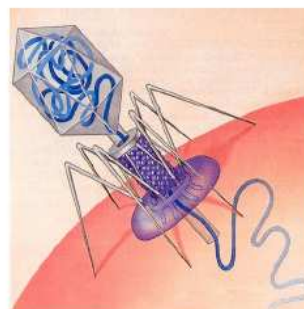
4.3 Transferência horizontal

Existem vários processos de interações extra-intracelulares que podem redundar em alterações permanentes do código genético. Um conjunto de tais processos está reunido sob o nome de transferência horizontal. A transferência genética horizontal é bem conhecida entre bactérias, mas foi somente nos últimos dez anos que ela foi reconhecida em plantas e animais superiores [r78]. Os mecanismos de transferência horizontal comportam os plasmídios, a transformação e transdução.

Os plasmídios são partículas genéticas móveis (anéis de DNA) que podem ser intercambiadas entre certas células. A transformação é um processo que permite transferir DNA "puro" de uma bactéria para outra, Figura 4.2a. As bactérias podem lançar DNA no contexto extracelular através de um processo denominado lise. A transdução é um processo que permite o transporte e transferência de genes com o auxílio de um vírus, Figura 4.2b.



(a) E. Coli realizando *lise*



(b) Ataque do Fago T4

Figura 4.2: Processo de transferência horizontal.

Os transposons são seqüências de DNA e fazem parte de outros elementos genéticos, os cromossomos ou os plasmídios. Eles se movem de uma locação para outra no DNA dentro da mesma célula, formando uma transposição ou permutação. Os transposons atuam em dois mecanismos diferentes. Um mecanismo que permite cortar e colar

sequências de DNA, e um outro que permite apenas copiar e colar.

As evidências de que as células e seus genomas são o palco de eventos complexos, ordenados, e altamente criativos em suas interações com o ambiente, são reforçadas pelas descobertas efetuadas nas últimas décadas [Sha99]. Na década de 50 a descoberta foi vista como uma heresia, porém na década de 70 vários outros pesquisadores redescobriram o fenômeno. Este processo atua em toda a biosfera, com bactérias e vírus servindo tanto como intermediários para o tráfego de genes quanto como reservatórios para a multiplicação e recombinação de genes [r78].

4.4 Algoritmos baseados em endosimbiose

O primeiro trabalho apresentando elementos para uma abordagem simbiótica no contexto computacional deve-se a Hillis, em 1990 [Hi190]. Os trabalhos na computação evolucionária sugerindo elementos de mimetização endosimbiótica surgem com Bull e Fogarty [BF95]. Baseado nesta proposta, Kim et al. [KKK01] propõem um algoritmo co-evolucionário que introduz o conceito de indivíduos endosimbiontes, formados pela união de dois outros indivíduos.

Outros algoritmos são apresentados na literatura dando mais ênfase à co-evolução, ou simbiose (não propriamente a endosimbiose). Nessa linha destacam-se os trabalhos de Tsujimura et al. [TMG01], Pagie e Mitchell [PM02], Rosin e Belew [RB97], Husbands [Hus94], De La Cal Marín e Ramos [DLCMR04], Kim et al. [KPK03], e Michaelian [Mic98].

Uma abordagem correlata, baseada em iteração evolucionária entre indivíduos de naturezas diferentes é apresentada por Kubota et al. [KSF96b]. O autor propõe um algoritmo evolucionário denominado *VEGA* (*Virus-Evolutionary Genetic Algorithm*) em que a evolução é simulada através de propagação horizontal e vertical da informação genética. O algoritmo *VEGA* desenvolve o processo clássico de evolução acrescido de uma etapa de transdução viral. A literatura reporta alguns experimentos computacionais usando a arquitetura *VEGA* [KAFS97], [KFAS97], [KSF96b], [KSF96a].

4.5 Os algoritmos transgenéticos

Os algoritmos transgenéticos pertencem à classe dos algoritmos evolucionários. Eles baseiam sua metáfora em um processo de endossimbiose/*Quorum Sensing*, e são capazes de considerar outras alterações genéticas típicas do contexto intracelular que independam de mutações ou de compartilhamento sexual de genes.[GG02].

Segundo Goldberg e Goldberg [GG02], os conceitos anteriores são adaptados ao contexto computacional da seguinte maneira. Os algoritmos transgenéticos trabalham basicamente com populações de indivíduos de naturezas distintas em um processo endossimbiótico de troca de informações, consideradas as possibilidades de interferência de um dado meio ambiente. A metáfora é desdobrada em três níveis:

1. Uma população de cromossomos hospedeiros que representa a memória corrente da busca, o conjunto de configurações em exame.
2. Vetores transgenéticos, atuadores de natureza diversa da dos cromossomos. São utilizados para promover a intensificação e diversificação da busca.
3. O contexto extracelular rico em informações ambientais capazes de influenciar vetores e cromossomos.

A abordagem propõe utilizar informações não genéticas obtidas *a priori* para guiar a ação de seus vetores. Essa característica abre o processo evolucionário a influências do meio ambiente e a supostas experiências evolucionárias anteriores.

Por se tratar de uma abordagem evolucionária, informações obtidas ao longo do processo de evolução também são utilizadas. A transgenética, todavia, por ter sido constituída dentro de um contexto de endossimbiose, pode dispensar a troca direta de informações entre os cromossomos que evoluem (recombinação da população de hospedeiros, ou cruzamentos). As flutuações aleatórias do reservatório gênico (mutações), igualmente podem ser deixadas de lado no processo, ou podem representar um papel secundário na evolução, como preconiza a teoria da evolução neutra.

Assim, as trocas de informação entre os indivíduos das populações em processo de endossimbiose são feitas pelas regras que administram o processo de *Quorum Sensing*. Consequentemente, a troca de informações entre indivíduos da população de hospedeiros

é realizada através de intermediação no fluxo extracelular, com o auxílio dos vetores transgenéticos.

São incorporadas ao algoritmo informações obtidas *a priori* com três finalidades adicionais:

1. Gerar e realimentar a evolução, na medida em que as regras de administração do processo podem ser alteradas pelo resultado da busca;
2. Avaliar o fluxo de informação dentro de cada nível e na transição entre os níveis. Os vários estágios de avaliação introduzem a possibilidade de que a informação seja submetida a diversos processos de seleção bem como que a informação incorporada *a priori* possa ter alguns de seus graus de liberdade ajustados através de parâmetros;
3. Conduzir a evolução da população de hospedeiros, independentemente de um processo reprodutivo sexual (recombinação entre indivíduos) ou de mutações, características indispensáveis da abordagem genética e memética, porque emprega um mecanismo transgenético para efetuar a endosimbiose (evolução induzida).

Os algoritmos transgenéticos podem decompor o processo de seleção em vários níveis, tais como:

- Competição entre as regras de administração pela oportunidade de fornecerem informações que se incorporem aos vetores de manipulação;
- Competição entre os vetores pela possibilidade de manipularem os cromossomos;
- Competição entre as informações contidas nos cromossomos com as informações transportadas pelos vetores para preservarem sua continuidade dentro dos genótipos.

O nível associado às regras de administração do modelo transgenético corresponde ao projeto e coordenação de um processo de inter-relações entre populações de indivíduos de naturezas diferentes (co-evolução) [ER64]. A transgenética sugere uma modelagem simplificada das regras de administração da co-evolução entre a população de cromossomos e os vetores transgenéticos, considerando-as expressas através de procedimentos heurísticos.

Algoritmo 2: Meta-algoritmo transgenético.

Entrada: Grafo G **Saída:** Cromossomo C

- 1 Criar uma população inicial de cromossomos;
 - 2 Carregar as regras transgenéticas (RET);
 - 3 Carregar o Banco de Informações Genéticas (BIG);
 - 4 **repita**
 - 5 | Gerar vetores de manipulação;
 - 6 | Selecionar cromossomos para manipulação;
 - 7 | Manipular os cromossomos conforme as RET;
 - 8 | Atualizar as RET e o BIG;
 - 9 **até atender algum critério de parada**
-

O Algoritmo 2 apresenta um meta-algoritmo transgenético. A população inicial de cromossomos hospedeiros pode ser criada da mesma forma que em outros algoritmos evolucionários clássicos: geração aleatória, geração através de uma heurística de baixo custo computacional, etc. As regras transgenéticas serão definidas conforme a aplicação, e irão administrar a manipulação dos cromossomos (ver subseção 4.5.1). O banco de informações genéticas é obtido através de informações teóricas sobre a aplicação, e pode ser atualizado durante a evolução (ver subseção 4.5.2).

Os primeiros resultados da transgenética computacional foram obtidos com um algoritmo que empregava exclusivamente plasmídios como vetor de manipulação. Esse tipo de algoritmo foi denominado ProtoG [GGS01].

4.5.1 Regras transgenéticas (RET)

O controle da evolução das populações de vetores, cromossomos e do meio ambiente é realizado através de três classes de regras denominadas "transgenéticas". No estado atual da pesquisa existem três tipos de regras, são as regras do tipo 1, 2 e 3.

Regras do tipo 1: são responsáveis pela constituição de informações que serão transportadas pelos vetores transgenéticos. As regras tipo 1 podem utilizar qualquer conhecimento armazenado no banco de informações. O modo como essas informações serão incorporadas ao cromossomo são definidas pelas regras do tipo 2.

Regras do tipo 2: são orientações para constituição de operadores de manipulação genética. Se um vetor não é bem sucedido em sua tentativa de manipulação (ataque), diz-se que o cromossomo "resistiu" à manipulação. O critério de resistência à manipulação

pode levar em conta a variação de adequação decorrente da manipulação. Além disso, a avaliação do vetor pode considerar o número de manipulações bem sucedidas do vetor. O sucesso do vetor permite avaliar as regras que o constituíram. A ação dos vetores de manipulação sobre a população de cromossomos é estabelecida pelas regras do tipo 3.

Regras do tipo 3: desempenham parte do papel atribuído ao *Quorum Sensing*, mas são acrescidas de outras especificidades necessárias à implementação computacional do processo. Podem se referir ao número de vetores que participarão de cada iteração do algoritmo, aos critérios que definirão quantos cromossomos serão atacados, a seqüência de ataque dos vetores, etc.

Denomina-se regra R_{ij} a j -ésima regra do tipo i . O conjunto das regras de administração e os procedimentos inerentes aos vetores constituem um esquema de manipulação. Em um algoritmo poderão existir várias regras do tipo 1 e 2. Através da Figura 4.3, pode-se observar que as regras tipo 3 orquestram o processo completo.

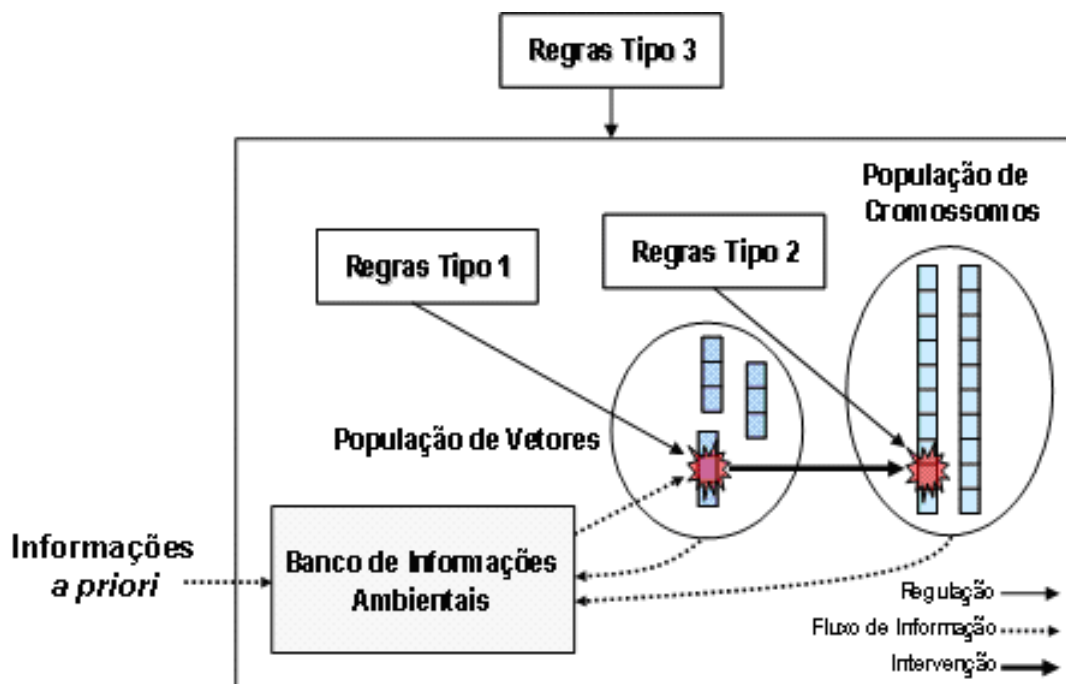


Figura 4.3: Atuação das regras transgenéticas.

