# OTIMIZAÇÃO APLICADA AO PLANEJAMENTO DE ANÉIS UNIDIRECIONAIS EM REDES DE TELECOMUNICAÇÃO

por

Patricia Regina de Abreu Lopes

IM/NCE - UFRJ

2004

# OTIMIZAÇÃO APLICADA AO PLANEJAMENTO DE ANÉIS UNIDIRECIONAIS EM REDES DE TELECOMUNICAÇÃO

**Patricia Regina de Abreu Lopes**

Dissertação submetida ao Corpo Docente do Núcleo de Computação Eletrônica - Instituto de Matemática da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários para a obtenção do grau de Mestre.

Aprovada por:

---

Profª. Marcia Helena Costa Fampa, D.Sc

orientadora

---

Dr. Silvio Binato, D.Sc

co-orientador

---

Profª. Maria Helena Cautiero Jardim, D.Sc

---

Profª. Fernanda Maria Pereira Raupp, D.Sc

Rio de Janeiro, RJ - Brasil

2004

## FICHA CATALOGRÁFICA

Lopes, Patricia Regina de Abreu.
Otimização Aplicada ao Planejamento de Anéis Unidirecionais em Rede deTelecomunicação / Patricia Regina de Abreu Lopes. Rio de Janeiro: UFRJ IM, 2004.
Dissertação - Universidade Federal do Rio de Janeiro, IM / NCE.
1. Introdução.
2. O Problema de Planejamento de Anéis Unidirecionais em Rede de Telecomunicações (PAURT).
3. Metaheurística
4. Aplicação do GRASP ao problema de PAURT
5. Resultados Computacionais
6. Conclusão e Trabalhos Futuros
(Mestrado-UFRJ/IM/NCE) I. Título.

A Deus e ao meu pai, Raimundo Lopes
(*in memorian)*

# Agradecimentos

A Deus, pelo término de mais uma etapa da minha vida.

A minha família por todo apoio ao longo do curso de mestrado.

A professora e orientadora Marcia Helena Costa Fampa, meus agradecimentos pelo seu apoio, carinho, por seu exemplo, pelos conhecimentos transmitidos durante o desenvolvimento deste trabalho e durante o curso.

Ao meu co-orientador Silvio Binato por sua colaboração, disponibilidade e pelo conhecimento transmitido.

Aos professores Ângela Beazutti, Sandra Malta e Carlos Frederico que me incentivaram a fazer o mestrado e colaboraram para que meu sonho pudesse ser realizado, o meu eterno agradecimento.

Ao professor Mauro Antônio Rincon por me aceitar no mestrado, acreditar no meu potencial, por sua dedicação como professor e orientador meu eterno agradecimento.

Aos professores que ministraram alguma matéria ao longo do meu curso de mestrado contribuindo assim para minha formação em especial a professora Sandra Malta e ao professor Nelson Maculan.

Aos professores Jayme Szwarcfiter, Fábio Protti e Marcos Passini pela colaboração e pelo tempo que dispuseram para tirar minhas dúvidas.

A minha irmazinha de coração e orientadora para assuntos aleatórios, Denise Candal por seu apoio nos momentos de crise existencial, por seu carinho, pelas inúmeras horas de risadas e troca de conhecimentos.

Ao meu amigão Júlio Tadeu Carvalho da Silveira por sua amizade, companheirismo e pelas inúmeras horas de programação. Devo a ele todo o êxito da implementação.

Ao Cracky por sua amizade, por me fazer rir com suas rabugices, pelos conhecimentos transmitidos e por ajudar a fazer uma parte do programa.

Ao amigos que fiz no Controlab, dentre eles Eliana Aude, Flavio Signorelli Mendes, Henrique Serdeira e Marcela Souto M. da Silva.

A todas as pessoas que me ajudaram com seu apoio, não posso deixar de citar meus colegas de mestrado Ismael, Eugênio, Ângelo, Anderson, Bruno, Werner, Reinaldo,

Mitre, Raquel, André, Christiane, Sidney, Rosa, Zero, Márcio, Haroldo, Vinícius, Juliana, Thayse, Marcelo, Gisely, Michaelle, etc.

Ao Rafael di Lego Gonçalves por sua amizade e por deixar sempre o computador funcionando para que eu pudesse trabalhar e ao Gabriel Pereira da Silva por sua amizade e colaboração em todos os momentos que precisei.

Aos amigos da secretaria, *tia* Deise Lobo Cavalcante por toda a sua dedicação e carinho, juntamente com Lina e Adriana.

Ao Professor Ageu Cavalcanti Pacheco Junior por sua disponibilidade e incentivo nos trabalhos acadêmicos.

Ao Sergio Guedes de Souza por seu incentivo para que entrasse no mestrado.

Aos funcionários do NCE, desde o suporte até a portaria por sua atenção em especial Roberto, Nilson, Juan, Sr Oswaldo, Ferreiro, Sr Didi, as(os) recepcionistas por seu carinho.

A minha amiga Doris Damian e suas filhas Hela Flavia e Hela Roberta por sua amizade, carinho e inúmeros fins de semana ao qual me disponibilizou sua casa, seu computador, principalmente por sua lazanha e chiffon imperdíveis meu eterno agradecimento.

Ao Antonio Carlos G. L. dos Santos, por todos os momentos que passamos e pelo carinho. Será sempre uma pessoa muito especial para mim. Eu nunca o esquecerei !

Ao amigos Marcus Iuri da Silva de Freitas, Eni Galheiros Poça e Letícia Doherty por seu carinho e amizade, nunca os esquecerei !

Ao professor Adilson e professor Maculan por disponibilizar o Labotim.

Aos amigos da Coppe em especial a Ana Lucia Sousa e Moises de Carvalho Teles Junior pelos momentos alegres e tristes que passamos, pelo carinho e apoio. Ao Jorge, Elder, Henrique, Tiberius, André Brito e Luidi por colaborarem com alguma informação.

A Maria Helena Cautiero Jardim e Fernanda Maria Pereira Raupp pelo carinho, atenção, e por participarem da banca.

A Capes e ao NCE por me disponibilizar uma bolsa.

A todos aqueles que de uma forma ou de outra me apoiaram o meu muito obrigada.

″ Quando a gente pensa que sabe todas as respostas, vem a vida e muda todas as perguntas. ″

# Resumo

São inúmeros os problemas de otimização de origem no setor de telecomunicações. Com o desenvolvimento de novas técnicas, novos equipamentos e o aumento da busca dos serviços surge a necessidade de otimizar a nova rede. Muitos desses problemas são de característica combinatorial.

Este trabalho considera um problema de planejamento de uma rede de telecomunicações, de sua expansão ou mesmo sua avaliação periódica. O problema consiste especificamente em determinar os anéis que serão utilizados e conseqüentemente determinar os equipamentos de transmissão que serão utilizados na rede de forma a atender toda a demanda prevista a um custo mínimo de instalação.

Para modelagem levamos em conta a utilização de uma rede de telecomunicações, anéis unidirecionais e o equipamento Add Drop Mutiplexer (ADM). A modelagem faz uso da teoria de grafos, programação linear inteira e da metaheurística GRASP e a resolução do problema é obtida implementando o algoritmo em C e utiliza o pacote de otimização XPress-MP.

# Abstract

There are many combinatorial optimization problems in telecommunications industry. Once new techniques and equipments are developed and the demand is increasing, it is necessary to optimize a new network. Many of these problems have combinatorial characteristics.

This work considers a problem of planning a telecommunication network, its expansion and even its periodic evaluation. It consists in choosing the rings to be used, as well as the equipments that have to be used in the network in order to satisfy the demand with a minimum installation cost.

To model the problem we take into account a network telecommunication, unidirectional synchronous rings and the quipment Add Drop Mutiplexer (ADM).

The model uses Graph Theory as well as Integer Linear Programming and we use the metaheuristic GRASP, implemented in C using XPress-MP, to solve the problem.

# Sumário

**1 Introdução** **1**

1.1 História da Rede de Telecomunicações . . . . . 1

1.2 Motivação . . . . . 5

1.3 Revisão Bibliográfica . . . . . 6

1.4 Organização do Trabalho . . . . . 8

**2 O Problema de Planejamento de Anéis Unidirecionais em Redes de Telecomunicação (PAURT)** **10**

2.1 Introdução . . . . . 10

2.1.1 Anéis Unidirecionais . . . . . 12

2.2 Apresentação do problema . . . . . 14

2.3 Modelo Matemático do Problema (PAURT) . . . . . 16

2.4 Considerações sobre o Modelo . . . . . 18

**3 Metaheurísticas** **21**

3.1 Introdução . . . . . 21

3.2 Heurísticas . . . . . 22

3.3 Metaheurísticas . . . . . 23

3.4 Metaheurística GRASP . . . . . 24

3.4.1 Algoritmo GRASP Básico . . . . . 24

3.4.2 Fase de construção (Construção Randômica) . . . . . 24

3.4.3 Algoritmo de Construção Básico . . . . . 27

3.4.4 Função Tendência . . . . . 27

3.4.5 Fase de Busca Local . . . . . 28
3.4.6 Conclusões . . . . . 29

**4 Aplicação de GRASP ao PAURT** **31**
4.1 O procedimento Leia Dados de Entrada . . . . . 32
4.2 Sub-fase Construção Ciclo . . . . . 33
4.3 Sub-fase Construção Anéis . . . . . 37
4.4 Busca Local PAURT . . . . . 45

**5 Resultados Computacionais** **52**
5.1 Gerador de Dados Iniciais . . . . . 52
5.2 Análise dos Resultados . . . . . 52

**6 Conclusões e Trabalhos Futuros** **56**

**A Definições de Grafos** **61**
A.1 Propriedades . . . . . 61
A.1.1 Grafo Simétrico . . . . . 61
A.1.2 Caminho . . . . . 62
A.1.3 Ciclo . . . . . 62
A.1.4 Conexo . . . . . 62

**B Gerador de Grafos** **63**
B.1 Algoritmo Gerador de Grafo . . . . . 63
B.2 Gerador de Demandas . . . . . 64
B.3 Algoritmo Gerador de Demandas . . . . . 64

**C Geração de Ciclos** **67**
C.1 Estruturas de dados utilizadas na representação de um Grafo . . . . . 67
C.2 Algoritmo de geração de ciclos . . . . . 69
C.2.1 Algoritmo Ciclos . . . . . 70

# Lista de Abreviaturas

| **Abrev** | **Descrição** |
|---|---|
| DDD | Discagem Direta a Distância |
| SBTS | Sistema Brasileiro de Telecomunicações via Satélite |
| DDI | Discagem Direta Internacional |
| CPA | Central com Controle por Programa Armazenado |
| Telebrás | Sistema Brasileiro de Telecomunicações |
| STFC | Sistema Telefônico Fixo Comutado |
| WAP | Wireless Access Protocol |
| GPRS | General Packet Radio System |
| PCS | Personal Communucation Service |
| PAURT | Planejamento de Anéis Unidirecionais em Redes de Telecomunicação |
| ADM | ADD-DROP Multiplexer |
| LCR | Lista de Candidatos Restrita |
| GRASP | Greedy Randomized Adaptive Search Procedures |
| NA | Número de Anéis |
| bits | Menor unidade de informação |
| GSM | Global System for Mobile Communications |

# Lista de Tabelas

2.1 ADM's . . . . . 11

3.1 Opções para Função Tendência . . . . . 28

4.1 ADM's . . . . . 36
4.2 Estrutura E . . . . . 43

5.1 ADM's . . . . . 53
5.2 Problema Teste (5 nós) . . . . . 55

# Lista de Figuras


2.1 Exemplo de uma rede de telecomunicações. . . . . 12
2.2 Exemplos de ciclos suporte. . . . . 12
2.3 Exemplo de anel. . . . . 13
2.4 Anel Unidimencional - Tráfego. . . . . 13
2.5 Demandas. . . . . 15
2.6 Solução 1 . . . . . 15
2.7 Solução 2 . . . . . 16

3.1 pseudo código GRASP . . . . . 25
3.2 pseudo código Construção Randômica . . . . . 27
3.3 pseudo código Busca Local . . . . . 29

4.1 pseudo código GRASP-PAURT . . . . . 32
4.2 pseudo código da sub-fase Construção Ciclo . . . . . 34
4.3 Grafo G . . . . . 35
4.4 Exemplo Rede . . . . . 35
4.5 Tabela de demandas da rede . . . . . 36
4.6 Estrutura Ciclo - Demandas . . . . . 37
4.7 Estatística . . . . . 37
4.8 pseudo código da sub-fase Construção Anéis . . . . . 38
4.9 Estatistica 2 . . . . . 44
4.10 Ciclo - Demandas . . . . . 45
4.11 Estruturas de Busca Local . . . . . 46
4.12 pseudo código Busca Local . . . . . 47

4.13 pseudo código Avalia - Troca . . . . . 48
4.14 Solução . . . . . 50
4.15 Estruturas de Busca Local . . . . . 50
4.16 Solução Final . . . . . 51

5.1 Grafo G . . . . . 53
5.2 Tabela de demandas da rede . . . . . 53
5.3 Ciclo suporte . . . . . 54
5.4 Anel . . . . . 54

2.1 pseudo código Gerador de Grafo . . . . . 65
2.2 pseudo código Gerador de Demandas . . . . . 66

3.1 Estrutura de adjacência . . . . . 68
3.2 Exemplo 3 . . . . . 70
3.3 pseudo codigo Ciclo . . . . . 73
3.4 pseudo codigo nenhumciclo . . . . . 74
3.5 pseudo codigo Desmarca . . . . . 74
3.6 pseudo codigo Saidaciclo . . . . . 75

# Capítulo 1

# Introdução

## 1.1 História da Rede de Telecomunicações

Por Telecomunicações entende-se um *conjunto de dispositivos e técnicas para a transmissão de informações instantâneas a longa distância*. Essa transmissão pode ser de voz, sinais gráficos, dados, imagens ou sinais de televisão. Todos eles têm os mesmos princípios fundamentais, mas diferem na forma de manipular as informações e nos meios utilizados para transmiti-las.

Como exemplo temos os sistemas de telegrafia, telefonia, televisão e redes de dados informatizados que transmitem informações por meio da radiocomunicação, transmissão por cabo, por satélites artificiais e por fibras ópticas.

Três invenções foram o marco da impressionante evolução das telecomunicações até os dias de hoje. Em 1844, *Samuel Morse* inventou o telégrafo. Em 1876, *Graham Bell* inventou o telefone; em 1895 *Marconi*, o rádio. Mas é *Graham Bell* que se destaca como a grande figura mundial das telecomunicações.

Em 25 de junho de 1876, *Grahan Bell* demonstrou pela primeira vez em público seu invento e foi o imperador D. Pedro II, quem inaugurou o telefone. A uma distância de 150 metros, ele pôde ouvir *Grahan Bell* declamar *Shakespeare*.

Depois de *Graham Bell* o sistema de telecomunicações teve uma avanço muito grande e cada vez mais vem revolucionando a vida das pessoas trazendo mudanças profundas e

estas vêm de forma rápida impulsionando a economia e o modo de vida.

A fibra óptica é outra inovação revolucionária. Surgida em meados do século XX, essa tecnologia de informação permite a transmissão rápida e simultânea de milhares de chamadas telefônicas e dezenas de imagem por um filamento de vidro, sílica, náilon ou silicone de altíssima transparência e da espessura de um fio de cabelo humano, onde no seu interior circulam correntes pulsantes de luz laser. Para se ter uma idéia de seu impacto, um cabo de fibra óptica pode substituir até mil cabos coaxiais de cobre utilizados em conexões anteriores.

A microeletrônica criou o chip com dezenas de milhões de transistores e a digitalização deu às telecomunicações a mesma linguagem dos computadores. Nasceram as redes de computadores e, entre elas, a de maior impacto na vida das pessoas, a Internet.

O desenvolvimento dessa tecnologia permitiu a digitalização de todas as formas de comunicação como voz, dados, imagem, transformados em bits, que significa *a menor unidade de informação*, levando à convergência de sons, dados e imagem tratados em conjunto pelo computador, originando a multimídia e a realidade virtual.

Na telefonia fixa, serviços começam à ser oferecidos graças a tecnologia de acesso, que permitirá que usuários possam falar e usar a internet usando a mesma linha telefônica.

A humanidade assistiu ao longo do século XX várias evoluções tecnológicas. Uns dos maiores frutos dessa evolução são os satélites de telecomunicações, pois além de permitirem a retransmissão de programas de televisão, abriram novas perspectivas para a comunicação telefônica, a transmissão de dados, fax, Internet, comunicação móvel via telefone celular e muitos outros serviços especializados.

## Histórico

**1876**

7 de março - *Alexandre Graham Bell* obtém a patente da invenção do telefone.

**1877**

O primeiro telefone do país é instalado no Rio de Janeiro.

**1878**

O primeiro telefone público é instalado nos EUA.

**1879**

*D. Pedro II* autoriza o funcionamento da primeira empresa de telefonia no Brasil.

**1883**

A primeira estação telefônica do Brasil é instalada em Santos, com 75 assinantes.

**1884**

Os primeiros telefones começam a funcionar na cidade de São Paulo.

**1885**

*Lars M. Ericsson* revoluciona o design do telefone, acoplando bocal e fone numa única peça.

**1892**

*Alon B. Strowgeer* inaugura, em Indiana, a primeira central telefônica automática.

**1893**

As primeiras transmissões de sinais telegráficos e da voz humana em telefonia sem fio no mundo são realizadas pelo *Padre Landell de Moura*, na cidade de São Paulo.

**1922**

Os serviços de telegrafia e telefonia via rádio são introduzidos no Brasil, entre Rio de Janeiro e Nova Iorque. Nesse mesmo ano é inaugurada a primeira central telefônica automática do país em Porto Alegre.

**1946**

Entra em funcionamento o primeiro computador eletrônico dos Estados Unidos e Grã-Bretanha

**1956**

Começa a funcionar o primeiro cabo telefônico transatlântico entre Estados Unidos e Grã-Bretanha.

**1958**

O primeiro sistema de discagem direta a distância (DDD) da América do Sul é implantado no Brasil, entre Santos e São Paulo.

**1962**

Entra em operação o primeiro satélite mundial de telecomunicações, o Telstar, construído pelos Laboratórios Bell.

**1963**

Regulamentação do Código Brasileiro de Telecomunicações.

**1966**

Início da aplicação de fibra óptica em telecomunicações.

**1969**

Inaugurado o primeiro tronco sul de microondas da Embratel, interligando São Paulo, Curitiba e Porto Alegre.

**1970**

Inaugurado o sistema DDD.

**1971**

A Intel anuncia o invento do microprocessador, base dos futuros computadores.

**1972**

É criado o Sistema Telebrás, responsável pelas empresas governamentais de serviços públicos de telecomunicações do Brasil. São instalados os primeiros telefones públicos (orelhões) no Rio de Janeiro e em São Paulo. São implantados o Sistema Brasileiro de Telecomunicações via Satélite (SBTS) e o sistema de cabos submarinos ligando o Brasil à Europa, aos Estados Unidos e à Africa.

**1975**

O Brasil integra-se ao sistema de discagem direta internacional (DDI).

**1978**

A telefonia Móvel Celular é ativada no Japão.

**Década de 80** - foi marcada por problemas de congestionamento e falta de linhas, cada vez mais escassas, eram comercializadas a preço de ouro no mercado paralelo. O serviço precário tanto para os usuários domésticos como para os usuários corporativos a cada dia piorava.

**1982**

A primeira central de CPA (Central com Controle por Programa Armazenado) da América Latina é intalado em São Paulo.

**1990**

A primeira cidade brasileira a usar a Telefonia Móvel é o Rio de Janeiro.

**1995**

Quebra do monopólio estatal nas telecomunicações. Implantada a Internet comercial no Brasil.

**1997**

Assinada a Lei Geral de Telecomunicações que redefiniu o modelo institucional e possibilitou a criação de uma agência reguladora para o setor, a Anatel.

**1998**

Privatização do Sistema Brasileiro de telecomunicações - Telebrás. Início da competição no mercado nacional de Telecomunicações com a concorrência das operadoras ″espelho″ para o STFC (Sistema Telefônico Fixo Comutado). A competição passaria a mudar o cenário com maior e melhor oferta de linhas, melhoria de serviços e queda das tarifas. Um plano de investimentos setoriais até 2003. Inaugurada a privatização da banda B da telefonia celular, instalando a competição no segmento de telefonia celular. Aumento da competição com a entrada de novos operadores para o Serviço Móvel Pessoal (bandas C, D e E) e o uso mais intenso da tecnologia WAP (*Wireless access Protocol*) e GPRS (*General Packet Radio System*).

**1999**

A Telefônica introduz em São Paulo a tecnologia ADSL, que possibilita enviar e receber dados e imagens em altíssima velocidade.

**2000**

Escolha da faixa de 1.8 MHz para o PCS (personal Communucation Service).

**2001**

Assinatura das primeiras licenças GSM (Global System for Mobile Communications). Telefonia celular digital mais segura e usada no mundo.

## 1.2 Motivação

Um dos incentivos para o progresso tecnológico na indústria de telecomunicações é a necessidade dos usuários, porém com a popularização destes serviços surgiram novas

questões como a expansão e a eficiência da rede, como também novas formas de utilização (internet).

Neste trabalho consideramos o planejamento de redes, que vem a ser um problema que surge no estudo do dimensionamento de uma rede de telecomunicações ou na sua expansão. Com este planejamento, visa-se satisfazer a demanda por serviços, relacionados a grandes centros ou regiões.

Temos como intuito garantir a melhor relação custo-benefício, utilizando a arquitetura de anéis síncronos unidirecionais, ADM's (multiplexadores) em uma rede de telecomunicações.

Apresentamos um modelo principal e otimização para o problema descrito e outros modelos secundários que alteram ou mesmo excluem restrições do modelo principal. Os vários modelos apresentados fazem uso extensivo de váriáveis de decisão em geral associadas a escolha entre instalar ou não certos equipamentos na rede. Sugerimos uma metodologia para solução baseada na metaheurística GRASP.

Utilizamos como ferramentas computacionais para resolução deste problema a linguagem C e o pacote de otimização Xprees-MP.

## 1.3 Revisão Bibliográfica

Vários problemas ligados ao planejamento de redes de telecomunicação foram estudados, os quais diferem entre si por suas características e utilizam técnicas de resolução como branch and price, geração de colunas, metaheurística GRASP, etc.

O artigo interno da Embratel de M. Trindade [20] motivou esta dissertação. No referido artigo, um problema de planejamento de anéis síncronos unidirecionais, um modelo de programação linear inteira mista e um estudo do caso são propostos. Além disto o artigo traz alguns conhecimentos básicos, técnicos e específicos do problema.

O modelo descrito foi aplicado na cidade de Maceió, utilizando equipamentos SDH, modelo este semelhante a um dos utilizados nesta dissertação.

No artigo, uma solução onde não foi utilizada nenhuma ferramenta de software (resultado de um trabalho manual) é comparada a uma segunda solução utilizando o modelo de

otimização. Os resultados encontrados mostram que o modelo de otimização é a melhor opção.

O artigo de M. Resende [15] descreve quatro temas diferente onde são aplicados a metaheurística GRASP. Dentre estes, se encontra o problema de planejamento de uma rede SONET (rede americana correspondendo a Synchronous Digital Hierarchy - Hierarquia Digital Síncrona).

Nele é descrito um modelo de programação inteira para um problema de planejamento de rede. Além disto é apresentado uma abordagem heurística usada para encontrar soluções aproximadas do modelo inteiro. O limite inferior produzido pela resolução da relaxação linear do problema inteiro fornece uma indicação da qualidade da solução aproximada encontrada. Um problema pequeno de rede é usado para ilustrar o procedimento.

A dissertação de mestrado A. Brito [2], faz uso da técnica de geração de coluna e de uma heurística para resolver o problema de dimensionamento da rede, utilizando arquitetura de anéis e malhas.

O método heurístico utiliza basicamente os conceitos de ciclos e anéis com a finalidade de alocar as demandas. Ambas as técnicas levam em consideração o tamanho do ciclo suporte viável (csv) ($3 \leq n \leq 16$), um valor máximo para a soma dos comprimentos das arestas do csv e um comprimento máximo de cada aresta.

A heurística foi testada para uma rede com 10 nós, 36 demandas e na qual foi possível alocar todas as demandas. Para uma rede de 33 nós foi possível alocar 953 demandas num total de 956 demandas. Para as três demandas que não foram alocadas nos anéis foi proposta a alocação na malha ou a criação de novas conexões.

O artigo utiliza o método de geração de colunas, com um modelo que se divide em dois problemas. Um problema mestre com as restrições principais relacionadas ao anel e a malha e um problema secundário que gera em cada iteração uma nova coluna (novo anel) para o problema mestre. O método foi aplicado em problema com 10 nós, 36 demandas tendo sido atendidas todas as demandas.

A tese de doutorado M. Passini [10], estuda a resolução de problemas relacionados a redes de telecomunicações envolvendo concepção de redes e roteamento de multifluxos. Especificamente é estudado o problema de definição de anéis auto-regenerativos unidi-

recionais e o problema prático de sobrepor anéis a uma rede em malha pré-existente (Problema de duas arquiteturas).

Esta tese faz uso da técnica de branch and price tendo como idéia básica a decomposição do programa em um programa mestre que guia a busca ao longo da árvore, ora ramificando-a, ora gerando novas colunas. É proposta a solução utilizando algoritmo COLGEN, apresentado em duas versões e também mostrando algumas possíveis opções, heurísticas ou exatas, para tratar o programa secundário, reduzindo o número de soluções viáveis através de procedimentos e conhecimentos específicos do problema. Leva em conta também o comprimento total dos ciclos suportes criados como em [2].

Os resultados númericos apresentados constituem uma rede de 10 nós e de 7 nós, usando o algoritmo COLGEN que utiliza conhecimentos de geração de coluna e resolvendo o problema relaxado através do método simplex. Outro método empregado faz uso de resoluções de programação linear inteira, do método de branch and bound e branch and price.

No artigo [9], Maculan,Passini e Brito descrevem o resultado dos trabalhos [2] e [10] descritos anteriormente.

## 1.4 Organização do Trabalho

O trabalho apresentado nesta dissertação está dividido em 6 capítulos devidamente descritos abaixo.

No primeiro capítulo apresentamos, um breve resumo da história da rede de telecomunicações e a motivação que nos levou a escolher este assunto.

No segundo capítulo apresentamos informações básicas para facilitar o entendimento do problema considerado neste trabalho. Dentre os conceitos introduzidos, descrevemos o equipamento de transmissão de dados utilizado, o tipo de proteção de rede disponível, e os conceitos de anéis. Apresentamos o problema que deu origem a esta dissertação, o modelo matemático que o representa e fazemos algumas considerações importantes sobre o mesmo.

No terceiro capítulo apresentamos uma introdução à heurística e meta-heurística.

Consideramos algumas meta-heurísticas indicando referências para maiores detalhes e descrevemos com maior detalhe a metaheurística GRASP utilizada na resolução do problema em questão.

No quarto capítulo apresentamos detalhadamente a resolução do problema proposto nesta dissertação utilizando a metaheurística GRASP.

No quinto capítulo apresentamos os resultados numéricos.

No sexto capítulo apresentamos a conclusão da dissertação e propomos alguns trabalhos futuros.

No apêndice A, apresentamos definições, propriedades e conceitos básicos de grafos que utilizaremos.