4.5.2 O banco de informações genéticas (BIG)

O banco de informações genéticas é o repositório geral de informações. Ele 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 (formato das soluções parciais do problema), quanto sob um formato abstrato que representa outras formas de informação como, por exemplo, procedimentos ou regras de compartilhamento genético.

Na transgenética computacional, a obtenção de informações externas à população de cromossomos hospedeiros possui papel fundamental. Informações úteis sobre como organizar uma solução de boa qualidade podem ser encontradas em resultados teóricos do problema a ser solucionado ou mesmo através de procedimentos heurísticos.

Um exemplo é utilizar a AGM de um grafo G para apresentar seqüências de vértices úteis à constituição da rota para o problema do caixeiro viajante sobre o mesmo grafo. Na mesma linha de raciocínio, o caminho mais curto entre um determinado par de vértices em um grafo G pode também sugerir uma boa seqüência de vértices para ser utilizada como informação no problema da árvore de Steiner.

A Figura 4.4 ilustra as possíveis fontes de informação que compõem o banco de informações genéticas.

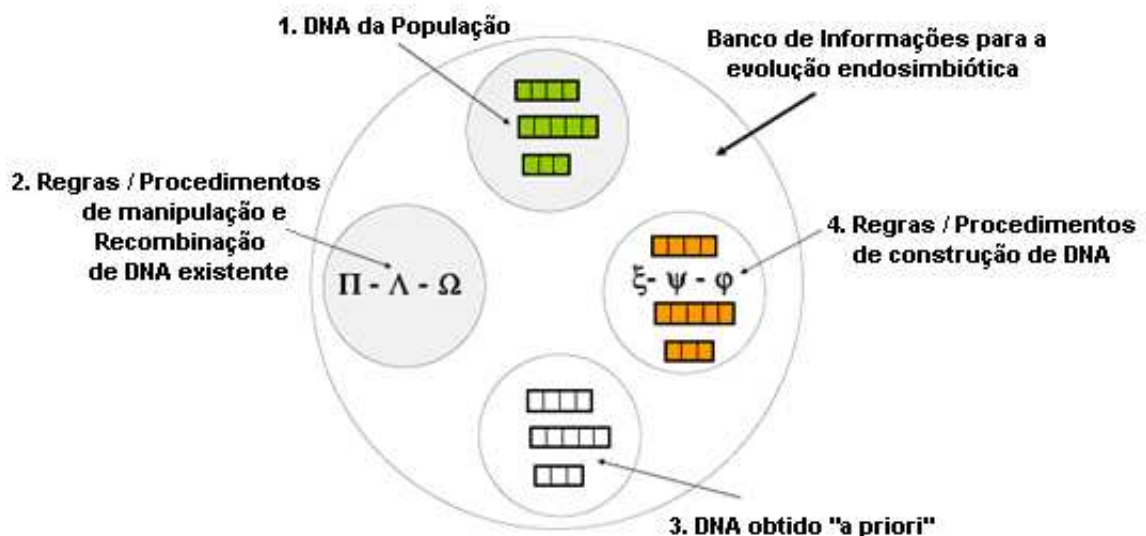


Figura 4.4: Componentes do banco de informações.

4.5.3 Vetores transgenéticos

Os vetores transgenéticos, assim como os microorganismos, são especializados no transporte de certos tipos e formatos de informação. 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 transgênicos provavelmente alterarão o cromossomo e sua adequação.

A Figura 4.5 mostra os vetores atualmente propostos, e a Tabela 4.1 resume os procedimentos que compõem o método de manipulação dos vetores transgênicos. 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 transposon quando seu método utiliza os procedimentos p_1, p_2, p_5 e p_6 . No caso dos vírus, os métodos utilizados são p_1, p_2, p_3 e p_4 , e p_1 poderá ser constituído por uma função probabilística. Os plasmídios recombinados são semelhantes aos plasmídios, porém, além dos métodos de ataque e transcrição, utiliza também o método de recombinação, p_6 .





Nome do Vetor	Metáfora	Procedimentos Embutidos
Vírus		$\Phi = (p_1, p_2, p_3, p_4)$
Plasmídeo		$\Phi = (p_1, p_2)$
Transposon		$\Phi = (p_1, p_2, p_5, p_6)$
Plasmídeo Recombinado		$\Phi = (p_1, p_2, p_6)$

Figura 4.5: Procedimentos que definem os vetores transgênicos.

A cadeia de informação que será transportada pelo vetor transgênico pode ser obtida através de:

- Células doadoras (cromossomos);
- Recombinações endosimbióticas entre os vetores do processo (procedimentos de composição de fragmentos de DNA);
- DNA obtido *a priori*;
- Procedimentos construtivos.

Tabela 4.1: Alguns procedimentos típicos dos vetores transgenéticos.

Procedimento	Descrição
p_1 Ataque (A)	Define o critério de avaliação que estabelece quando o cromossomo é suscetível à manipulação do vetor. $A : S_i \rightarrow \{v, f\}, i = 1, 2, \dots, n$
p_2 Transcrição(I)	Se $A(S_i) = v$, o procedimento define como a informação I , transportada pelo vetor, será transferida para o cromossomo.
p_3 Bloqueio(Ψ)	Torna o resultado da manipulação inviolável por um período de tempo (número de iterações, gerações, etc).
p_4 Desbloqueio($\bar{\Psi}$)	Torna irrestrito o resultado de uma manipulação.
p_5 Identificação(Λ)	Identifica que posições serão utilizadas para limitar a operação do vetor
p_6 Recombinação(Π)	Identifica a origem e o comprimento de cada subcadeia de informação transportada pelo vetor de manipulação.

A Figura 4.6 exibe a associação entre as fontes de informações e os vetores transgenéticos. Os transposons transportam basicamente regras para manipulação do código genético. Os plasmídios, plasmídios recombinados e vírus transportam DNA puro, variando, nesse caso, a forma como o DNA é obtido e transcrito.

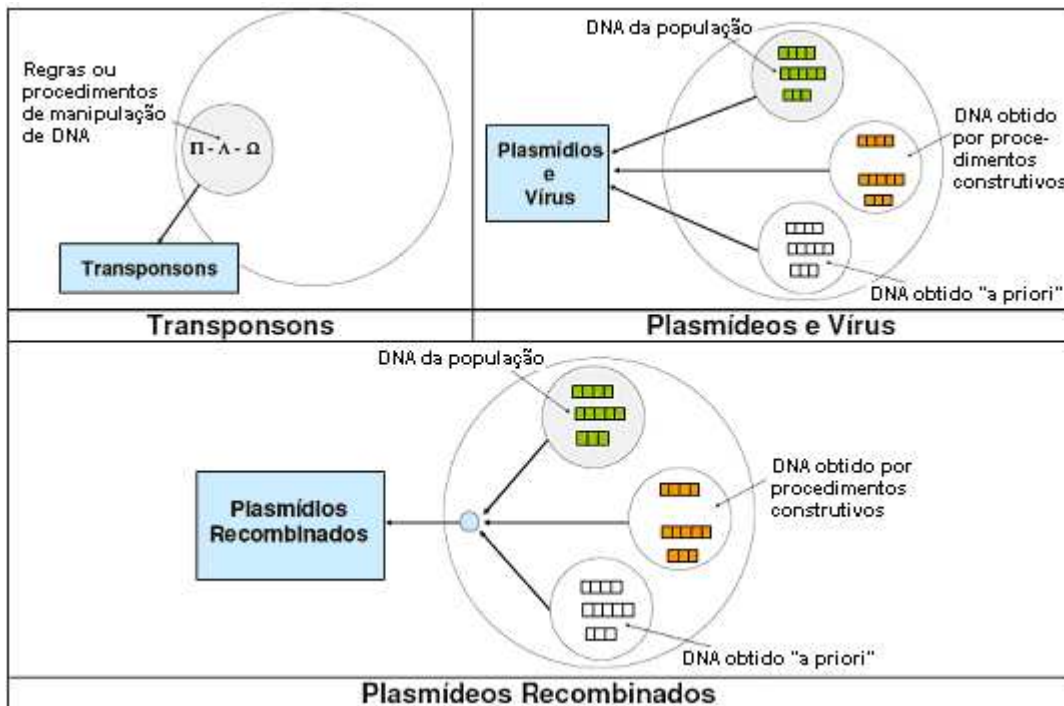


Figura 4.6: Origem da informação dos vetores.

4.6 Algoritmos transgenéticos já desenvolvidos

A transgenética computacional foi introduzida por Gouvêa [Gou01] em sua tese de doutorado. O trabalho explorou basicamente dois tipos de ATs: os que usam o fluxo extracelular, e os que não o fazem. O teste de viabilidade da proposta da TC e de seus algoritmos foi realizado através da solução de dois problemas combinatórios cuja abordagem evolucionária clássica não reporta, na literatura, bons resultados - O Problema Quadrático de Alocação (PQA) e Problema da Coloração em Grafos.

A metáfora tem sido bastante aplicada a problemas relacionados à exploração de petróleo e GN. Destacam-se os trabalhos de Goldberg et al. aplicado à exploração de petróleo [GGMN04] [GGMN02], programação de sondas de produção [GGC02], localização de poços [GGS02], passeio do pistoneio periódico [GDG06], e de Castro et al. [CGG02] que aborda o problema de gás lift.

Os trabalhos de Castro [Cas04] e Goldberg et al. [GCG04] apresentam uma proposta de desenvolvimento de algoritmos para o problema de dimensionamento de dutos em uma rede de distribuição de GN. Entre os algoritmos propostos, está um algoritmo transgenético do tipo ProtoG. Os agentes utilizados são as Partículas Genéticas Móveis (PGMs). As informações que compõem as PGMs são obtidas a partir de uma heurística para o dimensionamento dos dutos. No algoritmo, uma PGM de comprimento q tem seu conteúdo representado por uma seqüência de q duplas (identificação do duto, diâmetro). O método de transcrição dessa PGM simplesmente substitui os diâmetros especificados no cromossomo pelos valores correspondentes aos trechos da PGM.

Goldberg et al. [GGMN05] apresentam um caso especial do problema geral da determinação de uma configuração para um sistema de co-geração de energia que utiliza o GN como fonte energética. O problema visa determinar a configuração de custo mínimo de um sistema de co-geração. Para solucioná-lo, são apresentados e comparados algoritmos evolucionários desenvolvidos de acordo com as metáforas dos AGs, AMs e ATs. O algoritmo ProtoG implementado utiliza apenas o vetor PGM. O algoritmo ProtoG obteve um desempenho superior ao dos demais algoritmos do experimento em todo o espectro das instâncias.

Além dos problemas relacionados a petróleo e gás, tem-se diversas aplicações em

problemas clássicos de otimização combinatória, como em [GG01], [GG02], [GGC04], [GGR03], [GBG06] e [LFGG04]. A Tabela 4.2 resume os ATs desenvolvidos desde o início dos estudos na abordagem.

Tabela 4.2: Algoritmos transgenéticos já desenvolvidos.

Ano	Autores	Problema abordado
2001	Gouvêa [Gou01]	Trabalho pioneiro, tese de doutorado
2001	Goldbarg et al. [GG01]	Problema da coloração em grafos
2002	Goldbarg e Goldbarg [GG02]	Problema quadrático de alocação
2002	Goldbarg et al. [GGMN02]	Exploração de petróleo
2002	Gouvêa et al. [GGC02]	Programação de sondas de produção
2002	Goldbarg et al. [GGS02]	Localização de poços de petróleo
2002	Castro et al. [CGG02]	Gás Lift
2003	Goldbarg et al. [GGR03]	Problema do Caixeiro Viajante
2003	Ramos et al. [dOR03]	Problema do Caixeiro Viajante
2004	Goldbarg et al. [GGMN04]	Exploração de petróleo
2004	Goldbarg et al. [GGC04]	Flow shop de permutação
2004	Goldbarg et al. [GCG04]	Dimensionamento de dutos
2005	Goldbarg et al. [GGMN05]	Sistemas de co-geração
2005	Leite et al. [LFGG04]	Distribuição de vídeo
2006	Barboza [Bar05]	Distribuição e estocagem de petróleo
2006	Goldbarg et al. [GDG06]	Passeio do pistoneio periódico
2006	Bagi [BGG06]	Problema do caixeiro comprador
2006	Schmidt [SGG06]	Distribuição de gás natural

Atualmente estão em desenvolvimento algoritmos transgenéticos que utilizam em conjunto informações dos contextos extra e intracelulares. Utilizando essa abordagem tem-se o presente trabalho, e o algoritmo de Bagi [Bag06] e Goldbarg et al. [GBG06] para o problema do caixeiro comprador. Este algoritmo apresenta excelentes resultados, reportando 11 novas melhores soluções para as instâncias do problema.

Capítulo 5

Algoritmo transgenético para o PSP

O algoritmo proposto neste trabalho utiliza dois vetores da transgenética computacional para solucionar o problema abordado, são eles o plasmídeo e o transposon, explicados detalhadamente no capítulo anterior. O algoritmo transgenético é hibridizado com um procedimento de intensificação através de *path-relinking*. Em linhas gerais, o algoritmo é composto da seguinte seqüência de passos.

- Carregar um banco de informações genéticas;
- Gerar uma população inicial;
- Atacar os indivíduos da população com os vetores da transgenética;
- Aplicar *path-relinking* ao cromossomo manipulado;

As soluções do PSP são representadas através de cromossomos que identificam os pontos atendidos pela rede. Estes pontos, ligados através de uma AGM, formam uma solução viável. A adequação do cromossomo é calculada somando-se os custos das arestas de ligação entre os pontos da árvore com o valor dos prêmios de cada ponto fora da árvore.

A Figura 5.1 ilustra um cromossomo que representa uma solução para o grafo mostrado na mesma figura. No grafo, as arestas tracejadas não fazem parte da solução. Os nós destacados em cinza têm seu valor de demanda posicionado ao lado. A solução é representada pelo cromossomo de oito posições, onde cada posição representa um nó do grafo. Fazem parte da solução apresentada no exemplo apenas os nós de números 0, 1, 3, 4 e 7. A adequação do cromossomo é calculada através da soma do custo das arestas da solução com a soma dos prêmios dos nós que não estão na solução.

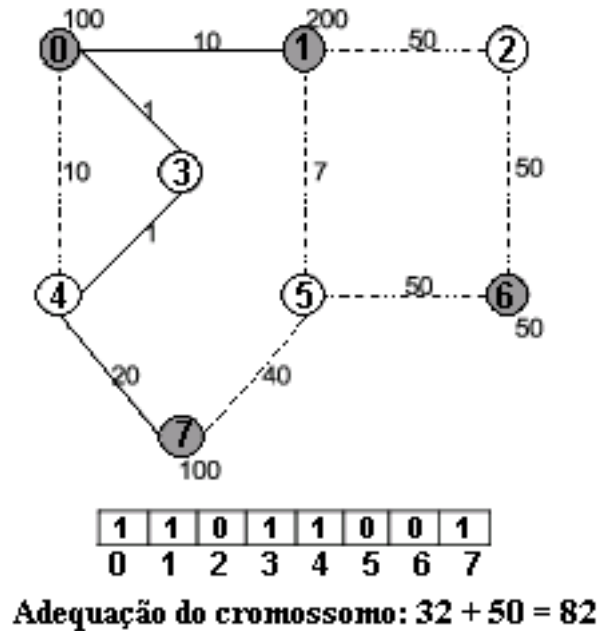


Figura 5.1: Exemplo de representação através do cromossomo.

A sequência de passos está detalhada no Algoritmo 3. Mimetizando o processo biológico, o algoritmo transgenético emprega mais intensamente a informação obtida no contexto extracelular no início da evolução. Na medida em que a população de cromossomos evolui, a informação de origem endógena passa a ser privilegiada.

No passo 1 é organizado o banco de informações que será utilizado pelos plasmídios. O banco contém, a princípio, informações obtidas a priori. Posteriormente serão incorporadas informações alcançadas durante o processo evolucionário. As informações obtidas a priori formam uma matriz que contém os caminhos mais curtos entre todos os nós do grafo de entrada. As informações do processo evolucionário são representadas por trechos dos melhores cromossomos da população, armazenados em um *pool*.

No passo 2 é gerada uma população inicial de cromossomos através de uma implementação do algoritmo primal-dual de Johnson et al. [JMP00] com perturbações, conforme feito em [BF95]. Este algoritmo determina um conjunto de nós que devem compor uma solução. Em seguida é construída a AGM sobre este conjunto de nós, formando uma solução viável. É aplicada então uma poda sobre esta árvore para que não sejam mantidos na solução nós folha que tenham prêmio menor do que o custo da aresta que o conecta à árvore.

No passo 3, uma variável armazena a melhor solução da população, que será atualizada no decorrer do algoritmo. O parâmetro β , juntamente com a variável u , vai definir a probabilidade de ataque dos vetores transgenéticos. Ele recebe o valor 1 no passo 4 e é incrementado a cada geração, passo 30.

Algoritmo 3: Algoritmo transgenético híbrido.

Entrada: Grafo G

Saída: Cromossomo C^*

```

1 Carregar o banco de informações genéticas -  $BIG$ 
2 Gerar População de cromossomos -  $P$ 
3  $C^* \leftarrow$  melhor cromossomo de  $P$ 
4  $\beta \leftarrow 1$ 
5 enquanto  $\beta \leq ngeracoes$  faça
6   para  $j \leftarrow 1$  até  $tampop$  faça
7      $u \leftarrow rand()\%ngeracoes$ 
8     se  $u > \beta$  então
9        $C' \leftarrow Ataque\_Plasmidio(C_j, \beta)$ 
10    senão
11       $C' \leftarrow Ataque\_Transposon(C_j)$ 
12    fim
13     $C' \leftarrow PathRelinking(C')$ 
14    se  $C' \in P$  então
15       $C' \leftarrow Perturba(C')$ ;
16    fim
17    se  $C'$  é melhor que  $C^*$  então
18       $C^* \leftarrow C'$ 
19       $AtualizaPoolElite(C^*)$ 
20    fim
21    se  $C'$  melhor que  $C_j$  então
22       $C_j \leftarrow C'$ 
23    senão
24       $x \leftarrow rand()\%100$ 
25      se  $x \leq 5$  então
26         $C_j \leftarrow C'$ ;
27      fim
28    fim
29  fim
30   $\beta ++$ 
31 fim

```

Enquanto o critério de parada não for atendido, o algoritmo realiza as etapas de iteração entre as populações de vetores e a população de cromossomos. A cada iteração, um tipo de vetor, plasmídeo ou transposon, é selecionado. Como a metáfora sugere que

a infiltração de informações extracelulares tende a ser mais útil nas etapas iniciais de um processo evolucionário, a probabilidade de escolha do tipo de vetor varia ao longo das iterações. Na fase inicial do processo o fluxo de informações extracelular é privilegiado, de modo que o ataque via plasmídios possui maior probabilidade de ocorrência. Isso ocorre porque o valor de u , gerado aleatoriamente, tem grande probabilidade de ser maior do que β no início da evolução. Ao final do processo, a probabilidade de atuação é maior para os transposons, pois o valor de β será maior, e a probabilidade de u ser menor do que β cresce.

A geração de números aleatórios é feita através da função *rand()* da linguagem C. A função gera números pseudo-aleatórios usando um valor semente (*seed*). A semente tem o valor inicial igual a um, mas este valor pode ser modificado através da função *srand()*. Neste algoritmo, o tempo do computador foi utilizado como semente, para que números diferentes sejam gerados a cada chamada à função.

Após o ataque com os vetores da transgenética, a solução resultante é submetida a um procedimento de *path-relinking* [CRR01] para intensificação, passo 13. É escolhido do *pool* de soluções elite um cromossomo X que seja mais próximo do cromossomo C que está sendo manipulado. São realizados todos os movimentos possíveis da solução C em direção à solução X , sem violar a viabilidade da solução. A melhor solução gerada nesta série de movimentos é retornada.

Se o processo de manipulação gerou um cromossomo que já existe na população é realizada uma perturbação para manter uma certa diversidade na população, passo 15. Caso o novo indivíduo represente uma solução melhor que todas as encontradas até o momento, dito cromossomo campeão, a melhor solução da população e o *pool* de soluções elite são atualizados, passos 18 e 19. Se o cromossomo resultante deste processo é melhor que o original em termos de adequação, então o novo indivíduo substitui o antigo na população, passo 22. Caso contrário, o indivíduo só substituirá o antigo com 5% de probabilidade, passo 26. Essa possibilidade está presente pois uma piora momentânea pode acarretar em uma melhoria futura.

As subseções a seguir detalham cada procedimento citado acima, que em conjunto formam o algoritmo proposto.

5.1 Banco de informações genéticas

O banco de informações genéticas é representado por uma matriz $n \times n$, onde n é o número de nós do grafo. Esta matriz é composta pelo caminho mais curto entre todos os pares de nós do grafo, calculado através do algoritmo de Dijkstra [Dij59] que tem complexidade $O(n^2)$. Esta matriz é conhecida como rede de distâncias, e é definida como um grafo completo não direcionado cujo custo das arestas é dado pelo caminho mais curto entre os pares de nós.

Esta matriz é utilizada como uma fonte de informações extracelulares no algoritmo, obtidas "a priori" do processo de evolução. Quando é necessário adicionar um nó a uma solução, é resgatado do banco de informações genéticas o caminho mais curto que liga aquele nó à árvore solução.

Cada caminho possui dois atributos que informam o seu custo e o seu lucro. O custo nada mais é do que a soma dos pesos das arestas contidas no caminho, e o lucro é a soma dos pesos das arestas menos a soma dos prêmios dos nós de demanda contidos no caminho.

Como exemplo pode-se considerar uma matriz *BIG* onde o elemento $BIG[2][30]$ é o caminho mais curto entre o nó 2 e o nó 30. O caminho é composto pelos nós 5, 10 e 103. Suponha que cada aresta tem custo 10, e apenas um nó deste conjunto contém demanda maior do que zero, de valor 40. Portanto o valor do custo é $(10+10+10) = 30$, e o valor do lucro é $(10+10+10-40) = -10$. Quanto menor for o valor do lucro, mais lucrativo é o caminho, pois o valor dos prêmios contidos no caminho são subtraídos da soma do custo das aretas.

São utilizadas também informações obtidas no decorrer da evolução. Estas informações são trechos extraídos dos melhores cromossomos gerados durante a evolução do algoritmo, que estão armazenados em um *pool* de soluções elite. Estes trechos são arborecências de tamanho variável, sorteado no momento da infiltração.

5.2 População inicial

Para obter soluções iniciais viáveis e de boa qualidade foi feita uma implementação do algoritmo primal-dual $O(n^2 \log n)$ de Johnson et al. [JMP00]. O algoritmo está ex-

plicado detalhadamente na subseção 3.5.3. O algoritmo primal-dual retorna um conjunto X de nós. Seja $G'(X)$ o subgrafo de G de onde foram eliminados todos os nós que não pertencem a X e todas as arestas em que ao menos uma das extremidades não está em X . $G'(X)$ é chamado de subgrafo de G induzido por X . Se $G'(X)$ é um grafo conexo, é possível construir uma AGM de $G'(X)$. Caso contrário, não será possível a construção da AGM e o conjunto X deve ser desconsiderado.

A estratégia para formar soluções iniciais viáveis consiste em, dado o conjunto X , construir a AGM do grafo induzido por X e reduzi-la através de um procedimento de poda, para que não sejam conectados à solução nós cujo prêmio é menor do que o custo das arestas que o conectam.

Para a construção da AGM foi implementado um algoritmo de Prim [Pri57] com complexidade $O(n^2)$. A poda é realizada retirando-se da árvore todas as arestas onde um dos nós de extremidade possui grau um e prêmio menor que o custo da aresta. Este procedimento tem complexidade $O(n^2)$ no pior caso, e está detalhado no Algoritmo 4. O algoritmo para geração da população inicial tem complexidade total de $O(n^2)$.

Algoritmo 4: Algoritmo de poda.

Entrada: Árvore T

Saída: Árvore reduzida T'

```

1 reduz ← 1
2 enquanto reduz faça
3   reduz ← 0
4   para (cada aresta  $e \in T$ ) faça
5     se ( $\min(\text{grau}(u), \text{grau}(v)) = 1$ ) então
6        $a \leftarrow \text{argmin}(\text{grau}(u), \text{grau}(v))$ 
7        $b \leftarrow \text{argmax}(\text{grau}(u), \text{grau}(v))$ 
8       se  $\pi_a < c_e$  então
9          $T' \leftarrow T' - a$ 
10         $T' \leftarrow T' - e$ 
11        reduz ← 1
12      fim
13    fim
14  fim
15 fim

```

Para que as soluções iniciais formem uma população diversificada, foram utilizadas perturbações, como sugerido em [CRR01]. O algoritmo utilizado para a geração da população inicial é descrito no Algoritmo 5. As perturbações são de dois tipos:

- Perturbação por eliminação: para forçar a diversificação, o algoritmo é induzido a construir novas soluções sem alguns dos nós que apareciam na solução obtida na iteração anterior. Ou seja, com 50% de chance, cada nó de demanda que aparecia na solução da iteração anterior tem seu prêmio trocado para zero.
- Perturbação por troca de prêmios: são introduzidos pequenos ruídos nos prêmios. Para cada nó i é gerado um fator α no intervalo $[0,1]$, e o prêmio associado ao nó i é substituído por $\pi_i = \pi_i \times \alpha$.

Algoritmo 5: Algoritmo de geração da população inicial.

Entrada: Grafo G

Saída: População de cromossomos

```

1 para  $i = 0; i < tam\_pop; i++$  faça
2   População[i] = PrimalDual()
3   População[i] = AGM(População[i])
4   AddElite(Populacao[i])
5   PerturbaCustos()
6 fim
```

5.3 Vetores transgenéticos

5.3.1 O vetor plasmídio

Este vetor atua utilizando informações do contexto extracelular e informações obtidas durante a evolução. Como sugere a metáfora, no início da evolução é privilegiado o uso de informações extracelulares, sendo que com apenas 20% de probabilidade serão utilizadas informações obtidas durante a evolução. Quando a evolução estiver em uma etapa avançada, os papéis se invertem, sendo que as informações obtidas durante a evolução têm 80% de probabilidade de atuarem.