No apêndice B, apresentamos os geradores de grafo e de demanda, com seus respectivos algoritmos.

No apêndice C, apresentamos o algoritmo utilizado para geração de ciclos numa dada rede. Consideramos a geração de anéis e discutimos alguns tópicos que serão importantes para o entendimento do algoritmo que resolve o problema em questão.

# Capítulo 2

# O Problema de Planejamento de Anéis Unidirecionais em Redes de Telecomunicação (PAURT)

## 2.1 Introdução

Consideraremos neste trabalho uma rede de telecomunicação que se distribui em cidades, centros ou regiões. Representaremos a rede de telecomunicações por um grafo não direcionado $G = (\mathcal{N}, E)$, onde cada nó $i$ representa uma central da rede e cada aresta do grafo representa uma conexão da rede (cabo de fibra ótica). Para cada par de pontos $i$ e $j$ desta rede, define-se um ou mais pacotes de dados $d_{ij}$, que deverão ser transmitidos de $i$ para $j$. Quando não existir dados entre os nós $i$ e $j$ então $d_{ij}$ é igual a zero.

Para transmitir estes dados utilizaremos o equipamento de transmissão *ADD-DROP Multiplexer*, ou simplesmente ADM. Este é composto por duas pontas, entrada e saída de dados, isto é, um ADM é capaz de inserir e subtrair dados da rede simultaneamente.

Neste trabalho faremos uso de dois tipos de ADM's apresentados na tabela 2.1, onde a capacidade máxima de transmissão e o custo de cada tipo do equipamento são também fornecidos.

Definimos ciclo suporte como um subconjunto de pontos da rede que se interligam por

Tabela 2.1: ADM's

| ADM | Capacidade | Preço ($10^6$R$) |
|---|---|---|
| ADM - 1 | 1008 Mbps | 250,00 |
| ADM - 2 | 4032 Mbps | 450,00 |

fibra ótica. Quando instalamos equipamentos de transmissão nos ciclos suportes, estes passam a ser definidos como anel.

A transmissão de dados na rede considerada será implementada através de uma arquitetura de anel. Neste tipo de arquitetura a transmissão de dados entre cada par de pontos da rede é realizada através de um anel que contém estes pontos.

A transmissão de dados de um ponto $i$ para um ponto $j$ da rede será então realizada através de um destes anéis, o qual deverá ter equipamentos de transmissão de dados instalados tanto no ponto que envia os dados, como no ponto que os recebe. Observamos que cada pacote de dados deverá ser inteiramente transmitido em um mesmo anel, não podendo ser subdividido.

Um nó pode fazer parte de mais de um anel e consequentemente ter nele equipamentos diferentes, porém estes não se interligam, isto é, em um mesmo anel, todos os equipamentos de transmissão instalados deverão ser do mesmo tipo.

A capacidade máxima de transmissão do equipamento utilizado definirá a capacidade de transmissão do anel, a qual independe portanto, do número de equipamentos instalados no anel e cada conexão (fibra) também não poderá carregar mais do que a capacidade máxima do anel. Por exemplo, se nos pontos por onde passa um anel, foram instalados equipamentos do tipo ADM-1, então a soma de todas as demandas atendidas por este anel não deve ultrapassar 1008 Mbps, (ver tabela 2.1).

Por outro lado, o custo do anel é dependente do número de equipamentos que foi nele instalado e será dado pela soma dos custos destes equipamentos. Neste trabalho optamos por não incorporar o custo da fibra ao custo do anel (ver [20]).

Como exemplo, representamos uma rede de telecomunicações utilizando o grafo na figura 2.1. Neste existem três ciclos suportes como podemos ver na figura 2.2.

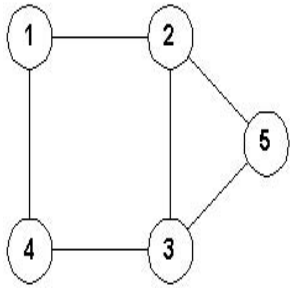


Figura 2.1: Exemplo de uma rede de telecomunicações.

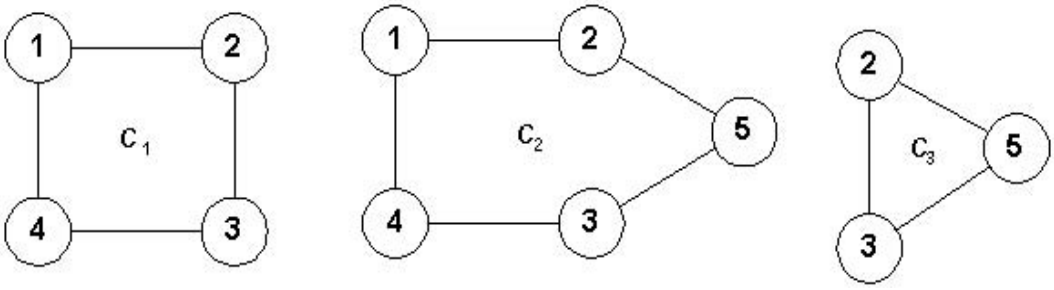


Figura 2.2: Exemplos de ciclos suporte.

Ao selecionarmos o ciclo suporte $C_1$ para instalação de equipamentos de transmissão de dados nos nós 1 e 3, este passará a ser chamado de anel, o qual chamaremos de anel $k_1$.

O anel $k_1$ poderá transmitir dados do ponto 1 para o 3 e vice-versa utilizando a capacidade do ADM instalado, figura 2.3.

Podemos ter várias demandas relacionadas a um mesmo par de nós e distribuí-las em um ou mais anéis. Para tratar casos como este, introduziremos um índice na lista de demandas a serem atendidas, com isto poderemos diferenciá-las. Por exemplo, se desejamos transmitir um fluxo de RJ para SP de 100 Mbps e outro fluxo também de RJ para SP de 1000 Mbps, então o primeiro fluxo terá índice 1 e o segundo terá índice 2.

### 2.1.1 Anéis Unidirecionais

Anéis Unidirecionais *são compostos pelo tráfego normal e o tráfego de proteção, onde o tráfego normal é transmitido em uma direção em uma fibra e o tráfego de proteção*

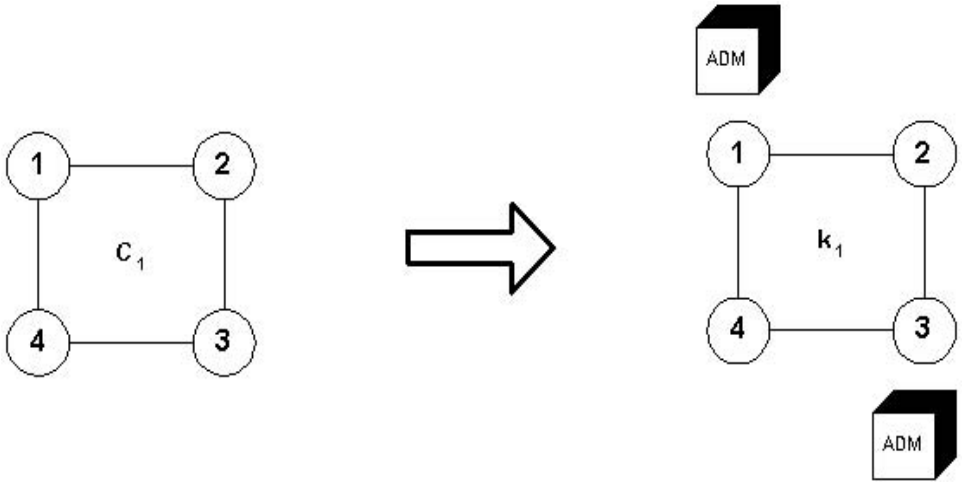


Figura 2.3: Exemplo de anel.

*é feito no sentido oposto em outra fibra*, garantindo assim que mesmo que haja um rompimento ao longo do caminho, os dados cheguem ao seu destino, figura 2.4 [2]. No anel unidirecional a capacidade é dada pelo fluxo máximo (somatório de todas as demandas no anel) que pode passar no anel e esta deverá ser menor ou igual a capacidade do equipamento instalado.

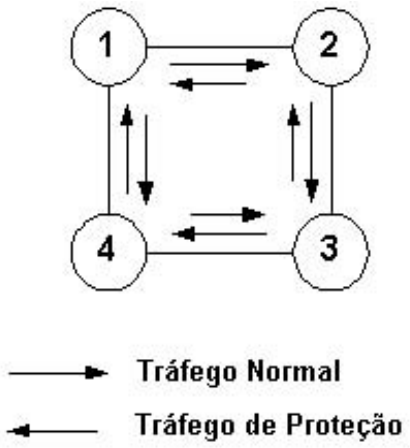


Figura 2.4: Anel Unidimencional - Tráfego.

## 2.2 Apresentação do problema

O problema de planejamento da expansão de anéis unidirecionais em rede de telecomunicação consiste em determinar os anéis que serão utilizados e, consequentemente os equipamentos de transmissão que serão instalados na rede, de forma a atender toda a demanda prevista a um custo mínimo de instalação.

Os dados de entrada para a resolução deste problema são:

- A previsão de demanda entre cada par de pontos da rede.
- Os tipos de equipamentos de transmissão disponíveis, bem como suas capacidades máximas de transmissão e seus custos.
- As interligações (fibra) existentes entre os pontos da rede.

As informações obtidas com a solução do problema são:

- Indicação dos anéis que serão utilizados na transmissão dos dados.
- Indicação dos equipamentos ADM's que serão instalados em cada anel e sua localização física, ou seja, em que nós da rede os equipamentos serão instalados.
- Os fluxos de dados circulantes em cada anel.
- O custo total de instalação.

Segue um exemplo que ilustra o problema e duas possíveis soluções.

Suponha que precisamos transmitir as demandas apresentadas na figura 2.5 na rede representada na figura 2.1. Disponibilizamos dos equipamentos descritos na tabela 2.1 com seus respectivos preços e capacidades.

Para a rede dada, definimos as três possíveis estruturas para construir os anéis representadas na figura 2.2.

Observemos que uma possível solução é dada pela instalação do ADM - 1 nos quatro pontos de $C_1$, que definem o primeiro anel ($k = 1$). Poderemos alocar $d_{13}$, $d_{23}$ e $d_{24}$ neste anel, pois a soma destas demandas é igual a capacidade do ADM utilizado, isto é,

| i | j | demanda |
|---|---|---|
| 1 | 3 | 320 |
| 2 | 3 | 300 |
| 2 | 4 | 388 |
| 2 | 5 | 480 |

Figura 2.5: Demandas.

$d_{13} + d_{23} + d_{24} = 1008$. A demanda $d_{25}$ seria atendida, por exemplo, em $C_3$, utilizando o ADM - 2, que possui capaciade de 4032, portanto $d_{25}$ é menor que a capacidade de ADM - 2. Ilustramos a solução na figura (2.6).

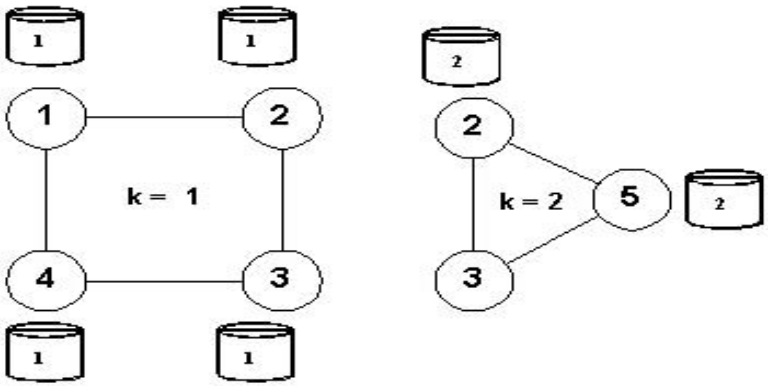


Figura 2.6: Solução 1

Neste caso teremos o custo da rede sendo quatro vezes o preço do ADM - 1, que deveria ser instalado nos nós 1, 2, 3 e 4, mais duas vezes o preço do ADM - 2, que deveria ser instalado nos nós 2, e 5, isto é, R\$ $1.9 \times 10^9$.

Outra solução possível seria utilizar o ADM - 2 em $C_2$, onde podemos alocar todas as demandas, pois a soma destas é menor que a capacidade do ADM utilizado, isto é, $d_{13} + d_{23} + d_{24} + d_{25} < 4032$. Ilustramos a solução na figura (2.7).

Neste caso teriamos o custo da rede sendo cinco vezes o preço do ADM - 2, que seria instalado em todos os nós do ciclo suporte $C_2$, isto é, R\$ $2.25 \times 10^9$

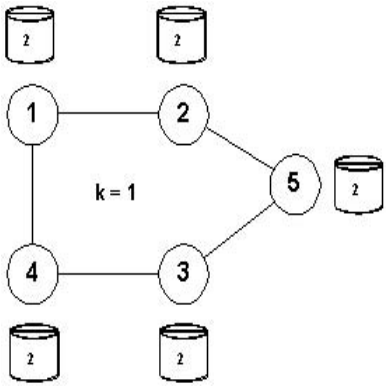


Figura 2.7: Solução 2

## 2.3 Modelo Matemático do Problema (PAURT)

Nesta seção apresentaremos um modelo de programação linear inteira mista para o problema de planejamento de anéis unidirecionais em rede de telecomunicações (PAURT), sua expansão ou mesmo a sua avaliação periódica.

Representaremos no modelo, a rede de transmissão de dados por um grafo não direcionado $G(V, E)$, onde o conjunto de vértices $V$, representa os pontos da rede de transmissão (cidades, centros, etc.) e o conjunto de arcos $E$ representa as ligações entre os pontos.

No modelo apresentado a seguir, consideramos todos os ciclos contidos no grafo $G$ (definimos grafo e ciclo de um grafo no apêndice A). Eles representam todas as possíveis estruturas para formação dos anéis por onde os dados serão transmitidos na rede considerada. A geração destes ciclos será estudada no apêndice C deste trabalho.