Existem dois repositórios de informações, uma matriz BIG contendo informações obtidas "a priori", e um *pool* de soluções elite contendo informações obtidas durante a evolução.

As informações obtidas "a priori" são introduzidas da seguinte maneira. Seja C um cromossomo a ser manipulado. Este cromossomo será atacado por k plasmídios, e k será definido da seguinte maneira. É sorteado um intervalo de no máximo n posições. Dentro

desse intervalo, no cromossomo, existirão k nós de demanda fora da solução representada pelo mesmo. Estes k nós de demanda serão infiltrados no cromossomo através de plasmídios que fazem a ligação do nó que está sendo inserido até a árvore solução. Cada plasmídio é obtido do banco de informações genéticas, onde $BIG[i][j]$ contém o caminho que liga o nó i ao nó j .

Os plasmídios são infiltrados através da maneira que retorna a melhor adequação ao cromossomo: através do caminho mais curto (através do atributo custo) de um determinado nó de demanda até a solução, ou através do caminho mais lucrativo (através do atributo lucro).

A Figura 5.2 ilustra um exemplo de competição entre dois plasmídios do contexto extracelular. Suponha que se deseja inserir o nó 6 à solução. Consultando o banco de informações genéticas, pode-se verificar que o caminho mais curto de 6 para a árvore solução é através do plasmídio que liga o nó 6 ao nó 1 passando pelo nó 5, e tem custo $= (7 + 50) = 57$ e lucro $= (50 + 7 - 200 - 200) = -343$ (caminho em azul). Porém o caminho mais lucrativo seria através do plasmídio que liga o nó 6 ao nó 1, passando pelo nó 2, onde o custo $= (30 + 50) = 80$ e lucro $= (30 + 50 - 200 - 200 - 50) = -370$ (caminho em vermelho).

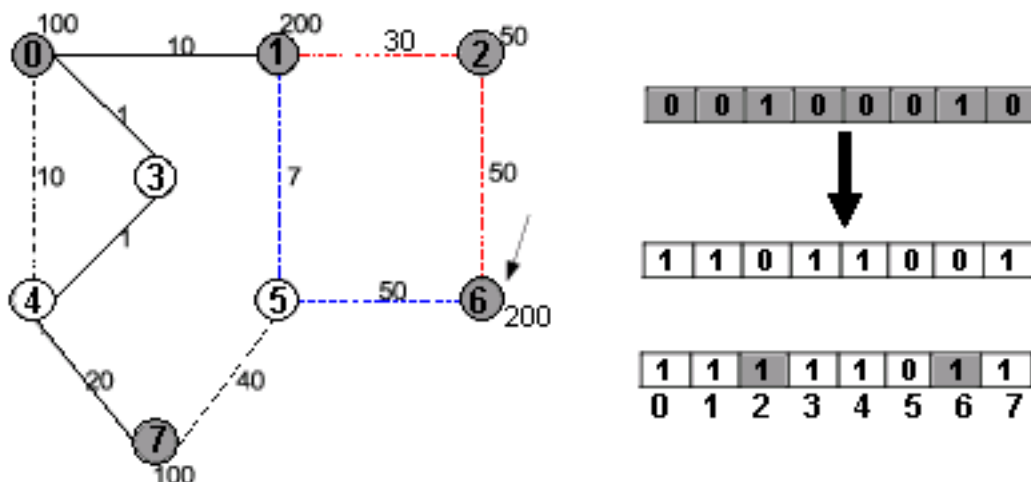


Figura 5.2: Dois plasmídios competindo.

Ganha a competição o plasmídio que retornar a melhor adequação da função objetivo. No exemplo, se o caminho azul for inserido, a função objetivo do cromossomo seria $(20 + 1 + 1 + 10 + 7 + 50 + 50) = 139$. Se o caminho em vermelho for inserido, a

função objetivo seria $(20 + 1 + 1 + 10 + 30 + 50) = 112$. Portanto, como o objetivo é a minimização, ganha o plasmídio que forma o caminho em vermelho.

O *pool* de soluções elite é o repositório de informações obtidas durante o processo evolucionário. Quando estas informações forem necessárias será sorteado um cromossomo deste repositório e um trecho será definido aleatoriamente. A sequência de nós contidas neste trecho, formando uma arborescência, será infiltrada no cromossomo.

No Algoritmo 6, os passos de 4 a 10 e 17 a 23 , mostram o processo de infiltração de informações do contexto extracelular. Os passos 11 a 15 e 24 a 28 correspondem à infiltração de informações obtidas no decorrer da evolução.

A complexidade de inserção de informações do *BIG* é de $O(n^2)$. A inserção se um trecho do *pool* de soluções elite também tem complexidade $O(n^2)$. Portanto, a complexidade deste algoritmo é de $O(n^2)$.

Algoritmo 6: Ataque_Plasmidio(C, β)**Entrada:** Cromossomo C , inteiro β **Saída:** Cromossomo manipulado C

```

1  $u \leftarrow rand()\%n_{geracoes}$ 
2  $x \leftarrow rand()\%100$ 
3 se  $u > \beta$  então
4   se  $x < 80$  então
5      $v_1 \leftarrow rand()\%n; v_2 \leftarrow rand()\%n$ 
6     para  $i \leftarrow v_1$  até  $v_2$  faça
7       se  $\pi_i > 0$  and  $i \notin C$  então
8          $C \leftarrow C + BIG[i][w]$ 
9       fim
10    fim
11  senão
12     $x \leftarrow rand()\%tampool$ 
13     $C \leftarrow C + trecho(Pool[x])$ 
14     $C \leftarrow AGM(C)$ 
15  fim
16 senão
17   se  $x < 20$  então
18      $v_1 \leftarrow rand()\%n; v_2 \leftarrow rand()\%n$ 
19     para  $i \leftarrow v_1$  até  $v_2$  faça
20       se  $\pi_i > 0$  and  $i \notin C$  então
21          $C \leftarrow C + BIG[i][w]$ 
22       fim
23     fim
24   senão
25      $x \leftarrow rand()\%tampool$ 
26      $C \leftarrow C + trecho(Pool[x])$ 
27      $C \leftarrow AGM(C)$ 
28   fim
29 fim

```

5.3.2 O vetor transposon

O vetor transposon implementado neste trabalho tem como objetivo aperfeiçoar os cromossomos da população de maneira gulosa, através das regras para manipulação do código genético que ele carrega.

São escolhidos aleatoriamente dois nós que formam um intervalo dentro do conjunto de nós do grafo. Dentro deste intervalo o algoritmo vai verificar quais nós pertencem ao cromossomo que está sendo manipulado. Através da remoção de nós da solução dentro do

intervalo estipulado são feitas tentativas de melhora, construindo-se a AGM do conjunto atual de nós da solução seguida pelo procedimento de poda. A solução nunca perderá a viabilidade, pois toda a manipulação é feita através do algoritmo de AGM.

A Figura 5.3 mostra um exemplo de manipulação. Suponha que o intervalo sorteado para manipulação seja o marcado em cinza no cromossomo da Figura 5.3(a). O algoritmo irá excluir cada um dos nós do intervalo e construir a AGM para o grafo induzido pelo conjunto de nós corrente, e realizar uma poda.

No exemplo, o valor da função objetivo da solução de entrada (Figura 5.3(a)) é $(20 + 1 + 1 + 10 + 7 + 50 + 100) = 189$. Há uma melhora ao se excluir o nó 5 da solução, tornando o valor da função objetivo igual a $(20 + 1 + 1 + 10 + 50 + 100) = 182$. Nota-se que o nó 6 não faz parte do intervalo de manipulação, porém, com a retirada do nó 5 não é possível alcançar o nó 6 (isso só seria possível através dos nós 2 e 5, que não fazem parte da solução). Portanto, o algoritmo retorna a manipulação realizada, representada na Figura 5.3(b).

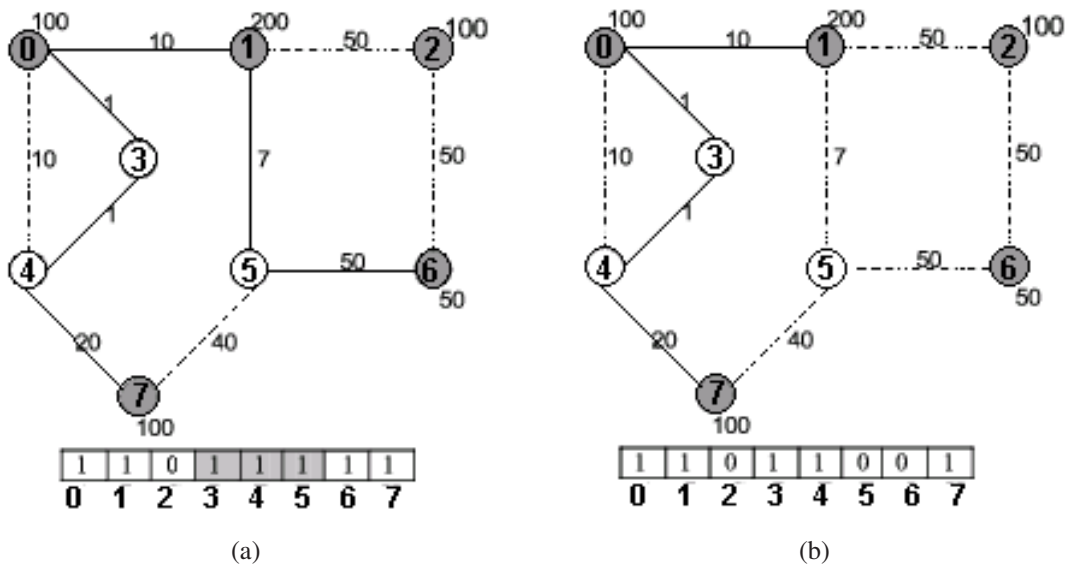


Figura 5.3: Cromossomo sendo manipulado pelo transposon.

O vetor transposon não permite nenhuma piora na função objetivo do cromossomo. A sua função é apenas fazer tentativas de melhorar a adequação dos indivíduos da população. O Algoritmo 7 mostra a sequência de passos do transposon. A complexidade deste algoritmo é de $O(n^3)$, pois o procedimento de AGM será executado n vezes, conforme o intervalo sorteado.

Algoritmo 7: Ataque_Transposon(C)**Entrada:** Cromossomo C **Saída:** Cromossomo manipulado C

```

1  $v_1 \leftarrow rand()\%n$ 
2  $v_2 \leftarrow rand()\%n$ 
3 para  $i \leftarrow v_1$  até  $v_2$  faça
4    $C' \leftarrow C - i$ 
5    $C' \leftarrow AGM(C')$ 
6   se  $C'$  melhor que  $C$  então
7      $C \leftarrow C'$ 
8   fim
9 fim

```

5.4 Path – relinking

Path-relinking [Glo96] é uma técnica que gera novas soluções através da exploração de trajetórias que conectam soluções de um conjunto de elite. O *path-relinking* foi adaptado para ser usado juntamente com procedimentos *GRASP* por Laguna e Marti [LM99], e uma proposta semelhante foi utilizada por Canuto et al. em [CRR01].

No presente trabalho, a idéia foi adaptada para o uso conjunto com o algoritmo transgenético. O conjunto de soluções elite é formado por um *pool* com as melhores soluções encontradas até então. Para fazer a conexão de caminhos, é selecionada a solução mais próxima do cromossomo em questão, através da distância de Hamming. A distância é calculada através da subtração do número de nós que as duas tem em comum ($|V_X \cap V_{X'}|$) do número de nós da união das duas soluções ($|V_X \cup V_{X'}|$).

$$dist(X, X') = |V_X \cup V_{X'}| - |V_X \cap V_{X'}| \quad (5.1)$$

Por exemplo, a Figura 5.3 representa duas soluções diferentes, e a distância entre elas é calculada da seguinte maneira. O número da união de nós das duas soluções é 7, e o número de nós que elas tem em comum é 5. Portanto a distância entre elas é $7 - 5 = 2$.

O Algoritmo 8 mostra a sequência de passos do *path-relinking*. Cada iteração do algoritmo transgenético produz um cromossomo C através do ataque de um agente, e então é escolhido do *pool* de soluções elite o cromossomo X mais próximo de C através da equação 5.1 (passo 1).

Algoritmo 8: *Path – relinking***Entrada:** Cromossomo C **Saída:** Cromossomo manipulado C

```

1 Encontrar solução elite  $X$  mais próxima de  $C$ 
2 para  $i \leftarrow 1$  até  $n$  faça
3   se  $C[i] \neq X[i]$  então
4     se  $C[i] = 1$  então
5        $M[i] \leftarrow 1$ 
6     senão
7        $M[i] \leftarrow 2$ 
8     fim
9   senão
10     $M[i] \leftarrow 0$ 
11  fim
12 fim
13 para  $i \leftarrow 1$  até  $n$  faça
14   se  $M[i] \neq 0$  então
15     se  $M[i] = 1$  então
16        $C'[i] \leftarrow 0$ 
17     senão
18        $C'[i] \leftarrow 1$ 
19     fim
20      $C' \leftarrow \text{AGM}(C')$ 
21   fim
22   se  $C'$  melhor que  $C$  então
23      $C \leftarrow C'$ 
24   fim
25 fim

```

Nos passos 2 a 12 é calculado o vetor M , que contém todos os movimentos possíveis de C para X . Se $M[i] = 1$ significa que o nó i está em C mas não está em X , portanto deve ser excluído de C . Se $M[i] = 2$ significa que i não está em C mas está em X , portanto deve ser inserido em C .

Nos passos 13 a 24 são realizados todos os movimentos possíveis da solução C em direção à solução X , sem violar a viabilidade da solução. A transformação consiste em inserir ou retirar nós da solução C conforme os movimentos de M e, em seguida, construir a AGM do grafo induzido pelo conjunto de nós da solução, de modo que C se torne, a cada movimento, mais parecida com X .

A melhor solução gerada nesta série de movimentos é retornada. A complexidade deste algoritmo é $O(n^3)$, pois o procedimento de AGM, que tem complexidade $O(n^2)$ será

executado n vezes, onde n depende do tamanho do intervalo sorteado para a manipulação.

5.5 Perturbação

Para evitar que soluções repetidas sejam mantidas na população, é feita uma perturbação nos cromossomos repetidos. A cada ataque com os vetores da transgenética e com o *path-relinking*, é feita uma verificação na população da existência do cromossomo gerado. Isto é feito através de uma varredura na população de maneira a encontrar um indivíduo que tenha o mesmo valor da função objetivo e o mesmo número de nós em sua solução.

Se o cromossomo já existe, um nó de demanda é escolhido aleatoriamente e é adicionado à solução através do caminho mais curto. Não é verificado neste caso se a inserção resultou em uma piora na solução ou não, o objetivo é apenas a diversificação.

5.6 Critérios de parada

O principal critério de parada utilizado neste algoritmo é um número máximo de iterações. O número máximo de iterações é limitado pelo número de gerações do algoritmo. Porém, o algoritmo pode não atingir o número de gerações estabelecido se um determinado número de iterações sem melhora for atingido. Outro critério de parada utilizado é o alcance do valor da solução ótima para a instância que está sendo resolvida.

Portanto, são três os possíveis critérios de parada. O algoritmo pára quando é atingido o número de gerações, quando é atingido um certo número de iterações sem melhora, ou quando o valor da solução ótima é encontrado.

Capítulo 6

Resultados computacionais

6.1 Definição de parâmetros

Para definir o valor dos parâmetros necessários no algoritmo transgenético híbrido, foram realizados alguns experimentos preliminares. Foram escolhidas algumas instâncias para as quais os três algoritmos, GRASP, AMH e ATH, não encontram a solução ótima com facilidade, e foram realizadas 10 execuções das mesmas.

Foram considerados os seguintes parâmetros: tamanho da população (*tampop*), número de gerações (*n_geracoes*), tamanho do *pool* de soluções elite (*tampool*), e probabilidade de aceitação de soluções que pioraram com o ataque dos vetores transgenéticos. As tabelas que seguem resumem os resultados obtidos para as respectivas configurações de parâmetros.

- *tampop* = 20; *n_geracoes*=40; *tampool*=3; *probabilidade*=3.

Tabela 6.1: Resultados obtidos para a primeira configuração de parâmetros

Instância	ATH
k400.2	0,2
p200	0
p400	0,24
c3-b	0
c4-b	0,47
d10-a	0,23
d13-b	0,61

- $tampop = 25$; $n_geracoes=50$; $tampool=3$; $probabilidade=3$.

Tabela 6.2: Resultados obtidos para a segunda configuração de parâmetros

Instância	ATH
k400.2	0,2
p200	0
p400	0
c3-b	0,14
c4-b	0,18
d10-a	0,23
d13-b	0,61

- $tampop = 25$; $n_geracoes=60$; $tampool=5$; $probabilidade=5$.

Tabela 6.3: Resultados obtidos para a terceira configuração de parâmetros

Instância	ATH
k400.2	0
p200	0
p400	0,2
c3-b	0
c4-b	0,09
d10-a	0,11
d13-b	0,2

- $tampop = 20$; $n_geracoes=60$; $tampool=3$; $probabilidade=5$.

Tabela 6.4: Resultados obtidos para a quarta configuração de parâmetros

Instância	ATH
k400.2	0,2
p200	0
p400	0,3
c3-b	0
c4-b	0,18
d10-a	0,17
d13-b	0,41

- $tampop = 25$; $n_geracoes=60$; $tampool=10$; $probabilidade=10$.

Tabela 6.5: Resultados obtidos para a quinta configuração de parâmetros

Instância	ATH
k400.2	0
p200	0
p400	0,24
c3-b	0
c4-b	0,47
d10-a	0,29
d13-b	0,61

- $tampop = 35$; $n_geracoes=80$; $tampool=5$; $probabilidade=5$.

Tabela 6.6: Resultados obtidos para a sexta configuração de parâmetros

Instância	ATH
k400.2	0
p200	0
p400	0,23
c3-b	0
c4-b	0,37
d10-a	0,29
d13-b	0,41

Através destes experimentos, concluiu-se que os parâmetros da Tabela 6.3 formam uma configuração estável:

- tamanho da população $tampop = 25$;
- número de gerações $n_geracoes = 60$;
- tamanho do *pool* de soluções elite $tampool = 5$;
- probabilidade de aceitação de uma solução que piorou depois do ataque do algoritmo é de 5%;

Além destes parâmetros, também são considerados o critério de parada e o parâmetro β , que regula a probabilidade de ataque dos vetores transgenéticos. Foi adotado um

critério de parada por número de iterações sem melhoria. O total de iterações do algoritmo é de 1500 iterações (60 gerações * 25 indivíduos), porém, com um total de 1200 iterações sem melhoria, o algoritmo transgenético pára. O parâmetro β é iniciado com 1 e é incrementado a cada geração. Sendo assim, a probabilidade de ataque dos vetores é uniforme.

6.2 Resultados obtidos

O algoritmo foi testado para um conjunto de 114 instâncias (grupos K, P, C e D) propostas na literatura e explicadas na subseção 3.6. Foi utilizada a linguagem C e o compilador gcc, sob o sistema operacional Linux Kubuntu 6.06. O computador utilizado para os testes tem a seguinte configuração: processador Pentium IV com 2.8 GHz, 256 MB de RAM.

O algoritmo foi comparado com o algoritmo *GRASP* híbrido de Canuto et al. [CRR01] e com o algoritmo memético híbrido de Klau et al. [KLM⁺04]. Os computadores utilizados em seus experimentos têm as seguintes configurações, respectivamente: Pentium II com 400 MHz, e 64 MB de RAM; Pentium IV com 2.8 GHz, e 2 GB de RAM.

Por causa da natureza estocástica de algoritmos como este, foram realizadas 50 execuções do algoritmo para cada instância. Os resultados dos algoritmos comparados foram transcritos dos respectivos trabalhos publicados.

As Tabelas 6.7, 6.8 e 6.9 mostram dados estatísticos das 50 execuções do algoritmo proposto realizadas no presente experimento. Na primeira coluna está o nome da instância. Na segunda coluna está o valor da solução ótima para a mesma, obtido pelos métodos de Lucena e Resende [LR00] e de Ljubic [Lju04b]. Na sequência tem-se a média, a mediana e o desvio padrão.

Pode-se notar que para as instâncias da série P e para algumas da série K o algoritmo apresentou um elevado desvio padrão. Porém, na maioria dos casos a mediana é igual ou próxima ao valor da solução ótima. Para a série C, 29 das 40 instâncias apresentaram desvio padrão menor do que 1. Para a última série de instâncias, 21 das 40 instâncias apresentaram desvio padrão menor do que 1.

Tabela 6.7: Média, mediana e desvio padrão para as séries K e P.

Instância	Ótimo	Média	Mediana	Desvio Padrão
K100	135511	135511	135511	0
K100.1	124108	124108	124108	0
K100.2	200262	200300,8	200262	245,7
K100.3	115953	115953	115953	0
K100.4	87498	87498	87498	0
K100.5	119078	119078	119078	0
K100.6	132886	132886	132886	0
K100.7	172457	172457	172457	0
K100.8	210869	212903,4	210869	2392,2
K100.9	122917	122917	122917	0
K100.10	133567	133567	133567	0
K200	329211	329211	329211	0
K400	350093	350203,76	350093	463,8
K400.1	490771	491181,5	490819	664,8
K400.2	477073	478755,4	478129	1139,8
K400.3	415328	415328	415328	0
K400.4	389451	394123,1	393112	3154,6
K400.5	519526	521263,7	521081,5	1991,8
K400.6	374849	374849	374849	0
K400.7	474466	477891,1	477469	2187,4
K400.8	418614	418614	418614	0
K400.9	383105	385657	384978	1637,7
K400.10	394191	397410,6	397238	2092,6
P100	803300	803529,4	803300	917,3
P100.1	926238	927129,8	926238	1016,3
P100.2	401641	401881,4	401641	323,8
P100.3	659644	660437	659644	2281,9
P100.4	827419	827826,3	827419	1941,8
P200	1317874	1319466,3	1317874	2609,1
P400	2459904	2470578,4	2470915	5346,7
P400.1	2808440	2812910	2811541	5006,1
P400.2	2518577	2519380,3	2518993	1624,8
P400.3	2951725	2954195,6	2952966	2730,5
P400.4	2852956	2875241,3	2872186	11887,5

Os resultados comparativos são mostrados nas Tabelas 6.10, 6.11 e 6.12. Na primeira coluna está o nome da instância. As três colunas seguintes identificam os algoritmos a serem comparados. Estas colunas são subdivididas em duas colunas. A coluna %gap descreve o afastamento percentual entre a solução obtida e o valor da solução ótima. A coluna t(s) relata o tempo médio em segundos de processamento nas instâncias exami-

Tabela 6.8: Média, mediana e desvio padrão para a série C.

Instância	Ótimo	Média	Mediana	Desvio Padrão
C01-A	18	18	18	0
C01-B	85	85,1	85	0,5
C02-A	50	50	50	0
C02-B	141	141	141	0
C03-A	414	414,9	414	1,9
C03-B	737	737,5	737	2,4
C04-A	618	619,4	619	1,3
C04-B	1063	1069,4	1068	2,6
C05-A	1080	1084,4	1084,5	2,8
C05-B	1528	1528	1528	0
C06-A	18	18	18	0
C06-B	55	58,5	58	2,9
C07-A	50	50	50	0
C07-B	102	108,4	108	3,4
C08-A	361	363,8	363,5	2,6
C08-B	500	501,6	502	1,3
C09-A	533	537,1	537	2,1
C09-B	694	694,86	695	0,9
C10-A	859	860,7	861	0,9
C10-B	1069	1069,1	1069	0,4
C11-A	18	18	18	0
C11-B	32	32,2	32	0,7
C12-A	38	38,02	38	0,1
C12-B	46	46,2	46	0,4
C13-A	236	241,6	242	2,3
C13-B	258	206,2	261	0,9
C14-A	293	293,6	294	0,6
C14-B	318	318,5	319	0,5
C15-A	501	501,2	501	0,4
C15-B	551	551	551	0
C16-A	11	11,9	12	0,3
C16-B	11	11,9	12	0,2
C17-A	18	18,3	18	0,4
C17-B	18	18,3	18	0,4
C18-A	111	112,7	113	0,6
C18-B	113	114,9	115	0,6
C19-A	146	146,9	147	0,4
C19-B	146	146,9	147	0,4
C20-A	266	266	266	0
C20-B	267	267,02	267	0,1

Tabela 6.9: Média, mediana e desvio padrão para a série D.