Sejam $C_1, ..., C_L$ todos os ciclos do grafo $G$, $C = \{1, 2, ..., L\}$ o conjunto dos índices correspondentes e $\mathcal{N}^l$ o conjunto de nós que pertencem ao ciclo $C_l$, onde $l \in C$.

Considere que:

$p_k$ é o preço de cada equipamento utilizado no anel $k$;

$M$ é uma constante, com valor suficientemente grande;

$d_{ij}$ é a demanda que passa do nó $i$ para o nó $j$;

$cap_k$ é a capacidade do anel $k$, onde a capacidade do anel $k$ é igual a capacidade do equipamento nele utilizado;

$NA^l$ é um limite superior para o número de anéis construidos sobre o ciclo $C_l$ do problema. Para isto analisamos o pior caso, onde cada anel atenderia apenas uma demanda, o que nos daria um limite superior para o número de anéis igual ao número de demandas. Porém como estamos trabalhando com dois tipos de equipamentos estas demandas poderão estar utilizando o equipamento 1 ou 2. Caso utilizem o equipamento 1 estaremos caracterizando os anéis de 1 ao número de demandas ($Ndem$) e caso seja utilizado o equipamento 2 estaremos caracterizando os anéis do número de demandas mais um ($Ndem + 1$) a duas vezes o número de demandas ($Ndem + Ndem$). Então $NA^l$ é igual ao número de demandas entre os pares de nós contidos no ciclo $C_l$ vezes o número de tipos de ADM's;

e que $e_{ikl}$ e $y_{ijkl}$ são variáveis binárias definidas da seguinte forma:

$$e_{ikl} = \begin{cases} 1 & \text{se existe equipamento no nó } i \text{ do anel } k \text{ construido no ciclo } l; \\ 0 & \text{caso contrário.} \end{cases}$$

$$y_{ijkl} = \begin{cases} 1 & \text{se } d_{ij} \text{ é atendida pelo anel } k \text{ construido no ciclo } l; \\ 0 & \text{caso contrário.} \end{cases}$$

Podemos assim representar o modelo matemático para o problema de planejamento de anéis unidirecionais em uma rede de telecomunicações como:

$$\text{Minimizar} \sum_{l \in \mathcal{C}} \sum_{i \in \mathcal{N}^l} \sum_{k=1}^{NA^l} p_k e_{ikl},$$

sujeito a:

$$\begin{aligned}
&\sum_{i,j \in \mathcal{N}^l, i \neq j} d_{ij} y_{ijkl} \leq cap_k, k = 1, \ldots, NA^l, l \in \mathcal{C}, \\
&\sum_{j \in \mathcal{N}^l \setminus i | d_{ij} \neq 0} y_{ijkl} \leq M e_{ikl}, k = 1, \ldots, NA^l, i \in \mathcal{N}^l, l \in \mathcal{C}, \\
&\sum_{i \in \mathcal{N}^l \setminus j | d_{ij} \neq 0} y_{ijkl} \leq M e_{jkl}, k = 1, \ldots, NA^l, j \in \mathcal{N}^l, l \in \mathcal{C}, \\
&\sum_{l \in \mathcal{C}} \sum_{k=1}^{NA^l} y_{ijkl} = 1, i, j \in \hat{\mathcal{N}^l} | i \neq j | d_{ij} \neq 0, \\
&e_{ikl}, y_{ijkl} \in \{0, 1\}, i, j \in \mathcal{N}^l | i \neq j, k = 1, \ldots, NA^l, l \in \mathcal{C}.
\end{aligned} \tag{2.1}$$

No problema (2.1) a **função objetivo** representa a minimização do custo total com a instalação dos equipamentos considerando todos os nós, todos os anéis e todos os ciclos.

A **primeira restrição** estabelece que a demanda total atendida no anel $k$ construido no ciclo $l$, não deve ser maior que sua capacidade. Por exemplo, se no anel $k = 1$ instalarmos um equipamento do tipo 1 que possui capacidade de 1000 Mbps, então o somatório das demanda que passará por aquele anel não deverá ultrapassar 1000 Mbps.

A **segunda restrição** estabelece que não haverá fluxo partindo de $i$, no anel $k$ construido no ciclo $l$, se não houver equipamento nele instalado, ou seja, se $e_{ikl} = 0$.

Observe que $M$ deve ser grande o suficiente para tornar esta restrição redundante quando $e_{ikl} = 1$. Desta forma, utilizamos $M = n$, onde $n$ é o número de nós da rede.

A **terceira restrição** estabelece que não haverá fluxo chegando no nó $j$, no anel $k$ construido no ciclo $l$, se não houver equipamento nele instalado.

A **quarta restrição** estabelece que a demanda $d_{ij}$ será atendida por exatamente um anel construido em um dos ciclos.

## 2.4 Considerações sobre o Modelo

Este modelo determina onde serão feitas as instalações dos ADM's, isto é, escolhe os anéis da rede de tal forma que as demandas sejam todas atendidas a um custo mínimo.

Além disto este modelo pode ser usado para avaliação periódica da rede de telecomunicações. Neste caso basta inicializar o procedimento proposto nesta dissertação com algumas variáveis $e_{ikl}$ fixas em um, ou seja, bastaria determinar onde já existe equipamento instalado. Este tipo de avaliação determinará onde será necessário instalar equipamentos de transmissão de tal forma que as demandas sejam atendidas porém, levando em conta os equipamentos que já estiverem instalados.

Para que isto seja feito precisamos, com base na rede, encontrar um algoritmo eficiente que gere os ciclos em um dado grafo C, além de uma heurística adequada ao modelo, o que será descrito no próximo capítulo.

Observa-se que para uma rede real, a qual possui um grande número de vértices, o número de ciclos cresce muito rápido como também o número de restrições, tornando-se uma grande dificuldade para a resolução computacional.

Por exemplo, o número de variáveis binárias do problema 2.1 é determinado pelo somatório de $e_{ikl}$ e $y_{ijkl}$. Para definir o número de variáveis $e_{ikl}$ e $y_{ijkl}$ deve-se multiplicar o valor máximo que os índice das variáveis podem assumir, isto é, para $e_{ikl}$ tomaremos o máximo de $i$, $j$ e $k$ e para $y_{ijkl}$ o máximo que $i$, $j$, $k$ e $l$ podem assumir. Então o número de variáveis binárias do problema 2.1 é dado por $n * NA * L + n * (n-1) * NA * L$.

Para determinar o número de restrições de um problema devemos considerar o máximo que os índices das restrições podem assumir. Por exemplo, no problema 2.1 os índices da primeira restrição $k$ e $l$, assumem valor máximo $NA$ e $L$ respectivamente, sendo assim teremos $NA * L$ restrições deste tipo. Na segunda e terceira restrição $k$, $i$ e $l$ assumem valor máximo $NA$ e o número de nós de cada ciclo $l$, sendo assim teremos $\sum_{l \in C} \mid N^l \mid * NA$ restrições deste tipo. Na quarta restrição teremos valor máximo de $i$ e $j$, sendo assim teremos respectivamente $n * (n-1)$ restrições deste tipo. Então o número total de restrições será dado por $NA * L + 2 * \sum_{l \in C} \mid N^l \mid * NA + n * (n-1)$.

Por exemplo, tomaremos o grafo $G$, apresentado na figura 2.1. Este possui 3 ciclos como podemos ver na figura 2.2. Consideremos 2 tipos de ADM's e tomemos as demandas a serem atendidas, apresentadas na figura (2.5).

Teremos $n = 5$, $L = 3$ e $NA = 8$. Então o número de variáveis binárias será 600 e o número de restrições será 236.

Se tomarmos 5 ADM's, teremos $NA = 20$, o número de variáveis binárias passará para 1500 e o número de restrições será 560.

Deste pequeno exemplo concluímos que a utilização de um método de otimização combinatória clássico, como por exemplo *branch and bound*, tornaria difícil a resolução de problemas reais quando as dimensões das instâncias são muito maiores do que o exemplo analisado. Isto nos leva a optar por uma metaheurística que nos forneça uma solução próxima a exata porém com um tempo computacional razoável.

# Capítulo 3

# Metaheurísticas

## 3.1 Introdução

Um problema de otimização combinatória pode ser definido como:

$$(P) \qquad \text{Minimizar } f(x), \qquad (3.1)$$
$$\text{sujeito a: } x \in X,$$

onde $f : X \rightarrow \mathbb{R}$ e uma função real e $X$ é um conjunto finito de soluções viáveis. Objetiva-se encontrar uma solução ótima para $(P)$, $x^* \in X$, tal que $f(x^*) \leq f(x)$, $\forall x \in X$ . Definimos $x^*$ como uma solução ótima global de $(P)$.

Teoricamente é possível encontrar uma solução ótima para $(P)$ utilizando enumeração, porém na prática a estratégia de enumeração completa se torna inviável, pois o número de combinações cresce exponencialmente com o tamanho do problema [13].

Alguns problemas podem ser muito grandes para o uso de algoritmos exatos, por exemplo alguns problemas ligados ao setor de telecomunicações. A estes problemas usualmente emprega-se métodos heurísticos.

## 3.2 Heurísticas

> Heurística, palavra derivada do grego "heuriskein" que significa encontrar ou descobrir. É definida como sendo qualquer método construído com base em uma idéia para solucionar um problema [22].

Estes métodos geralmente encontram boas soluções, isto é, próximas a solução exata e com um custo computacional razoável, porém não garantem encontrar a solução exata. São empregados nas seguintes situações:

- quando o algoritmo exato conhecido consome alto tempo computacional;
- como passo intermediário para aplicação de um outro algoritmo, acelerando a convergência para a solução exata, quando utilizado em conjunto com métodos exatos;
- quando deseja-se achar soluções viáveis para problemas cuja resolução é muito complexa, como por exemplo, problemas de otimização combinatória;
- quando o objetivo não é encontrar uma solução exata e sim uma solução satisfatória, por exemplo na solução de problemas em cujo modelo matemático foram realizadas simplificações e ou aproximações.

Uma característica encontrada em muitos problemas combinatoriais é a presença de soluções ótimas locais. Desta forma, é importante termos uma noção de vizinhaça de uma dada solução.

Definiremos como vizinhança $N(x, \sigma)$ de uma solução $x$, o conjunto de soluções que podem ser alcançadas a partir de $x$, por uma operação simples $\sigma$. Tal operação pode ser a remoção ou adição de um elemento a uma solução ou mesmo o intercâmbio de dois elementos em uma solução. Se uma solução $y$ não for pior do que qualquer outra solução em sua vizinhança $N(y, \sigma)$, então $y$ é definido como um ótimo local com respeito a esta vizinhança. [16].

Heurísticas em geral são eficientes quando aplicadas a problemas específicos, sendo pouco eficientes na solução de uma classe geral de problemas, pois leva em consideração características muito particulares do mesmo [16].

Heurística construtiva é um procedimento que seleciona sequencialmente elementos de um conjunto finito, com o objetivo de obter, ao final, uma solução viável, isto é, a solução final de uma heurística é construida de forma iterativa, um elemento de cada vez.

Tornar aleatória a seleção do elemento que é incluído na solução, fazendo com que a heurística construtiva, antes determinística, seja probabilística, tem por finalidade possibilitar a geração de soluções distintas (diferentes). O resultado disto é que a heurística construtiva aleatorizada tem o poder de evitar que o processo de busca fique preso a ótimos locais.

## 3.3 Metaheurísticas

As metaheurísticas são consideradas heurísticas de uso geral, que nos fornecem boas soluções.

As metaheurísticas mais conhecidas são:

1. Algoritmo Genético: baseado em uma analogia com a teoria da evolução biológica, utilizando princípios e termos comuns aos sistemas biológicos. Proposto por Holland e seus alunos na Universidade de Michigan em (1975) [21].

2. Busca Tabu: proposto inicialmente por *Fred Glover* e *Pierre Hansen* (1986). Tem como idéia principal uma busca inteligente utilizando como elemento básico uma memória flexível para guardar o conhecimento obtido na exploração do espaço de busca, não retornando a uma solução previamente visitada [21].

3. Simulated Annealing: proposta inicialmente por Metropolis (1953) para adaptar o processo de annealing, isto é, o processo de recozimento de sólidos e formação de cristais (mecânica estática) para solução de problemas combinatórios. Annealing refere-se a um processo térmico que se inicia pela liquefação de um cristal a uma alta temperatura, seguido por uma lenta e gradativa diminuição da temperatura, até o ponto de solidificação, definido como estado de energia mínima. Nos anos 80, Simulated Annealing foi proposta para solução de problemas combinatórios por

Kirkpatrick, em (1983) e Cerny em (1985). Ambos os trabalhos mostram uma correspondência entre o estado do sistema (annealing) e o espaço de possíveis soluções de um problema combinatório [21].

4. GRASP: composto por duas heurísticas, as quais descreveremos abaixo. Esta será a metaheurística aplicada na resolução do problema considerado nesta dissertação.

## 3.4 Metaheurística GRASP

Greedy Randomized Adaptive Search Procedures (GRASP), foi baseado nos trabalhos de Lin & Kernighan (1973) [7] e Hard & Shogan (1977) [8] e proposto por Thomas Feo e Maurício Resende no final dos anos 80 [3]. Consiste na combinação de uma heurística construtiva aleatorizada com um método de busca local em um procedimento iterativo cujas iterações são independentes [22].

### 3.4.1 Algoritmo GRASP Básico

O pseudo-código da figura 3.1 ilustra o algoritmo básico do GRASP para o problema de minimização 3.1, onde MaxIter é o número máximo de iterações a serem realizadas, *Sem* é a semente inicial para o gerador de números pseudos aleatórios e $\alpha$ é um número que pertence ao intervalo $[0, 1]$.

### 3.4.2 Fase de construção (Construção Randômica)

Tem como objetivo prover soluções viáveis distintas, produzidas utilizando-se um procedimento de construção, tal como uma heurística construtiva. As soluções obtidas serão exploradas na fase de busca local.

Na fase de construção, uma solução viável é iterativamente construida, inserindo-se na solução parcial um elemento de cada vez. A cada iteração da fase construtiva, são avaliados apenas elementos que podem ser adicionados à solução, isto é, não devem ser considerados os elementos que uma vez na solução fazem com que está seja inviável. Ao

```
Prodedimento GRASP (MaxIter, Sem, α)
  1  Leia Dados de Entrada ();
  2  Para k = 1, ...,MaxIter faça
  3  Início
  4    x̂ ← Construção Randômica ( Sem,α);
  5    x̃ ← Busca Local (x̂);
  6    x̄ ← Atualização da Solução(x̃);
  7  Fim;
  8  Retorna x̄
Fim GRASP;
```

Figura 3.1: pseudo código GRASP

conjunto destes elementos, aqueles que podem ser adicionados à solução, da-se o nome de conjunto $C$.

Para fazer a escolha de qual elemento será adicionado ao conjunto solução devemos montar uma lista com os melhores elementos que podem ser colocados naquele conjunto (elementos do conjunto $C$). Os elementos membros da *lista restrita de candidatos* (LRC) são escolhidos com base em uma função gulosa, isto é, uma função que "avalia" o benefício de cada elemento do conjunto $C$ fazer parte da solução.

A escolha do elemento que é adicionado ao conjunto solução é realizada de maneira aleatória entre os membros da LRC, isto é, todos os elementos desta lista tem a mesma probabilidade de serem considerados na solução que está sendo iterativamente construída nesta fase.

Selecionado um elemento da LRC para fazer parte da solução, sua viabilidade deve ser varificada frente às restrições do problema. Sendo a solução viável para todas as restrições, a fase de construção se encerra. No caso de a solução ser ainda inviável, um novo elemento deve ser adicionado. Para isto, devem ser reavaliados o conjunto $C$ e a LRC para considerar a nova solução.

A metaheurística GRASP é adaptativa porque os benefícios associados a cada um dos elementos do conjunto $C$ são atualizados a cada iteração da fase de construção para

refletir as mudanças trazidas pela seleção do elemento selecionado na iteração anterior desta fase.

O componente probabilístico do GRASP é caracterizado pela escolha aleatória de um dos elementos da lista de candidatos LRC. Note que em uma heurística puramente construtiva o elemento que sempre é selecionado é aquele com a melhor avaliação (função gulosa). É importante ressaltar que nesse caso o método é determinístico, ou seja, se reaplicado nas mesmas condições provém a mesma solução.

Para controlar a qualidade dos elementos da que são considerados para formar a LRC, considere $h_{max}$ e $h_{min}$ respectivamente o melhor e o pior valor da função gulosa correspondente a todos os elementos que podem ser candidatos a fazer parte da solução na fase de construção (por exemplo, podendo corresponder respectivamente ao menor e ao maior aumento na função objetivo) e um parâmetro $\alpha$ que assume valores entre zero e um ($0 \leq \alpha \leq 1$). O tamanho da LRC é definida em função do pâmetro $\alpha$, que analisa todos os elementos que pertencem a lista de candidatos, e que não fazem parte da solução, cujo valor da função gulosa associado, esteja no intervalo $[h_{min},\ \alpha(h_{max} - h_{min}) + h_{min}]$.

Observa-se que para um problema de minimização $\alpha = 0$ implica em uma escolha puramente gulosa, enquanto que para $\alpha = 1$, implica em uma escolha aleatória, isto é, a LRC contém todos os elementos possíveis.

O parâmetro $\alpha$ pode ser escolhido de diversas maneiras:

- fixo - selecionado pelo usuário, em geral, próximo ao guloso para garantir qualidade média, mas longe o suficiente para gerar diversidade;
- aleatório - selecionado de forma aleatória no intervalo [0,1] , discretizado ou contínuo;
- auto-ajustável (reativo) - o parâmetro $\alpha$ é ajustado em tempo de execução de acordo com informações de soluções visitadas anteriormente. Em cada iteração do algoritmo, o valor do parâmetro $\alpha$ é escolhido aleatoriamente. Porém, ao invés de escolher esse parâmetro de uma distribuição uniforme, ele é escolhido a partir de um conjunto de valores discretos $\{\alpha_1, \alpha_2, ..., \alpha_m\}$. A cada $\alpha_k$ associa-se uma probabilidade de escolha $p(\alpha_k)$ . Essas probabilidades são inicialmente idênticas. O procedimento

**Procedimento** `Construção Randômica` (`sem`, $\alpha$)

1 $\hat{x} = \emptyset$;
2 `Inicializa o conjunto` $C$;
3 **Enquanto** $C \neq \emptyset$ **faça**
4 **Início**
5 $h_{min} = min\{g(t)|t \in C\}$;
6 $h_{max} = max\{g(t)|t \in C\}$;
7 `LRC` $= \{s \in C \mid g(s) \leq h_{min} + \alpha(h_{max} - h_{min})\}$;
8 `Seleciona s, aleatoriamente da LRC`;
9 $\hat{x} = \hat{x} \bigcup \{s\}$;
10 `Atualize o conjunto candidato` $C$;
11 **Fim**;
12 **Retorna** $\hat{x}$

**Fim** `Construção Randômica`;

Figura 3.2: pseudo código Construção Randômica

atualiza em tempo de execução as probabilidades $\{p(\alpha_1), p(\alpha_2), ..., p(\alpha_m)\}$para favorecer valores de $\alpha$ que tenham produzido soluções de qualidade. Maiores detalhes podem ser visto em [1],[14].

### 3.4.3 Algoritmo de Construção Básico

O pseudo-código da figura 3.2 descreve a fase de construção GRASP para um problema de minimização, onde $g(\dot{})$ é a função gulosa, $\alpha$ é o parâmetro responsável pelo tamanho da LRC e $C$ o conjunto de elementos candidatos.

### 3.4.4 Função Tendência

Na fase de construção do GRASP básico o próximo elemento a ser inserido na solução é selecionado aleatoriamente entre os elementos candidatos na LRC. Cada elemento da LRC possui a mesma probabilidade de ser selecionado.

Bresina propõe em [14] uma seleção do elemento da LRC, onde a cada elemento da

Tabela 3.1: Opções para Função Tendência

| | | |
|---|---|---|
| 1 | Logarítmica | $bias(r) = 1/log(r+1)$ |
| 2 | Linear | $bias(r) = 1/r$ |
| 3 | Polinomial | $bias(r) = 1/r^n$ |
| 4 | Exponencial | $bias(r) = 1/e^r$ |
| 5 | Randômica | $bias(r) = 1$ |

lista de canditatos restrita é atribuída uma probabilidade de seleção. Como podemos observar na tabela 3.1, uma família de distribuição de probabilidade é proposta.

Primeiro, o valor da função gulosa de cada elemento $c \in C$ é calculado, uma vez que é necessário eleger os membros de $C$ que farão parte da lista LRC. Daí, os elementos da LRC devem ser ordenados de acordo com o valor da função gulosa. Seja $rank(c), c \in C$ a posição relativa do elemento $c$ na lista LRC de acordo com a ordenação dada pela função gulosa. A cada elemento da LRC é associado um valor, $bias(rank(c))$, onde $bias(\dot{})$ é calculado de acordo com a função de tendência, cujas expressões propostas no trabalho de Bresina estão colocadas na tabela 3.1. A probabilidade de seleção de um elemento $c$ da lista de candidatos é então determinada por:

$$p(c) = \frac{bias(rank(c))}{\sum_{\sigma \in \mathrm{LRC}} bias(rank(\sigma))}, c \in \mathrm{LRC} \tag{3.2}$$

### 3.4.5 Fase de Busca Local

Geralmente é possível melhorar a solução construida na fase de construção do GRASP, aplicando-se a ela uma busca local.

Um procedimento de busca local parte sempre de uma solução inicial $x^0$ e gera uma seqüência de soluções $x^1$, $x^2$, ..., $x^k$, isto é, dada uma solução $x^0$, os elementos da sua vizinhança $N(x^0)$ são soluções que podem ser obtidas através da aplicação de uma modificação elementar em $x^0$, chamada de movimento. Por exemplo, se $x^0 = (1, 2, 3)$ e $N_1(x^0)$ é a vizinhança gerada por movimentos de permutação de dois elementos, obteremos como

```
Procedimento Busca Local (x̂)
  1  Enquanto existe x ∈ N(x) tal que f(x) < f(x̂)  faça
  2  Início
  3    Seleciona x ∈ N(x) tal que f(x) < f(x̂);
  4    x̂ = x;
  5    Busca Local (x̂) ;
  6  Fim;
  7  Retorna (x̂);
Fim Busca Local;
```

Figura 3.3: pseudo código Busca Local

vizinhança de $x^0$ o conjunto $N(x^0) = \{(2,1,3),(1,3,2),(3,2,1)\}$.

Em um problema de minimização, a vizinhança $N(x^k)$ de $x^k$ é analisada na k-ésima iteração e procura-se encontrar uma solução $x^{k+1} \in N(x^k)$, tal que $f(x^{k+1}) < f(x^k)$. Quando uma solução melhora a solução corrente, ela passa a ser a nova solução e a sua vizinhança é analisada. Caso contrário , a busca termina e a solução corrente é um ótimo local.

Logo, a vizinhança $N$ para o problema $P$ em relação a solução $x^k$, define um subconjunto de soluções $N(x^k)$, onde $x^k$ é dito um ótimo local, se não exite solução melhor que $x^k$ em $N(x^k)$.

Uma característica especial do GRASP é a facilidade na qual pode ser implementado. Basicamente necessita-se configurar e sintonizar o tamanho da lista de candidatos ($\alpha$), o número de iterações e a função gulosa. Sua idéia básica consiste em usar diferentes soluções iniciais como pontos de partida para a busca local.

### 3.4.6 Conclusões

As principais características relacionadas a metaheurística GRASP são:

- existem poucos parâmetros a serem ajustados: limitação da lista de candidatos ($\alpha$), número de iterações e a função gulosa;

- uma função gulosa deve ser definida de acordo com as características do problema;
- a vizinhaça da solução deve ser definida, isto é, o mecanismo de movimentos que será utilizado na fase de busca local.

# Capítulo 4

# Aplicação de GRASP ao PAURT

Neste capítulo estudaremos a aplicação da metaheurística GRASP ao problema de planejamento de anéis unidirecionais em redes de telecomunicações, procedimento este que denominaremos GRASP - PAURT. Apresentaremos os procedimentos utilizados em cada fase do GRASP para obter-se uma solução para problema.

Como foi feito no capítulo 2, representaremos a rede de telecomunicações por um grafo não direcionado $G = (\mathcal{N}, E)$, onde cada nó $i$ representa uma central da rede e cada aresta do grafo representa uma conexão da rede (cabo de fibra ótica).

A aplicação da metaheurística GRASP ao PAURT proposta nesta tese faz da fase de construção um procedimento construtivo de geração de anéis e alocação de equipamentos de transmissão de dados da rede. Este procedimento é dividido em duas etapas (duas sub-fases). A primeira consiste em selecionar um ciclo de suporte (sub-fase Construção Ciclo) para os anéis e a segunda consiste em definir os anéis (sub-fase Construção Anéis), ou seja, estabelecer *onde* (nós da rede) e *quais* equipamentos de transmissão de dados (tipo de ADM) devem ser instalados, de forma a atender todas as demandas por transmissão de dados entre quaisquer pares de nós que pertençam ao cliclo selecionado.

A seguir detalharemos os principais procedimentos do algoritmo, cujo pseudo-código pode ser visto na figura 4.1, onde MaxIter é o número de iterações do GRASP-PAURT, $Sem1$ e $Sem2$ são as sementes utilizadas pelo gerador de números pseudo-aleatórios em cada uma das sub-fases da fase de construção e, $\alpha_1$, $\alpha_2$ são os parâmetros que determinarão o tamanho das LRCs em cada fase de construção.

```
Procedimento GRASP-PAURT ( MaxIter, Sem1,Sem2, α1, α2)
 1  Leia Dados de Entrada ();
 2  Para k = 1, ..., MaxIter faça
 3  Início
 4    Enquanto todas as demandas não forem atendidas  faça
 5    Início
 6      C_l̂ ← Sub-fase Construção Ciclo (Sem1,α1);
 7      x̂ ← Sub-fase Construção Anéis (Sem2,α2, C_l̂);
 8    Fim
 9    x̃ ← Busca Local (x̂);
10   x̄ ← Atualização da Solução(x̃);
11  Fim;
12  Retorna x̄
Fim GRASP-PAURT;
```

Figura 4.1: pseudo código GRASP-PAURT

Após a leitura dos dados de entrada executa-se a fase de construção (sub-fase Construção Ciclo e sub-fase Construção Anéis) até que tenhamos atendido todas as demandas da rede, ou seja, até que tenhamos uma solução viável para o problema de planejamento de anéis unidirecionais em uma rede de telecomunicações. Em seguida executa-se a busca local onde o objetivo é obter uma solução que seja um mínimo local, dada uma vizinhança.

A cada iteração do procedimento GRASP (MaxIter iterações) uma nova solução para o problema PAURT é produzida. Esta solução deve ser comparada a melhor solução encontrada nas iterações anteriores, atualizando esta solução caso seja melhor do que ela.