Instância	Ótimo	Média	Mediana	Desvio Padrão
D01-A	18	18	18	0
D01-B	106	106,3	106	0,7
D02-A	50	50	50	0
D02-B	218	225,9	226	5,4
D03-A	807	810,5	810	2,4
D03-B	1509	1523,1	1522	3,1
D04-A	1203	1222,8	1225,5	6,9
D04-B	1881	1888,4	1888	1,8
D05-A	2157	2171,3	2172	3,4
D05-B	3135	3139,3	3139	0,7
D06-A	18	18	18	0
D06-B	67	69,5	68	2,4
D07-A	50	50	50	0
D07-B	103	103,8	103	1
D08-A	755	760,2	760,5	1,6
D08-B	1036	1041,8	1041	2,5
D09-A	1070	1079,4	1080	2,3
D09-B	1420	1423,2	1424	1
D10-A	1671	1677,8	1677	2
D10-B	2079	2083,1	2083,5	1,4
D11-A	18	18	18	0
D11-B	29	30,2	30	0,4
D12-A	42	42	42	0
D12-B	42	42	42	0
D13-A	445	446,6	447	0,8
D13-B	486	489,8	490	1
D14-A	602	603,8	604	0,8
D14-B	665	667,4	668	1,6
D15-A	1042	1042,1	1042	0,3
D15-B	1108	1108,8	1109	0,4
D16-A	13	13	13	0
D16-B	13	13	13	0
D17-A	23	23	23	0
D17-B	23	23,03	23	0,1
D18-A	218	222,1	222	1,2
D18-B	223	227,5	228	0,6
D19-A	306	311,9	312	1,1
D19-B	310	314,6	315	1,1
D20-A	536	536,03	536	0,1
D20-B	537	537,1	537	0,3

nadas. O %gap é calculado através da seguinte fórmula:

$$\%gap = \frac{v(T) - OPT}{OPT} \times 100\% \quad (6.1)$$

onde $v(T)$ é o valor da solução encontrada pelo algoritmo, e OPT é o valor da solução ótima da instância.

A Tabela 6.13 apresenta um resumo conforme a classe e o tamanho das instâncias, formando assim 7 classes. Os tempos computacionais entre o *GRASPH* e os demais algoritmos não pode ser diretamente comparado, devido a diferença entre as plataformas utilizadas nos experimentos. Entretanto, o AMH e o ATH são executados em plataformas semelhantes. O ATH exibe melhor tempo médio para 6 das 7 séries de instâncias. O fato de o AMH exibir melhor tempo para a última série de instâncias pode ser atribuído ao pré-processamento realizado pelo algoritmo, reduzindo consideravelmente os grafos de entrada, e diminuindo assim o tempo para resolvê-los.

O ATH exibe melhores resultados que o AMH em 5 das 7 classes, e resultados equivalentes em 2 classes. O ATH supera o *GRASPH* em 2 classes, e é superado pelo mesmo também em 2 classes. Para as demais classes os resultados são equivalentes. Pode-se concluir que os algoritmos *GRASPH* e ATH são igualmente eficientes, e superam o AMH no que diz respeito a qualidade das soluções.

Tabela 6.10: Resultados obtidos para as séries K e P.

Instância	<i>GRASPH</i>		<i>AMH</i>		<i>ATH</i>	
	%gap	t(s)	%gap	t(s)	%gap	t(s)
K100	0	2	0	1,7	0	0
K100.1	0	2	0	1,6	0	0
K100.2	0	3	0	1,6	0	0
K100.3	0	3	0	1,5	0	0
K100.4	0	6	0	1,9	0	0
K100.5	0	2	0	1,4	0	0
K100.6	0	2	0	1,2	0	0
K100.7	0	2	0	1,6	0	0
K100.8	0	2	2,3	1,5	0	0
K100.9	0	2	0	1,4	0	0
K100.10	0	1	0	1,4	0	0
K200	0	9	0	2,4	0	0
K400	0	68	0	6,9	0	0
K400.1	0	194	0	6,7	0	0
K400.2	0,2	234	0,2	7,1	0	1,5
K400.3	0	140	0	7,4	0	0
K400.4	0	204	0,1	7,7	0	0,8
K400.5	0	122	0,3	7,1	0	0,2
K400.6	0	60	0	8,3	0	0,1
K400.7	0,1	306	0,1	7,7	0	0,9
K400.8	0	42	0	6,7	0	0
K400.9	0	76	0,1	7,5	0	1
K400.10	0,4	231	0,7	8,7	0,2	0,4
P100	0	15	0	2,7	0	0
P100.1	0	14	0	3,3	0	0,01
P100.2	0	5	0	2,4	0	0,03
P100.3	0	10	0	2,8	0	0,03
P100.4	0	10	0	2,3	0	0,01
P200	0	72	0	7,2	0	0,7
P400	0	397	0,2	29,3	0,2	1,4
P400.1	0	382	0	19,3	0	1,3
P400.2	0	396	0	17,8	0	2,1
P400.3	0	500	0	23,1	0	11,6
P400.4	0	565	0,5	21,4	0,2	1,1

Foi solucionada aqui parte de um projeto de uma rede de distribuição de GN da cidade de Natal, cujo mapa foi mostrado na Figura 2.4. A Tabela 6.14 resume a estimativa de consumo, feita pela empresa distribuidora, para os cliente a serem atendidos. A coluna Consumidor indica o ponto a ser atendido, que pode ser um condomínio residencial, pré-

Tabela 6.11: Resultados obtidos para a série C.

Instância	<i>GRASPH</i>		<i>AMH</i>		<i>ATH</i>	
	%gap	t(s)	%gap	t(s)	%gap	t(s)
C01-A	0	3	0	1,9	0	0
C01-B	0	58	0	2,3	0	0
C02-A	0	7	0	1,8	0	0
C02-B	0	54	0	2,2	0	0
C03-A	0	87	0	2,9	0	0
C03-B	0	294	0	10	0	6
C04-A	0	148	0	4,5	0	7,6
C04-B	0	387	0,36	39	0,09	10,9
C05-A	0	447	0	9,4	0	10,6
C05-B	0	397	0	20,1	0	0
C06-A	0	9	0	4,1	0	0
C06-B	0	179	0	5,6	0	0,05
C07-A	0	34	0	3,5	0	0
C07-B	0,98	167	3,92	5,7	0,98	0,9
C08-A	0	313	0	8,5	0	0,5
C08-B	0	404	0,4	29	0	8
C09-A	0	475	0	13,4	0	27,4
C09-B	0	583	0,76	38,5	0	24,3
C10-A	0	628	0	35,4	0	90,3
C10-B	0	474	0	39,4	0	9,5
C11-A	0	128	0	6,1	0	0
C11-B	0	14	0	9,1	0	0,02
C12-A	0	162	0	9	0	0
C12-B	0	156	0	8,7	0	0
C13-A	0,42	1050	0,42	17,9	0,84	6,8
C13-B	0	733	0	35,9	0	2,8
C14-A	0	829	0	21	0	4,2
C14-B	0	766	0	29,8	0	2,3
C15-A	0	957	0	45,4	0	15,3
C15-B	0	837	0	45,7	0	0,01
C16-A	0	1920	9,09	10,6	0	0,7
C16-B	0	1758	9,09	11,5	0	0,9
C17-A	0	549	5,55	11,2	0	0,2
C17-B	0	434	0	12,7	0	0,3
C18-A	0	3990	0	24,1	0	9,5
C18-B	0	3262	0	26,2	0	16,7
C19-A	0	3928	0	17,9	0	7,3
C19-B	0	3390	2,05	15,8	0	12,3
C20-A	0	4311	0	7,3	0	8,1
C20-B	0	3800	0	5,2	0	7,4

Tabela 6.12: Resultados obtidos para a série D.

Instância	<i>GRASPH</i>		<i>AMH</i>		<i>ATH</i>	
	%gap	t(s)	%gap	t(s)	%gap	t(s)
D01-A	0	6	0	3,1	0	0
D01-B	0	257	0	3,8	0	0,6
D02-A	0	7	0	3,5	0	0
D02-B	4,58	486	0	7,3	0	7,1
D03-A	0	734	0	7,4	0	34,4
D03-B	0,06	2184	0,47	51	0,66	255,3
D04-A	0	1263	0	10,4	0,41	99,1
D04-B	0	2233	0,22	49,6	0,21	537,1
D05-A	0	3352	0	29,1	0,27	669,7
D05-B	0	2555	0,08	65,1	0,09	1200,2
D06-A	0	20	0	7,7	0	0
D06-B	4,47	702	8,35	10,5	1,49	8,7
D07-A	0	195	0	8,2	0	0
D07-B	1,94	711	1,94	9,5	0	0
D08-A	0	1727	0	19,1	0,39	160,7
D08-B	0,19	3175	0,93	123,8	0,19	346,1
D09-A	0,18	4109	0,43	52,1	0,56	365,5
D09-B	0	2754	1,15	151,2	0,07	224,8
D10-A	0	4193	0,2	122,2	0,11	878
D10-B	0	2644	0,51	107,3	0,04	1108,6
D11-A	0	540	0	15,4	0	0
D11-B	3,44	1280	0	17,4	3,44	2,2
D12-A	0	844	0	13,9	0	0
D12-B	0	687	0	15,1	0	0,1
D13-A	0	5047	0,38	58,7	0	195,4
D13-B	0	4288	1,17	97,2	0,2	359,3
D14-A	0	6388	0,59	102,3	0	302,9
D14-B	0	6178	1,38	102,8	0	452,9
D15-A	0	7840	0,64	145,7	0	274,4
D15-B	0	5220	0,6	95,6	0	389,6
D16-A	0	1397	7,69	23,1	0	0
D16-B	0	1043	0	26,4	0	0
D17-A	0	3506	0	24,8	0	0,07
D17-B	0	2089	0	23,7	0	0,08
D18-A	0	30044	1,28	81,4	0,45	215,5
D18-B	0,44	36643	3,22	98,7	1,34	312,5
D19-A	0,65	40955	3,82	87,6	1,3	525,2
D19-B	0,32	38600	2,51	81,9	0,64	700,5
D20-A	0	28139	0,18	18,4	0	111,9
D20-B	0	22104	0	12,7	0	170,7

Tabela 6.13: Resultados comparativos.

<i>Instância</i>	<i>GRASPH</i>		<i>AMH</i>		<i>ATH</i>		
	%gap	t(s)	%gap	t(s)	%gap	t(s)	
K	100	0,00	2,45	0,20	1,52	0,00	0,00
	200	0,00	9,00	0,00	2,40	0,00	0,00
	400	0,06	152,45	0,13	7,43	0,01	0,44
P	100	0,00	10,80	0,00	2,70	0,00	0,01
	200	0,00	72,00	0,00	7,20	0,00	0,70
	400	0,00	448,00	0,14	22,18	0,08	3,50
C	500	0,03	953,00	0,79	16,20	0,04	7,20
D	1000	0,40	6903,40	0,94	49,60	0,29	247,70

dio comercial, restaurante, posto de gasolina, etc. A coluna Unidades representa o número de clientes a serem atendidos no ponto. As duas colunas seguintes indicam o consumo de GN e de GLP. A Tabela 6.15 mostra o custo dos dutos de três fabricantes, e a média.

Tabela 6.14: Demanda estimada dos cliente da rede do bairro Candelária.

Consumidor	Unidades	Consumo m ³ /dia	Consumo GLP Kg/mês
1	8	3.4	80
2	8	3.4	80
3	30	12.6	300
4	48	20.3	480
5	11	4.7	110
6	15	6.3	150
7	20	8.4	200
8	96	40.5	960
9	6	2.5	60
10	11	4.7	110
11	-	39	908
12	14	5.9	140
13	81	34.2	810
14	54	22.8	540
15	27	11.4	270
16	120	50.6	1200
17	4	1.7	40
18	14	5.5	94

Tomando por base estas informações, foi montado um grafo representando este caso real. A Figura 6.1(a) é o modelo computacional para a fase 2 do projeto da Figura 2.4, e a Figura 6.1(b) é a solução obtida para o mesmo através do algoritmo aqui proposto. Os pontos em preto representam os clientes, e os pontos em branco possuem demanda igual a

Tabela 6.15: Custo dos dutos de três fabricantes.

Diâmetro	BRASTUBO	TIGRE	AFLON	Média
63mm	R\$ 11,50	R\$ 16,08	R\$ 10,25	R\$ 12,61
90mm	R\$ 23,28	R\$ 42,46	R\$ 19,15	R\$ 28,30
110mm	R\$ 34,53	R\$ 59,23	R\$ 27,39	R\$ 40,39
125mm	R\$ 44,80	R\$ 74,40	R\$ 34,92	R\$ 51,37

zero. Pode-se notar que apenas um subconjunto dos nós de demanda foi incluído na rede, proporcionando ao fornecedor do serviço a opção de não incluir na rede clientes que não proporcionarão lucros.

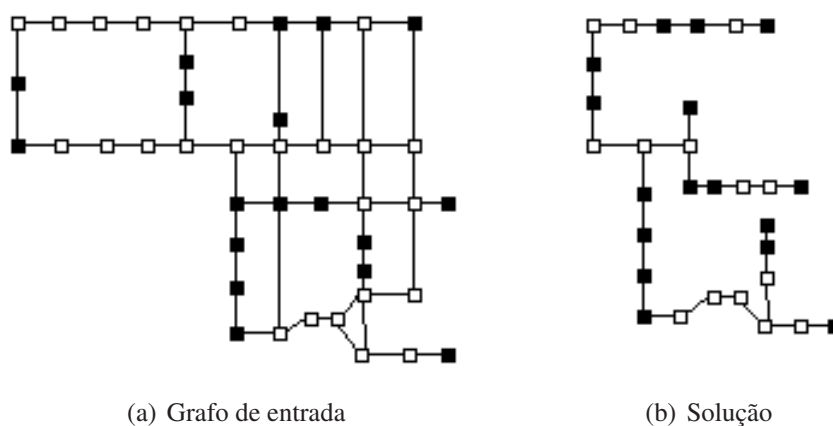


Figura 6.1: Representação computacional da Fase 2 do projeto de redes da cidade de Natal.