## 4.1 O procedimento Leia Dados de Entrada

Este procedimento consiste em ler os dados necessários para que seja possível executarmos o GRASP-PAURT. Estes dados são:

- O número de nós ($n$) da rede;
- O grafo $G = (N, E)$ que representa a rede de telecomunicações;
- O conjunto de demandas $D = \{d_{ij} \mid i = 1, ..., n,\ j = 1, ..., n, i \neq j\}$ que deverão ser atendidas entre cada par de nós da rede;
- O conjunto de ciclos, $\{C_1, \ldots, C_L\}$ que o grafo $G$ possui, gerados previamente (Apêndice C);
- Os tipos de equipamentos de transmissão disponíveis, bem como suas capacidades máximas de transmissão e seus custos;

## 4.2 Sub-fase Construção Ciclo

O procedimento *Construção Ciclo* que podemos ver na figura 4.2, descreve a primeira sub-fase de construção do GRASP-PAURT. Como já mencionado, o objetivo desta sub-fase é a construção de um ciclo suporte, o qual será utilizado na construção de um ou mais anéis na sub-fase Construção Anéis.

Cada vez que executarmos o procedimento Construção Ciclo, o conjunto de candidatos $C$ é inicializado com o conjunto de todos os ciclos que contém pares de nós entre os quais ainda existe demanda a ser atendida.

Como o objetivo do GRASP-PAURT é a construção de anéis para atender todas as demandas da rede tomaremos como função "gulosa" ($g(.)$) do procedimento construtivo da sub-fase Construção Ciclo, o número de demandas a serem atendidas entre pares de nós que pertencem ao ciclo $C_{\hat{l}}$.

A função gulosa é utilizada para determinar a lista restrita de candidatos (LRC), para isto definimos $h_{max}$ e $h_{min}$ como o melhor e o pior valor da função gulosa associados aos elementos do conjunto de candidatos $C$, $\alpha_1$ é o parâmetro responsável pelo tamanho da LRC, assumindo valor fixo ($0 \leq \alpha_1 \leq 1$), conforme apresentado na figura 4.2.

Cada vez que executarmos a sub-fase Construção Ciclo adicionamos à solução um novo elemento (o ciclo $C_{\hat{l}}$), o qual é selecionado dentre os elementos da LRC, com base

**Procedimento** `Construção Ciclo` ( `sem1`, $\alpha_1$, $Ft$ )

1 `Inicializa o conjunto de Candidatos` $C$ ;

2 $h_{min} = min\ \{\, g(t) | t \in C \,\}$ ;

3 $h_{max} = max\ \{\, g(t) | t \in C \,\}$ ;

4 $LRC = \{s \in C | g(s) \geq h_{min} + \alpha_1(h_{max} - h_{min})\}$ ;

5 $C_{\hat{l}} \leftarrow$ `Seleção` $s$ `da LRC (sem1,` $Ft$`);`

6 `Retorna` $C_{\hat{l}}$;

**Fim** `Construção Ciclo`;

Figura 4.2: pseudo código da sub-fase Construção Ciclo

na função tendência, isto é, de acordo com a função tendência escolhida, cada elemento da LRC receberá uma probabilidade de ser selecionado.

A medida que escolhemos um ciclo $C_{\hat{l}}$, todas as demandas associadas a este ciclo serão satisfeitas na sub-fase Construção Anéis, executado em seguida. Sendo assim, na próxima vez que a sub-fase Construção Ciclo for executada, o número de demandas a serem atendidas associadas a $C_{\hat{l}}$ assumirá valor zero, e este não entrará no conjunto de elementos candidatos. Desta forma, após a execução da sub-fase Construção Anéis haverá uma reavaliação do número de demandas a serem atendidas, associadas a cada ciclo.

Portanto a cada iteração desta sub-fase de construção executamos dois procedimentos:

1. Construção da LRC, a qual contém um conjunto reduzido de elementos (ciclos) candidatos a pertencer a solução. Este conjunto é definido fazendo-se uso da função gulosa (número de demandas a serem atendidas entre pares de nós que pertencem ao ciclo $C_{\hat{l}}$)e o parâmetro de restrição $\alpha_1$.

2. Escolha do elemento (um ciclo) da LRC com base na função tendência e inclusão do elemento na solução.

A seleção do ciclo suporte torna-se puramente gulosa quando assumimos $\alpha_1 = 1$, isto é, garantimos que a LRC conterá apenas os ciclos que atendem ao maior número de demandas. Escolher ciclos do grafo que suportam o maior número de demandas ainda

não atendidas tende a minimizar o custo final da rede. Caso $\alpha_1 = 0$ obteremos uma LRC com todos os elementos possíveis. Chamamos atenção para o fato que excluimos da LRC apenas os ciclos que possuem o número de demanda associado igual a zero.

Para ilustrar melhor este procedimento tomaremos como exemplo o grafo $G$, o qual representará a rede de telecomunicações, figura 4.3. Esta rede contém três ciclos suportes como podemos observar na figura 4.4.

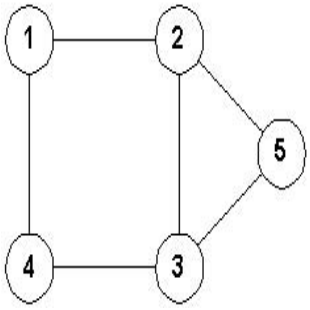


Figura 4.3: Grafo G

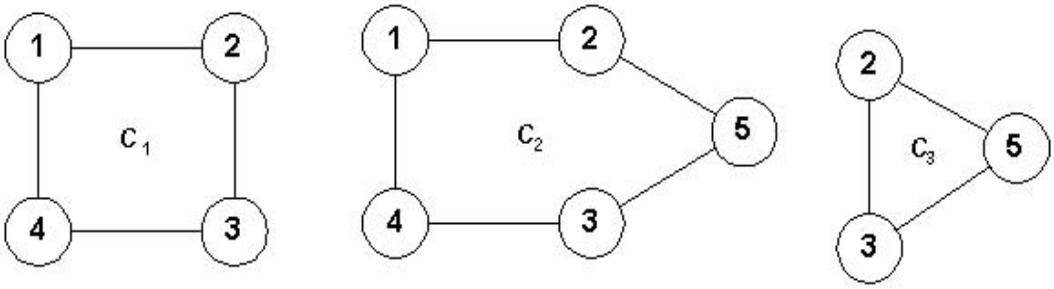


Figura 4.4: Exemplo Rede

Vamos supor disponíveis 2 tipos de ADM's, os quais podemos observar suas capacidades e preços na tabela 4.1.

Representamos o conjunto de demandas da rede ($D$) a ser atendido na figura 4.5 e o número de demandas a ser atendido, associado a cada ciclo suporte (função "gulosa" $g(.)$) na figura 4.6.

Tabela 4.1: ADM's

| ADM | Capacidade | Preço ($10^6$ R$) |
|---|---|---|
| ADM - 1 | 1008 Mbps | 250,00 |
| ADM - 2 | 4032 Mbps | 450,00 |

| i | j | demanda |
|---|---|---|
| 1 | 3 | 320 |
| 2 | 3 | 300 |
| 2 | 4 | 388 |
| 2 | 5 | 480 |

Figura 4.5: Tabela de demandas da rede

Neste exemplo, tomaremos $\alpha_1 = 0$. Com base na figura 4.6 determinamos o pior ($h_{min} = 2$) e o melhor ($h_{max} = 4$) elemento .

A LRC será montada de acordo com o procedimento *Construção Ciclo* utilizando $LRC = \{s \in C | g(s) \geq h_{min} + \alpha_1(h_{max} - h_{min})\}$. Neste exemplo, obtemos $LRC = \{C_1, C_2, C_3\}$.

Utilizando como *função tendência* a função radômica (GRASP padrão), isto é, para cada $C_i \in LRC$, teremos $bias(i) = 1$.

Para calcularmos a probabilidade associada a cada $C_i \in LRC$ utilizaremos:

$$p(C_i) = \frac{bias(C_i)}{\sum_{\sigma \in LRC} bias(\sigma)}, \ \forall C_i \in LRC,$$

para obter $p(1) = 0,333$, $p(2) = 0,333$ e $p(3) = 0,333$.

Dividimos o intervalo $[0, 1]$, nos sub-intervalos $I_1 = [0, p(1)]$, $I_2 = [p(1), p(1) + p(2)]$ e $I_3 = [p(1)+p(2), 1]$. Utilizando um gerador de número pseudo-aleatório (que utiliza como semente inicial $Sem1$) obtemos $v$, $v \in [0, 1]$. Dependendo de qual dos três sub-intervalos contém $v$, selecionaremos o novo elemento para incluir na solução do problema, isto é, obteremos o ciclo suporte para utilizarmos no procedimento *Construção Anéis*.

| l | Ciclos | $N^0$ de dem | | P( $C_l$ ) |
|---|---|---|---|---|
| 1 | 1-2-3-4 | 3 | → | 0,33 |
| 2 | 2-3-5 | 2 | → | 0,33 |
| 3 | 1-2-5-3-4 | 4 | → | 0,33 |

Figura 4.6: Estrutura Ciclo - Demandas

Suponha, por exemplo, que $v = 0,2$. Quando comparamos os três intervalos estaremos selecionando o segundo elemento da LRC, ou seja, o ciclo $C_1$, como podemos observar na figura 4.7. Com isto passamos a sub-fase Construção Anéis.

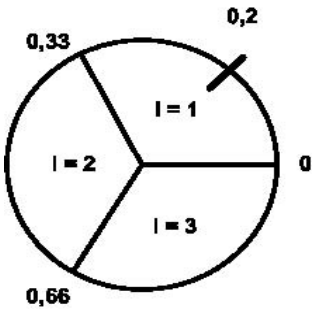


Figura 4.7: Estatística

## 4.3 Sub-fase Construção Anéis

O procedimento *Construção Anéis* que podemos ver na figura 4.8, descreve a segunda sub-fase de construção do GRASP-PAURT, a qual definirá os anéis a serem construidos sobre o ciclo suporte ($C_{\hat{l}}$), selecionado na sub-fase Construção Ciclo, ou seja, determinará quais ADM's serão alocados em cada nó deste ciclo.

Nesta seção, explicaremos cada passo do algoritmo apresentado na figura 4.8. Cada vez que executarmos o procedimento *Construção Anéis*, $D_{\hat{l}} = \{d_{ij} \in C_{\hat{l}} | d_{ij} \neq 0\}$ será inicializado com o subconjunto das demandas associadas ao ciclo $C_{\hat{l}}$, selecionado na sub-fase Construção Ciclo, que ainda não foram atendidas pela solução em construção.

**Procedimento** Construção Anéis ( sem1, $\alpha_1$, $C_{\hat{l}}$, $Ft$ )

1 $D_{\hat{l}} = \{d_{ij} \in C_{\hat{l}} | d_{ij} \neq 0\}$;
2 $\Omega = \{e_{ik\hat{l}}$ e $(e_{ik\hat{l}}, e_{jk\hat{l}})$ $\forall$ $i$, $j \in N^{\hat{l}}$ e $k = 1, .., NA^{\hat{l}}$ $\}$ ;
3 $\hat{X} = \emptyset$;
4 **Enquanto** $D_{\hat{l}} \neq \emptyset$ **faça**
5 **Início**
6 **Para** $\forall \omega \in \Omega$ **faça**
7 **Início**;
8 Fixar variáveis $e_{ik\hat{l}}$ e $y_{ijk\hat{l}}$ considerando $\hat{X}$ e t;
9 Resolver o problema (4.3) ;
10 **Fim**;
11 $h_{min} = min$ $\{ \Delta Z(t) | t \in E$ e $\Delta Z(t) > 0\}$ ;
12 $h_{max} = max$ $\{ \Delta Z(t) | t \in E$ e $\Delta Z(t) > 0\}$ ;
13 $LRC = \{s \in E | \Delta Z(s) \geq h_{min} + \alpha_2 (h_{max} - h_{min})\}$ ;
14 Ordenar LRC segundo o valor de $\Delta Z(t)$ ;
15 Selecione $s$ da LRC utilizando a função tendência ;
16 $\hat{X} = \hat{X} \bigcup \{s\}$ ;
17 $D_{\hat{l}} = D_{\hat{l}} - \{$ demandas já atendidas $\}$ ;
18 Atualização do conjunto $\Omega$ ;
19 **Fim**;
20 Retorna $\hat{X}$;

**Fim** Construção Anéis;

Figura 4.8: pseudo código da sub-fase Construção Anéis

Quando fixamos o ciclo suporte, escolhido pelo procedimento Construção Ciclo, o problema (2.1) transforma-se no problema (4.1),

$$\text{Minimizar } Z = \sum_{i \in \mathcal{N}^{\hat{l}}} \sum_{k=1}^{NA^{\hat{l}}} p_k e_{ik\hat{l}},$$

sujeito a:

$$\begin{aligned}
&\sum_{i,j \in \mathcal{N}^{\hat{l}}, i \neq j} d_{ij} y_{ijk\hat{l}} \leq cap_k, k = 1, \ldots, NA^{\hat{l}}, \\
&\sum_{j \in \mathcal{N}^{\hat{l}} \setminus i | d_{ij} \neq 0} y_{ijk\hat{l}} \leq M e_{ik\hat{l}}, k = 1, \ldots, NA^{\hat{l}}, i \in \mathcal{N}^{\hat{l}}, \\
&\sum_{i \in \mathcal{N}^{\hat{l}} \setminus j | d_{ij} \neq 0} y_{ijk\hat{l}} \leq M e_{jk\hat{l}}, k = 1, \ldots, NA^{\hat{l}}, j \in \mathcal{N}^{\hat{l}}, \\
&\sum_{k=1}^{NA^{\hat{l}}} y_{ijk\hat{l}} = 1, i, j \in \mathcal{N}^{\hat{l}}, i \neq j \mid d_{ij} \neq 0, \\
&e_{ik\hat{l}}, y_{ijk\hat{l}} \in \{0, 1\}, i, j \in \mathcal{N}^{\hat{l}} \mid i \neq j, k = 1, \ldots, NA^{\hat{l}},
\end{aligned} \tag{4.1}$$

onde um único ciclo de suporte é considerado (somente o ciclo $C_{\hat{l}}$) assim como somente as demandas que estejam associadas a este ciclo.

Seja $\Omega$ o conjunto de candidatos a serem incorporados à solução. Inicialmente $\Omega$ é inicializado com o conjunto de todas as variáveis $e_{ik\hat{l}}$ e dos pares de variáveis ($e_{ik\hat{l}}$,$e_{jk\hat{l}}$) para todo $i$, $j \in N^{\hat{l}}$, $i \neq j$, e $k = 1, .., NA^{\hat{l}}$.

A seleção de um elemento de $\Omega$ ($e_{ik\hat{l}}$ ou $(e_{ik\hat{l}}, e_{jk\hat{l}})$) representa a escolha desta variável (ou destas variáveis) para ser(em) fixada(s) em 1, ou seja, representa a escolha do nó $i$ (ou dos nós $i$ e $j$) para alocarmos nele(s) um ADM que pertencerá(ão) ao anel $k$.

Seja $\hat{X}$ o conjunto das variáveis $e_{ik\hat{l}}$ já fixadas até uma dada iteração da Sub-fase Construção Anéis.

Para decidir qual elemento de $\Omega$ será adicionado ao conjunto $\hat{X}$, propomos avaliar cada elemento de $\Omega$ pelo benefício que sua inclusão em $\hat{X}$ traz ao sistema. Este benefício pode ser calculado da seguinte forma: seja $\omega$ um elemento do conjunto $\Omega$. A idéia para calcular este benefício é resolver o problema (4.1) para $\{\omega \cup \hat{X}\}$.

Para isto, seja $\bar{e}_{ik\hat{l}}$ tal que:

$$\bar{e}_{ik\hat{l}} = \begin{cases} 1 & \text{se } e_{ik\hat{l}} \in \{\omega \cup \hat{X}\} \\ 0 & \text{caso contrário.} \end{cases}$$

Substituindo $\bar{e}_{ik\hat{l}}$ no problema (4.1), obtemos:

$$\text{Minimizar } Y = -w \sum_{k=1}^{NA^{\hat{l}}} \sum_{i=1}^{n} \sum_{j=1}^{n} y_{ijk\hat{l}} d_{ij},$$

sujeito a:

$$\begin{aligned}
&\sum_{i,j \in \mathcal{N}^{\hat{l}}, i \neq j} d_{ij} y_{ijk\hat{l}} \leq cap_k, k = 1, \ldots, NA^{\hat{l}}, \\
&\sum_{j \in \mathcal{N}^{\hat{l}} \setminus i | d_{ij} \neq 0} y_{ijk\hat{l}} \leq M \bar{e}_{ik\hat{l}}, k = 1, \ldots, NA^{\hat{l}}, i \in \mathcal{N}^{\hat{l}}, \\
&\sum_{i \in \mathcal{N}^{\hat{l}} \setminus j | d_{ij} \neq 0} y_{ijk\hat{l}} \leq M \bar{e}_{ik\hat{l}}, k = 1, \ldots, NA^{\hat{l}}, j \in \mathcal{N}^{\hat{l}}, \\
&\sum_{k=1}^{NA^{\hat{l}}} y_{ijk\hat{l}} \leq 1, i, j \in \mathcal{N}^{\hat{l}}, i \neq j \mid d_{ij} \neq 0, \\
&y_{ijk\hat{l}} \in \{0, 1\}, i, j \in \mathcal{N}^{\hat{l}} \mid i \neq j, k = 1, \ldots, NA^{\hat{l}}.
\end{aligned} \tag{4.2}$$

O parâmetro $w$ deve ser grande o suficiente para que $Z(\hat{X} \bigcup \{t\})$ seja negativo sempre existir pelo menos uma demanda associada ao ciclo de suporte $C_{\hat{l}}$ ainda não atendida.

Desta forma, o benefício de incorporar $\omega$ à solução em construção $\hat{X}$ pode ser medido pela diferença

$$\Delta Z(t) = Z(\hat{X}) - Z(\hat{X} \cup \{t\}),$$

onde $Z(X)$ equivale a $\sum_{i \in \mathcal{N}^{\hat{l}}} \sum_{k=1}^{NA^{\hat{l}}} p_k e_{ik\hat{l}} - Y$, onde Y é o valor da solução ótima do problema (4.2).

O procedimento *Construção Anéis* tem por objetivo construir iterativamente uma solução viável, logo a cada iteração um elemento do conjunto $\Omega$ é incorporado ao conjunto $\hat{X}$ (que é inicialmente um conjunto vazio) até que $\hat{X}$ seja uma solução viável para o problema (4.1).

Observe que no problema (4.2) a quarta restrição foi modificada de uma restrição de igualdade para uma restrição de desigualdade ($\leq 1$). Isto se deve ao fato que no problema

(4.2) não podemos mais garantir que todas as demandas pertencentes ao ciclo $C_{\hat{l}}$ sejam atendidas pois estamos fixando os nós que recebem equipamentos (fixamos as variáveis $e_{ik\hat{l}} = 1$ ou $e_{ik\hat{l}} = 0$). Para encorajar o atendimento as demandas, em (4.2), foi incluído um termo na função objetivo que penaliza o não atendimento as demandas.

Note ainda que podemos preprocessar este problema, reduzindo assim a complexidade de sua solução. Retirando as variáveis $y$ que estão fixas em zero e, consequentemente, retirando tambem todas as restrições $\sum_{j\in\mathcal{N}^{\hat{l}}\setminus i|d_{ij}\neq 0} y_{ijk\hat{l}} \leq Me_{ik\hat{l}}$ e $\sum_{i\in\mathcal{N}^{\hat{l}}\setminus j|d_{ij}\neq 0} y_{ijk\hat{l}} \leq Me_{jk\hat{l}}$, pois elas têm como finalidade fixar em zero as variáveis $y_{ijk\hat{l}}$ quando $e_{ik\hat{l}}$ assumir valor zero. Estas restrições já foram atendidas, portanto, ao fixarmos um subconjunto das variáveis $y_{ijk\hat{l}}$ em zero. Obtemos o problema reduzido representado a seguir:

$$\text{Maximizar } \sum_{k=1}^{NA^{\hat{l}}}\sum_{i=1}^{n}\sum_{j=1}^{n} y_{ijk\hat{l}}d_{ij},$$

sujeito a:

$$\begin{aligned}
&\sum_{i,j\in\mathcal{N}^{\hat{l}},i\neq j} d_{ij}y_{ijk\hat{l}} \leq cap_k, k = 1,\ldots,NA^{\hat{l}},\\
&\sum_{k=1}^{NA^{\hat{l}}} y_{ijk\hat{l}} \leq 1, i,j \in \mathcal{N}^{\hat{l}}, i\neq j \mid d_{ij}\neq 0,\\
&y_{ijk\hat{l}} \in \{0,1\}, \text{se } e_{ik\hat{l}} = 1, i,j \in \mathcal{N}^{\hat{l}} \mid i\neq j, k = 1,\ldots,NA^{\hat{l}},\\
&y_{ijk\hat{l}} = 0, \text{se } e_{ik\hat{l}} = 0, i,j \in \mathcal{N}^{\hat{l}} \mid i\neq j, k = 1,\ldots,NA^{\hat{l}}.
\end{aligned} \tag{4.3}$$

Finalmente observe que a função objetivo do problema (4.3) modifica-se para maximizar a utilização do atendimento a demanda, dado uma proposta de solução $\hat{X}$. Maximizar esta função objetivo é equivalente a minimizar a função objetivo do problema (4.2) pois o termo de penalização é o incentivo para a maximização do atendimento a demanda.

Então se $\Delta Z(t) \leq 0$, ou seja, $Z(\hat{X}\cup\{t\}) \geq Z(\hat{X})$, podemos concluir que a incorporação de $t$ ao conjunto $\hat{X}$ não possibilitou o atendimento de nenhuma demanda a mais. Sendo assim, o elemento $t$, tal que $\Delta Z(t) \leq 0$, não será considerado para a formação da lista restrita de candidatos, LRC. Observemos que quanto mais positiva for a diferença $\Delta Z(t)$, maior o benefício de incorporar os equipamentos, analisados no momento, a solução.

Definimos então a LRC para a sub-fase Construção Anéis, conforme apresentado na figura 4.8, onde $h_{max}$ e $h_{min}$ são o melhor e o pior valor da função gulosa para este procedimento, ou seja, o maior e o menor valor de $\Delta Z$.

O parâmetro $\alpha_2$ será responsável pelo tamanho da LRC, assumindo valor fixo $(0 \leq \alpha_2 \leq 1)$.

Após selecionarmos um dos elementos da LRC para ser incorporado ao conjunto $\hat{X}$, o conjunto $\Omega$ será atualizado. Dele serão excluídos os elementos formados pelas variáveis que já estão fixas em 1 e os elementos formados por pares de variáveis, nos quais uma delas já está fixa em 1. O procedimento de exclusão é necessário pois não podemos criar novamente um anel já criado, como também nao podemos colocar novos equipamentos nos nos dos anéis criados, onde já exite equipamento.

A cada iteração da sub-fase Construção Anéis adicionamos então à solução, um novo elemento que pode ser uma ou duas variáveis $e_{ik\hat{l}}$ fixas em 1. O procedimento será repetido até que seja obtida uma solução onde todas as demandas do ciclo $C_{\hat{l}}$ tenham sido atendidas.

Portanto a cada iteração da sub-fase de construção realizamos dois procedimentos:

1. Construção da LRC, a qual contém um conjunto reduzido de elementos candidatos a pertencer a solução. Este conjunto é definida fazendo uso da função gulosa e o parâmetro de restrição $\alpha_2$.

2. Escolha do elemento da LRC com base na função tendência e inclusão do elemento na solução $\hat{X}$.

Conclui-se que o procedimento *Construção Anéis* tem como objetivo a seleção dos melhores elementos (equipamentos), isto é, os que atendem as demandas com o menor custo possível. A seleção torna-se puramente gulosa quando assumimos $\alpha_2 = 1$, e caso tenhamos $\alpha_2 = 0$ obteremos uma LRC com todos os elementos possíveis.

Para ilustrar melhor o procedimento Construção Anéis tomaremos como ciclo suporte o ciclo $C_1$ selecionado na fase Construção Ciclos do exemplo anterior.

Com base na lista de demandas associadas ao ciclo $C_1$ selecionado, composto pelos nós $1, 2, 3$ e $4$, montamos a lista de candidatos $E$ como podemos observar a tabela 4.2. Na

Tabela 4.2: Estrutura E

| ind | i | j | k | $\Delta Z$ |
|---|---|---|---|---|
| 1 | -1 | 1 | 1 | ⋮ |
| 2 | -1 | 2 | 1 | ⋮ |
| 3 | -1 | 3 | 1 | ⋮ |
| 4 | -1 | 4 | 1 | ⋮ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 25 | -1 | 1 | 6 | ⋮ |
| 26 | -1 | 2 | 6 | ⋮ |
| 27 | -1 | 3 | 6 | ⋮ |
| 28 | -1 | 4 | 6 | ⋮ |
| 29 | 1 | 2 | 1 | ⋮ |
| 30 | 1 | 3 | 1 | ⋮ |
| 31 | 1 | 4 | 1 | ⋮ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| n | 3 | 4 | 6 | ⋮ |

tabela, os índices $i = -1$ indicam que o elemento correspondente de $E$ contém apenas a variável $e_{ik\hat{l}}$, enquanto os outros elementos contém o par de variáveis $e_{ik\hat{l}}$ e $e_{jk\hat{l}}$. Lembre-se que esta estrutura deve conter o conjunto das variáveis $e_{ik\hat{l}}$ e dos pares de variáveis ($e_{ik\hat{l}}$,$e_{jk\hat{l}}$) para todo $i$, $j \in N^{\hat{l}}$, $i \neq j$, e $k = 1, .., NA^{\hat{l}}$.

No nosso exemplo, possuimos seis opções de anéis ($k = \{1, 2, 3, 4, 5, 6\}$) pois temos dois tipos de ADM's e três demandas, logo $NA^1 = 6$.

Para cada linha da tabela 4.2 resolveremos o problema (4.3) e calculamos, a partir da solução deste problema , $\sum_{i \in \mathcal{N}^i} \sum_{k=1}^{NA^{\hat{l}}} p_k e_{ik\hat{l}} - w \sum_{k}^{NA^{\hat{l}}} \sum_{i=1}^{n} \sum_{j=1}^{n} y_{ijk\hat{l}} d_{ij}$. Note, por exemplo, que se estamos analisando a linha 30 da tabela, onde $i = 1$, $j = 3$, $k = 1$ e $l = 1$, estaremos fixando $e_{111}$ e $e_{311}$ em um e os outros $e_{ik\hat{l}}$ em zero. Com isto poderemos também fixar em zero algumas variáveis do conjunto de variáveis $y_{ijk\hat{l}}$.

Obteremos um conjunto de $\Delta Z's$ correspondentes a lista de candidatos $E$, com isto podemos determinar $h_{min}$ e $h_{max}$ como o melhor e o pior valor da função gulosa. Montaremos a LRC com base em $\{s \in E | \Delta Z(s) \geq h_{min} + \alpha_2 \ (h_{max} - h_{min})\}$.

Supondo que possuimos a seguinte LRC ordenada crescentemente:

$$\text{LRC} = \{E_{29},\ E_{30},\ E_{31}\}$$

Utilizando novamente a *função tendência* para cada $i \in$ LRC. Determinamos a probabilidade associada a cada elemento da LRC como $p\ (29) = 0,33$, $p\ (30) = 0,33$ e $p\ (31) = 0,33$.

Supondo que o número randômico escolhido seja $0,45$, então o elemento escolhido na LRC será 30 como podemos observar na figura 4.9.