Capítulo 7

Considerações finais

Foi introduzido aqui um novo algoritmo evolucionário para a solução do Problema da Configuração de uma Rede de Distribuição de Gás Natural, modelado computacionalmente através do Problema de Steiner com Prêmios. Para testar a viabilidade computacional dessa metáfora desenvolveu-se um experimento computacional objetivando testar a mimetização do comportamento dos vetores do fluxo intra e extracelular.

Uma das conclusões obtidas com este trabalho foi a constatação de que o uso conjunto de dois tipos de vetores transgenéticos pode resultar em um bom desempenho. O algoritmo é dependente de parâmetros, como um número máximo de gerações, tamanho da população, e parâmetros para o controle da probabilidade de ataque dos vetores. Porém, sabe-se que o mesmo ocorre também com as demais abordagens clássicas.

Pode-se notar através dos resultados apresentados no Capítulo 6, que o algoritmo proposto encontra boas soluções quando comparado com outros dois algoritmos heurísticos encontrados na literatura. De fato, o AMH é superado pelos outros dois algoritmos, que apresentam resultados equivalentes.

O tempo computacional do ATH é menor que os tempos reportados no experimento GRASPH e AMH, exceto para a última classe de instâncias, onde o tempo do AMH é melhor que o tempo do ATH. Todavia ressalte-se que o ATH foi executado em uma plataforma superior à do algoritmo GRASP, e semelhante à do AMH. Adicionalmente, o AMH realiza uma considerável redução das instâncias, e pode-se concluir que com a redução do tamanho da entrada diminui-se o tempo necessário para o processamento. Estes fatores impedem uma comparação conclusiva com respeito ao tempo.

Com os experimentos realizados, concluiu-se que a mimetização do comportamento

dos vetores do fluxo intra e extracelular pode inspirar algoritmos computacionais eficientes quantitativa e qualitativamente.

Como trabalhos futuros, pode-se sugerir o desenvolvimento de experimentos adicionais com ATs. Podem ser desenvolvidas novas heurísticas para a geração de uma população inicial, e para compor o banco de informações genéticas podem também ser estudadas outras fontes de informação. Sugere-se também utilizar os vetores da transgenética que não foram abordados no escopo deste trabalho. A utilização de reduções dos grafos de entrada também pode ser uma boa estratégia para a diminuição do tempo computacional.

Neste trabalho o problema da configuração de redes de distribuição de GN foi modelado através do PSP e solucionado através da metáfora dos ATs, porém, o problema pode também ser modelado de outras formas, e solucionado através de outros métodos. Podem ser considerados fatores como a confiabilidade e múltiplos objetivos, atendendo assim a outros requisitos também importantes quando se projeta uma rede de distribuição de GN.

Bibliografia

- [ABL⁺97] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. Watson. *Biologia Molecular da Célula. 3a edição*. Porto Alegre: Editora Artes Médicas, 1997.
- [AHU74] A. Aho, J. Hopcroft, and J. Ullman. The Design and Analysis of Computer Algorithms. *Addison-Wesley, Reading*, 1974.
- [ARM05] J. C. C. Amasifen, R. Romero, and J. R. S. Mantovani. Algoritmos evolutivos dedicados à reconfiguração de redes radiais de distribuição sob demandas fixas e variáveis - estudo dos operadores genéticos e parâmetros de controle. *Sba: Controle e Automação Sociedade Brasileira de Automatica*, 16(3):303–317, 2005.
- [AS77] E. Alperovits and U. Shamir. Design of optimal water distribution systems. *Water Resources Research*, 13(6):885–900, 1977.
- [ASW94] F. N. Abuali, D. A. Schoenefeld, and R. L. Wainwright. Designing telecommunications networks using genetic algorithms and probabilistic minimum spanning trees. In *Proceedings of the 1994 ACM Symposium on Applied Computing*, pages 242–246, 1994.
- [Bag06] Ligia Bagi. Algoritmo transgenético na solução do problema do caixeiro comprador. Master's thesis, Programa de Pós-graduação em Sistema e Computação, Universidade Federal do Rio Grande do Norte, 2006.
- [Bal89] E. Balas. The prize collecting traveling salesman problem. *Networks*, (19):621–636, 1989.

- [Bar05] A. O. Barboza. *Simulação e Técnicas da Computação Evolucionária Aplicada à Problemas de Programação Linear Inteira Mista*. PhD thesis, UTFPR - Universidade Tecnológica Federal do Paraná, Engenharia Elétrica e Informática Industrial, 2005.
- [Bea89] J. Beasley. An SST-Based algorithm for the Steiner problem in graphs. *Networks*, (19):1–16, 1989.
- [Bea90] J. Beasley. Distributing test problems by electronic mail. *Journal of the Operational Research Society*, (41):1069–1072, 1990.
- [Beh96] M. Behe. Darwin's Black Box: The Biochemical to Evolution. *The Free Press, New York*, 1996.
- [BF95] L. Bull and T. Fogarty. Artificial symbiogenesis. *Artificial Life 2*, pages 269–292, 1995.
- [BGG06] L. Bagi, M.C. Goldberg, and E. F. Goldberg. A transgenetic algorithm for the capacited traveling purchaser problem. In *19th International Symposium on Mathematical Programming - Abstracts*, page 76, 2006.
- [BGSLW93] D. Bienstock, M. Goemans, D. Simchi-Levi, and D. Williamson. A note on the prize collecting traveling salesman problem. *Mathematical Programming*, (59):413–420, 1993.
- [BNO01] A. Borelli, H. Naka, and V. Oliveira. *Gás Natural - Transporte e Distribuição*. Instituto de Eletrotécnica e Energia (PIPGE-IEE/USP), Universidade Federal de São Paulo. Disponível em http://geocities.yahoo.com.br/jbonds_0072001/index.htm, 2001.
- [Cas04] Melissa Pereira Castro. Algoritmos Evolucionários para o Problema de Dimensionamento dos Dutos de uma Rede Urbana de Distribuição de Gás Natural. Master's thesis, Programa de Pós-graduação em Sistema e Computação, Universidade Federal do Rio Grande do Norte, 2004.

- [CGG02] M. P. Castro, E. F. Gouvêa, and M.C. Goldberg. Gas lift optimization problem: A transgenetic approach. In *17th World Petroleum Congress - Annals*, 2002.
- [CLMR03] A. S. Cunha, A. Lucena, N. Maculan, and M. G. C. Resende. A relax and cut algorithm for the prize collecting steiner problem in graphs. In *Proceedings of Mathematical Programming in Rio*, pages 72–78, Búzios, Rio de Janeiro, Brazil, 2003.
- [CLRS01] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Algoritmos: Teoria e Prática*. Tradução da Segunda Edição Americana por Vandenberg D. de Souza - Rio de Janeiro: Campus, 2001.
- [CRR01] S. Canuto, M. Resende, and C. Ribeiro. Local search with perturbations for the prize collecting Steiner tree problem in graphs. *Networks*, (38):50–58, 2001.
- [CS03] Antônio Augusto Chaves and Marcone Jamilson Freitas Souza. Modelagens exata e heurística para resolução do problema do caixeiro viajante com coleta de prêmios. Master's thesis, Monografia Submetida ao Departamento de Ciência da Computação da Universidade Federal de Ouro Preto - MG, 2003.
- [DAS97] B. Dengiz, F. Altiparmak, and A. E. Smith. Local search genetic algorithm for optimal design of reliable networks. volume 1, pages 179–188, 1997.
- [DC99] M. Dorigo and G. Di Caro. The ant colony optimization meta-heuristic. *New Ideas in Optimization, McGraw-Hill*, pages 11–32, 1999.
- [Dij59] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, (1):269–271, 1959.
- [DLCMR04] E. A. De La Cal Marín and L. S. Ramos. Supply estimation using coevolutionary genetic algorithms in the spanish. *Applied Intelligence*, (21):7–24, 2004.

- [dOR03] Iloneide Carlos de Oliveira Ramos. A protog algorithm applied to the traveling salesman problem. In *Annais of XXIII International Conference of The Chilean Computer Science Society*, pages 23–30, 2003.
- [DSM96] G. C. Dandy, A. R. Simpson, and L. J. Murphy. An improved genetic algorithm for pipe network optimization. *Water Resources Research*, 1996.
- [EG02] Luis Gonzalo Acosta Espejo and Roberto D. Galvão. O uso das relaxações lagrangeana e surrogate em problemas de programação inteira. *Pesqui. Oper. [online]*, 22(3):387–402, 2002.
- [EGLV98] S. Engevall, M. Göthe-Lundgren, and P. Värbrand. A strong lower bound for the node weighted steiner tree problem. *Networks*, 31(1):11–17, 1998.
- [EL03] M. M. Eusuff and K. E. Lansey. Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources Planning and Management*, 129(3):210–225, 2003.
- [ER64] P. R. Ehrlich and P. H. Raven. Butterflies and plants: A study in coevolution. *Evolution*, (18):586–608, 1964.
- [ESBT94] G. Eiger, U. Shamir, and A. Ben-Tal. Optimal design of water distribution networks. *Water Res. Research*, 30(9), 1994.
- [FFFP03] P. Feofiloff, C. E. Fernandes, C. E. Ferreira, and J. C. Pina. Primal dual approximation algorithms for the prize collecting steiner tree problem. *Submitted*, 2003. To appear.
- [Fis91] M. Fischetti. Facets of two steiner arborescence polyhedra. *Mathematical Programming*, 51:401–419, 1991.
- [FR95] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, (6):109–133, 1995.
- [GB90] I. C. Goulter and F. Bouchart. Reliability-constrained pipe network model. *Journal of Hydraulic Engineering*, 116(2):211–229, 1990.

- [GBG06] Marco Cesar GOLDBARG, Ligia Bariani BAGI, and E. F. G. GOLDBARG. Algoritmo transgenético para o problema do caixeiro comprador capacitado. In *Proceedings of the International Joint Conference IBERAMIA/SBIA/SBRN 2006 - 1st Workshop on Computational Intelligence*, volume 1, pages 23–28, Ribeirão Preto, 2006.
- [GCG04] E. F. G. Goldberg, M. P. Castro, and M. C. Goldberg. Transgenetic algorithm for the gas network pipe sizing problem. In *Proceedings of the Brazilian Symposium on Neural Networks. IEEE. IEEE Computer Society Press 1*, pages 100–106. Símposio Brasileiro de Redes Neurais, São Luiz, 2004.
- [GDG06] Marco Cesar GOLDBARG, Herbert de Mélo DUARTE, and E. F. G. GOLDBARG. Algoritmo transgenético para a solução do problema do passeio do pistoneio periódico. In *Annals of XIII CLAIO Latin Iberoamerican Operations Research Conference*, 2006.
- [GG01] M. C. Goldberg and E. F. Gouvêa. An extra-celular transgenetic algorithm applied to the graph coloring problem. In *Anais do IV Congresso Chileno de Investigación Operativa*, 2001.
- [GG02] M. C. Goldberg and E. F. G. Goldberg. Transgenética computacional: Uma aplicação ao problema quadrático de alocação. *Pesquisa Operacional*, 22(3):359–386, 2002.
- [GGC94] M. C. Goldberg, E. F. G. Goldberg, and M. F. Campos. Software de apoio à decisão para o projeto de redes urbanas de distribuição de gás natural. In *In Rio Oil and Gas 2004 - Expo and Conference*, 1994.
- [GGC02] E. F. Gouvêa, M. C. Goldberg, and W. E. Costa. Evolutionary algorithms applied to the workover rigs schedule problem. In *Annals of XI Latin-Iberian American Congress of Operations Research 1*, 2002.
- [GGC04] E. F. G. GOLDBARG, M. C. GOLDBARG, and W. E. COSTA. A transge-

- netic algorithm for the permutation flow-shop sequencing problem. *WSEAS Transactions on Systems*, 3(1):40–45, 2004.
- [GGMN02] M. C. Goldberg, E. F. G. Goldberg, and F. D. Medeiros Neto. Piston pump mobile unity tours problem: An evolutionary view. In *Proceedings of the Genetic and Evolutionary Computation Conference 2002 - GECCO 2002*, page 1264, 2002.
- [GGMN04] E. F. G. Goldberg, M. C. Goldberg, and F. D. Medeiros Neto. An evolutionary approach for the piston pump mobile unit problem. In *Proceedings of the Fifth International Conference on Computer Sciences*, pages 281–290, 2004.
- [GGMN05] M. C. Goldberg, E. F. G. Goldberg, and Francisco Dantas de Medeiros Neto. Algoritmos evolucionários na determinação da configuração de custo mínimo de sistemas de co-geração de energia com base no gás natural. *Pesquisa Operacional*, 25(2):231–259, 2005.
- [GGR03] Marco César GOLDBARG, E. F. G. GOLDBARG, and Iloneide Carlos de Oliveira RAMOS. A protog algorithm applied to the traveling salesman problem. *Wseas Transactions On Computers*, 2(2):299–304, 2003.
- [GGS01] M. C. Goldberg, E. F. Gouvêa, and L. M. Silva. Protog: a computational transgenetic algorithm. In *In 4Th Metaheutistics International Conference - MIC2001*, pages 321–326, 2001.
- [GGS02] M. C. Goldberg, E. F. G. Goldberg, and C. M. P. Souza. An evolutionary approach to the manifolds and wells placement problem in offshore fields. In *Annais of XI Latin-Iberian American Congress of Operations Research*, 2002.
- [GL00] M. C. Goldberg and H. P. L. Luna. *Programação Linear e Otimização Combinatória: Modelos e Algoritmos.*, volume 1. Rio de janeiro: Campus, 2000.

- [Glo86] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 5, 1986.
- [Glo96] F. Glover. Tabu search and adaptive memory programming - advances, applications and challenges. *Interfaces in Computer Science and Operations Research*, pages 1–75, 1996.
- [Goe94] M. X. Goemans. The steiner tree polytope and related polyhedra. *Mathematical Programming*, 63:157–182, 1994.
- [Gol89] D.E. Goldberg. Genetic algorithms in search. In *Optimization and Machine Learning*, Addison-Wesley, Berkeley, 1989.
- [Gou01] E. F. Gouvêa. *Transgenética Computacional: Um Estudo Algorítmico*. Tese de Doutorado, Programa de Engenharia da Produção, COPPE-UFRJ, Rio de Janeiro, 2001.
- [GW96] M. X. Goemans and D. P. Williamson. The primal dual method for approximation algorithms and its application to network design problems. In D. Hochbaum editor, *Approximation algorithms for NP-hard problems*, pages 144–191, 1996.
- [Hil90] D. W. Hillis. Co-evolving parasites improve simulated evolution in an optimization procedure. *Physica D*, 42:228–234, 1990.
- [HS06] M. Haouari and J. C. Siala. A hybrid lagrangian genetic algorithm for the prize collecting steiner tree problem. *Comput. Oper. Res.*, 33(5):1274–1288, 2006.
- [Hus94] P. Husbands. Distributed coevolutionary genetic algorithms for multi-criteria and multiconstraint optimisation. *Evolutionary Computing, AISB Workshop Selected Papers*, 865:150–165, 1994.
- [JH06] Kamal Jain and Mohammad T. Hajiaghayi. The prize-collecting generalized steiner tree problem via a new approach of primal-dual schema. In

- SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 631–640, Miami, Florida, 2006.
- [JMP00] D. S. Johnson, M. Minkoff, and S. Phillips. The prize collecting steiner tree problem: Theory and practice. In *In Proceedings of 11th ACM-SIAM Symposium on Discrete Algorithms*, pages 760–769, San Francisco, CA, 2000.
- [KAFS97] N. Kubota, T. Arakawa, T. Fukuda, and K. Shimojima. Trajectory generation for redundant manipulator using virus evolutionary genetic algorithm. In *Robotics and Automation. Proceedings.*, volume 1, pages 205–210, IEEE International Conference, 1997.
- [Kar76] R. M. Karp. The probabilistic analysis of some combinatorial search algorithms. In: *J. F. Traub (Ed.) Algorithms and Complexity: New Directions and Recent Results*, Academic Press, pages 1–19, 1976.
- [KFAS97] N. Kubota, T. Fukuda, T. Arakawa, and K. Shimojima. Evolutionary transition on virus - evolutionary genetic algorithm. pages 291–296, 1997.
- [KG99] Jong Ryul Kim and Mitsuo Gen. Genetic algorithm for solving bicriteria network topology design problem. In *IEEE Congress on Evolutionary Computation*, 1999.
- [KGV83] S. Kirkpatrick, D.C. Gellat, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220, 1983.
- [Kim68] M. Kimura. Evolutionary rate at the molecular level. *Nature*, 217:624–626, 1968.
- [Kir04] Denis Kirchhoff. Avaliação de Risco Ambiental e o Processo de Licenciamento: O Caso do Gasoduto de Distribuição Gás Brasileiro Trecho São Carlos Porto Ferreira. Master's thesis, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2004.

- [KKK01] J. Y. Kim, Y. Kim, and Y. K. Kim. An endosymbiotic evolutionary algorithm for optimization. *Applied Intelligence*, 15:117–130, 2001.
- [KLM⁺03] G. W. Klau, I. Ljubic, P. Mutzel, U. Pferschy, and R. Weiskircher. The fractional prize collecting steiner tree problem on trees. In *G. D. Battista and U. Zwick, editors, ESA 2003*, 2832 of LNCS:691–702, 2003.
- [KLM⁺04] G. W. Klau, I. Ljubic, A. Moser, P. Mutzel, P. Neuner, U. Pferschy, G. Raidl, and R. Weiskircher. Combining a memetic algorithm with integer programming to solve the prize collecting steiner tree problem. In *In K. Deb, editor, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*, volume 3102 of LNCS, pages 1304–1315, Springer-Verlag, 2004.
- [KPK03] Y. K. Kim, K. Park, and J. Ko. A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. *Computers and Operations Research*, 30(8):1151–1171, 2003.
- [Kru56] J. B. N. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Amer. Math. Soc.*, 7:48–50, 1956.
- [KSF96a] N. Kubota, K. Shimojima, and T. Fukuda. The role of virus infection in virus-evolutionary genetic algorithm. In *Proceedings of IEEE International Conference*, volume 1, pages 182–187, Evolutionary Computation, 1996.
- [KSF96b] N. Kubota, K. Shimojima, and T. Fukuda. Virus-evolutionary genetic algorithm-coevolution of planar grid model. In *Proceedings of the Fifth IEEE International Conference*, volume 1, pages 8–11, Fuzzy Systems, 1996.
- [Lan01] E. Lander. The human genome. *Science*, 291(5507):1177–1180, 2001.
- [Lev03] A. Levin. A better approximation algorithm for the budget prize collecting tree problem. *Faculty of Industrial Engineering and Management Technion*, 2003.

- [LFGG04] L. E. C. Leite, G. Souza Filho, M. C. Goldberg, and E. F. Gouvêa. Comparando algoritmos genéticos e transgenéticos para otimizar a configuração de um serviço de distribuição de vídeo baseado em replicação móvel. In *Anais do 22º Simposio Brasileiro de Redes de Computadores, SBRC'2004*, volume 1, pages 129–132, 2004.
- [Lju04a] I. Ljubic. An evolutionary approach to the fractional prize-collecting steiner tree problem. *Walter Karplus Studen Summer Research Grant, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Áustria*, 2004.
- [Lju04b] Ivana Ljubic. *Exact and Memetic Algorithms for Two Network Design Problems*. PhD thesis, Technische Universität Wien, Fakultät für Informatik, 2004.
- [LM99] M. Laguna and R. Martí. Grasp and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing*, 11:44–52, 1999.
- [LR00] A. Lucena and M. G. C. Resende. Strong lower bounds for the prize collecting steiner tree problem in graphs. Technical report, AT&T Technical Report, disponível em <http://www.research.att.com/~mgcr>, 2000.
- [LWP⁺04] I. Ljubic, R. Weiskircher, U. Pferschy, G. W. Klau, P. Mutzel, and M. Fischetti. Solving the prize-collecting steiner tree problem to optimality. Technical Report TR-186-1-04-01,14, Forschungsbericht / Technical Report, 2004.
- [LWP⁺06] I. Ljubic, R. Weiskircher, U. Pferschy, G. W. Klau, P. Mutzel, and M. Fischetti. An algorithmic framework for the exact solution of the prize-collecting steiner tree problem. *Mathematical Programming*, 105:427–449, 2006.
- [Mar91] L. Margulis. Symbiosis as a source of evolutionary innovation: Speciation and morphogenesis. *The MIT Press*, 1991.

- [Mar98] L. Margulis. *Symbiotic Planet: A New Look at Evolution*. Amherst, Massachusetts: BasicBooks, 1st edition edition, 1998.
- [MH97] N. Mladenovic and P. Hansen. Variable neighbourhood search. *Computer and Operations Research*, 24:1097–1100, 1997.
- [Mic98] K. Michaelian. A symbiotic algorithm for finding the low energy isomers of large clusters and molecules. *Chem. Phys. Lett.*, 293:202–215, 1998.
- [Min00] M. Minkoff. The prize collecting steiner tree problem. Master's thesis, M.S. Thesis, Massachusetts Institute of Technology, 2000.
- [MM99] C. E. Mariano and E. Morales. A multiple objective ant-q algorithm for the design of water distribution irrigation networks. Technical Report HC-9904, Instituto Mexicano de Tecnologia del Agua, June 1999.
- [MN92] P. Moscato and M. G. Norman. A memetic approach for the travelling salesman problem - implementation of a computational ecology for optimisation on message-passing systems. In *Proceedings of the International Conference on Parallel Computing and Transputer Applications, Amsterdam, IOS Press*, 1992.
- [Mor92] H. J. Morowitz. *Beginning of Cellular Life*. New Haven, Conn, Yale University Press, 1992.
- [MS89] J. Maynard Smith. *Evolutionary genetics*. Oxford: Oxford University Press., 1989.
- [NW88] K. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization*. John Wiley & Sons, New York, 1988.
- [PM02] L. Pagie and M. Mitchell. A comparison of evolutionary and coevolutionary search. *International Journal of Computational Intelligence and Applications*, 2(1):53–69, 2002.

- [PR05] Jakob Puchinger and G. R. Raidl. Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. *Institute of Computer Graphics and Algorithms, Vienna University of Technology, Vienna, Austria*, 2005.
- [Pri57] R. C. Prim. Shortest connection networks and some generalizations. *Bell System Tech, J.*, 36:1389–1401, 1957.
- [r11] Potigás - companhia potiguar de gás. Disponível em <http://www.potigas.com.br>, último acesso em Janeiro de 2007.
- [r16] Ctgas - centro de tecnologias do gás. Disponível em <http://www.ctgas.com.br>, último acesso em Janeiro de 2007.
- [r25] Gasmig - companhia de gás de minas gerais. Disponível em <http://www.gasmig.com.br>, último acesso em Junho de 2006.
- [r26] Gasnet - site do gás natural. Disponível em <http://www.gasnet.com.br>, último acesso em Janeiro de 2007.
- [r3] Anp - agência nacional do petróleo. Disponível em <http://www.anp.gov.br>, último acesso em Janeiro de 2007.
- [r78] Wikipedia - a enciclopédia livre. Disponível em <http://www.wikipedia.org>, último acesso em Dezembro de 2006.
- [r84] Aneel - agência nacional de energia elétrica. Disponível em <http://www.aneel.gov.br>, último acesso em Setembro de 2006.
- [RB97] C. D. Rosin and R. K. Belew. New methods for competitive co-evolution. *Evolutionary Computation*, 5(1):1–29, 1997.
- [Rib96] C. C. Ribeiro. Metaheuristics and applications. *In Advanced School on Artificial Intelligence, Estoril, Portugal*, 1996.
- [RS04] L. C. A. Rodrigues and H. A. Silva. Otimização do projeto de redes urbanas baseado no problema de steiner. *VI Simpósio Brasileiro de GeoInformática - Campos do Jordão, SP, Brasil*, 2004.

- [Seg87] A. Segev. The node weighted steiner tree problem. *Networks*, 17:1–17, 1987.
- [SGG06] C. C. Schmidt, E. F. Goldberg, and M.C. Goldberg. A transgenetic algorithm for the prize collecting steiner tree problem. In *19th International Symposium on Mathematical Programming - Abstracts*, page 90, 2006.
- [Sha99] J. A. Shapiro. Genome system architecture and natural genetic engineering in evolution. *Ann. NY. Acad. Sci.*, 870:22–35, 1999.
- [Sou04] Givanaldo Rocha Souza. *Uma Abordagem Evolucionária para o Problema da Configuração de uma Rede Urbana de Distribuição de Gás Natural*. Trabalho de graduação apresentado à Universidade Federal do Rio Grande do Norte, 2004.
- [SS97] H. D. Sherali and E. P. Smith. A global optimization approach to a water distribution network design problem. *Journal of Global Optimization*, 11(2):107–132, 1997.
- [TMG01] Y. Tsujimura, Y. Mafune, and M. Gen. Effects of symbiotic evolution in genetic algorithms for the job-shop scheduling. *Proceedings of the 34 Hawaii International Conference on Systems Science*, pages 1–7, 2001.
- [Uch06] Eduardo Uchoa. Reduction tests for the prize-collecting steiner problem. *Oper. Res. Lett.*, 34(4):437–444, 2006.
- [WB05] C.M. Waters and B.L. Bassler. Quorum sensing: Cell-to-cell communication in bacteria. *Annual Review of Cell and Developmental Biology*, 21:319–346, 2005.
- [Wer01] R. F. F. Werneck. Problema de steiner em grafos: Algoritmos primais, duais e exatos. Master’s thesis, PUC-RJ, 2001.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)