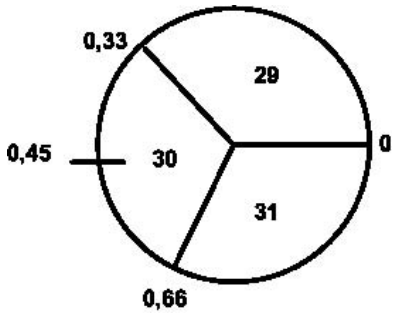


Figura 4.9: Estatistica 2

Com isto concluímos que estaremos construindo o anel $k = 1$, instalando nele o equipamento 1, ou seja, alocando equipamentos do tipo 1, no nó 1 e no nó 3, do ciclo $C_1$, possibilitando assim o atendimento da demanda $d_{13}$ .

Faremos $\hat{X} = \hat{X} \cup \{e_{111}, e_{311}\}$ e analisaremos cada posição da estrutura $E$ para excluir dela todas as possibilidades onde $k = 1$ e possui $i$ e/ou $j$ iguais a 1 ou 3. Em seguida voltaremos a etapa de avaliação de cada posição da estrutura $E$.

Finalizamos o procedimento Construção Anéis quando todas as demandas do ciclo $C_1$ forem atendidas. Cada vez que escolhemos um elemento para ser adicionado à solução $\hat{X}$ verificamos se a inclusão deste elemento fará com que sejam satisfeitas todas as demandas, isto acontece quando a quarta restrição do problema (4.1), que foi relaxada, esta atendida

(o número de variáveis $y_{ijk\hat{l}} = 1$ for igual ao número de demandas a serem atendidas).

Atualiza-se a estrutura *Ciclo* $\rightarrow$ *Demanda*, figura 4.10, excluindo desta todas as demandas que acabamos de satisfazer. No exemplo podemos observar que o ciclo $C_1$ satisfaz três das quatro demandas da rede. Voltaremos então ao procedimento Construção Ciclo para selecionar mais um ciclo suporte para que possamos atender esta última demanda.

| l | Ciclos | Nº de dem |
|---|---|---|
| 1 | 1-2-3-4 | 0 |
| 2 | 2-3-5 | 1 |
| 3 | 1-2-5-3-4 | 1 |

Figura 4.10: Ciclo - Demandas

Supondo que selecionamos o ciclo $C_2$, executamos novamente o procedimento de Construção Anéis para construir um novo anel que atenda a última demanda da rede. Com isto obteremos uma solução viável para o problema (2.1).

## 4.4 Busca Local PAURT

Partindo de cada solução construída na fase de construção do GRASP, emprega-se um procedimento de pesquisa local na tentiva de melhorar a solução.

O procedimento de *Busca Local* que podemos ver na figura 4.12, descreve a fase de Busca Local do GRASP-PAURT.

Para cada tipo de anel ($k$) construído, calculamos sua capacidade livre (Sb), o preço do equipamento ADM (Pr), o número de demandas atendidas (Nd) e o ciclo suporte utilizado ($\hat{l}$) pelo mesmo. Esta estrutura estará ordenada crescentemente pelo número de demandas atendidas e a denominaremos *tab-anel*. Para cada anel definido em *tab-anel* estará associado uma lista de demandas ($L_d$) atendida pelo mesmo. Esta possui nó inicial (de onde partiu o fluxo), o nó final (destino do fluxo), o valor do fluxo e os nós que compõe o ciclo suporte utilizado.

A estrutura *tab-anel* para o exemplo discutido anteriormente está ilustrada na figura 4.11, onde foram construidos os anéis $k = 1$ e $k = 2$ utilizando o ciclo suporte $C_1$, o anel $k = 1$ utilizando o ciclo suporte $C_2$, suas respectivas capacidade livre (sobras), o preço do equipamento utilizado e o número de demandas atendida por cada anel.

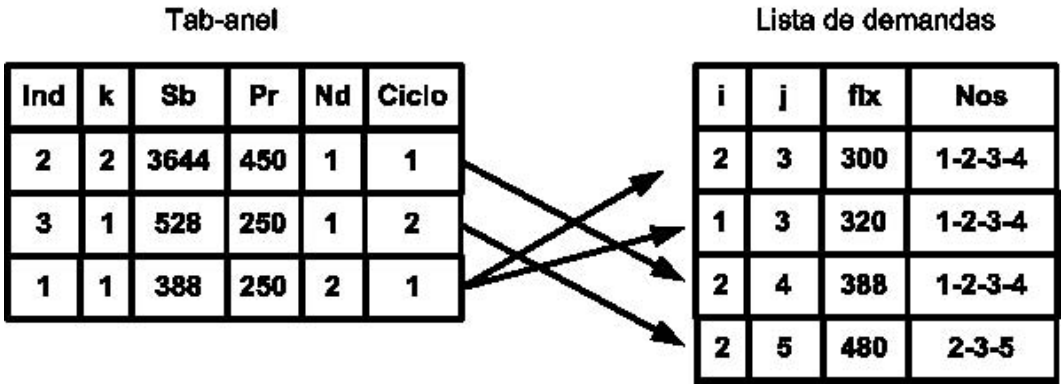


Figura 4.11: Estruturas de Busca Local

Para atender o seu objetivo, o procedimento de busca local proposto se baseia em alocar a demanda de um anel para outro de forma a economizar na instalação de ADMs e eventualmente, eliminar algum anel da rede.

Para isto, procuramos alocar as demandas pertencentes aos anéis que atendem ao menor número de demandas nas sobras de capacidade dos outros anéis da rede, com a intenção de otimizar a utilização da rede de transmissão. É interessante ressaltar que, definido que o procedimento de busca local troca uma demanda de um anel para outro estamos definindo o movimento na vizinhança que caracteriza o procedimento de busca local.

Quando avaliarmos os anéis com menor número de demandas ($Nd_{min}$) estaremos avaliando a estrutura *tab-anel* de cima para baixo, isto é, tentando alocar no anel com o maior número de demandas ($Nd_{max}$, última posição de *tab-anel*), as demanas atendidas no anel com o menor número de demandas avaliado no momento.

Avaliaremos se a demanda pertencente a $L_d[k]$, onde $k$ é a posição com o $Nd_{min}$, poderá ser alocada no anel com $Nd_{max}$, caso contrário avaliaremos a próxima demanda pertencente a $L_d[k]$.

**Procedimento** Busca Local $(\hat{x})$

1 Preencher as estruturas tab-anel e $L_d[k]$;
2 Ordenar tab-anel pelo número de demandas (Nd);
3 $ind \leftarrow$ posição inicial de tab-anel;
4 $fim \leftarrow$ posição final de tab-anel;
5 $indf \leftarrow$ posição final de tab-anel;
6 **Enquanto** $ind < fim$ **faça**
7 **Início**
8 **Enquanto** $indf > ind$ ou $L_d[k] \neq \emptyset$ **faça**
9 **Início**
10 **Para** cada $d_{ij} \in L_d[k]$ **faça**
11 **Início**
12 procedimento Avalia-Troca ($d_{ij}$, $L_d[k_{indf}]$ )
13 **Se** Avalia-Troca = verdadeiro **faça**
14 **Início**
15 Atualizar $k$, Sb, Pr, Nd e $\hat{x}$
16 **Fim**
17 **Fim**
18 $indf \leftarrow indf - 1$
19 **Fim**
20 $ind \leftarrow ind + 1$
21 **Fim**
22 **Retorna** $(\hat{x})$;

**Fim** Busca Local;

Figura 4.12: pseudo código Busca Local

```
Procedimento Avalia-Troca (d_ij, L_d[k_indf], k_ind)
 1  Avalia-Troca = falso
 2  Se i e j pertencem ao ciclo l^ faça
 3  Início
 4     Se d_ij ≤ Sb[k_indf] faça
 5     Início
 6        Se existe equipamento no nó i e no nó j faça
 7          Início
 8           Avalia-Troca = verdadeiro;
 9          Fim
10        Se existe equipamento no nó i ou no nó j faça
11          Início
12            Se Pr[k_indf] < Pr[k_ind]  faça
13            Início
14            Avalia-Troca = verdadeiro;
15            Fim
16          Fim
17        Se não existe equipamento no nó i nem no nó j faça
18          Início
19            Se Pr[k_indf] < Pr[k_ind]  faça
20            Início
21            Avalia-Troca = verdadeiro;
22            Fim
23          Fim
24     Fim
25  Fim
26  Retorna (Avalia-Troca );
Fim Avalia - Troca;
```

Figura 4.13: pseudo código Avalia - Troca

Cada vez que trocamos um elemento (demanda) atualiza-se o anel de onde saiu a demanda e o anel onde a demanda foi alocada, a capacidade livre (Sb), o preço do equipamento (Pr), o número de demandas atendidas (Nd), lista de demandas ($L_d$) atendida pelo mesmo dos respectivos anéis. Consequentemente estaremos atualizando a solução $\hat{x}$.

Finalizamos quando todas as demandas pertencente a $L_d[k]$ tiverem sido trocadas para o anel com $Nd_{max}$. Caso contrário, faremos a mesma tentativa, com as $d_{ij}$ que ainda não foram transferidas, para o penúltimo anel e assim sucessivamente até que $L_d[k]$ esteja vazio ou não exista possibilidade de troca.

Na tentativa de melhorar cada solução construída, avaliaremos todas as possibilidades de retirarmos alguns dos equipamentos alocados (uma de cada vez), isto é, a possibilidade de transferir alguma demanda (ou algumas demandas) para outro anel (anéis) de forma que possamos retirar algum(ns) equipamento sem que deixemos de atender todas as demandas da rede. Com isto teremos a possibilidade de eliminar algum(ns) anel (anéis).

Nesta tentativa analisamos todas as possibilidades de alocar algum(ns) equipamento (um ou dois) e a possibilidade de trocar a demanda sem a necessidade de instalarmos equipamentos, sem que haja aumento no valor da função objetivo do problema (2.1). Estes procedimentos se repetem até que não tenhamos a possibilidade de trocar mais nenhuma demanda.

Deste modo, para um problema de minimização, pesquisamos toda a vizinhança da solução $x_p$, construida na fase de construção, em busca de uma nova configuração $x_{p+1}$, tal que $f(x_{p+1}) < f(x_p)$. Se tal configuração existe torna-se a nova solução e o processo se repete para $x_{p+1}$, caso contrário, $x_p$ é um ótimo local dentro de $N(x_p)$.

Para ilustrar melhor este procedimento tomaremos a solução encontrada anteriormente no procedimento Construção Anéis como sendo a ilustrada na figura 4.14, isto é $e_{211} = 1$, $e_{311} = 1$, que atende a $d_{23}$, $e_{111} = 1$, $e_{311} = 1$ que atende a $d_{13}$, $e_{221} = 1$, $e_{421} = 1$ que atende a $d_{24}$, $e_{212} = 1$ e $e_{512} = 1$ que atende a $d_{25}$.

Na fase de busca local tentaremos encontrar uma solução melhor (custo da rede menor), isto é, tentaremos eliminar algum (ou alguns) equipamento(s) alocado(s) sem que as demandas da rede deixem de ser atendidas e sem que o custo da mesma aumente.

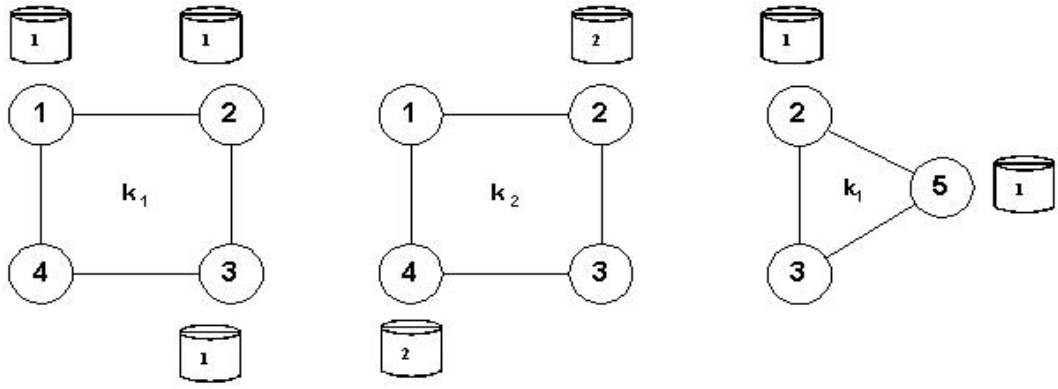


Figura 4.14: Solução

Verificamos que $d_{24}$ pode ser alocado no anel $k = 1$, pois este ainda possui capacidade suficiente para transmitir o fluxo e possui um número maior de demandas que está sendo atendida no mesmo. Além disto o preço do equipamento instalado no anel $k = 1$ é menor do que o alocado em $k = 2$. Com esta troca obtemos uma nova solução $\hat{x}$ e eliminamos o anel $k_2$, como podemos observar na figura 4.16 e na figura 4.15.

Tab-anel

| Ind | k | Sb | Pr | Nd | Ciclo |
|---|---|---|---|---|---|
| 3 | 1 | 528 | 250 | 1 | 2 |
| 1 | 1 | 0 | 250 | 3 | 1 |

Lista de demandas

| i | j | flx | Nos |
|---|---|---|---|
| 2 | 3 | 300 | 1-2-3-4 |
| 1 | 3 | 320 | 1-2-3-4 |
| 2 | 4 | 388 | 1-2-3-4 |
| 2 | 5 | 480 | 2-3-5 |

Figura 4.15: Estruturas de Busca Local

Avalia-se uma nova possibilidade de troca, isto é, estudaremos se $d_{25}$, que pertence ao anel $k_1$ que utilizou o ciclo suporte $C_2$, pode ser transferido para o anel $k_1$ que utilizou o ciclo suporte $C_1$. Porém esta demanda não poderá ser transferida pois o anel $k_1$ que utilizou o ciclo suporte $C_1$ não possui o nó 5.

Observe que quando trocamos a demanda do anel $k = 2$ para o anel $k = 1$, eliminamos

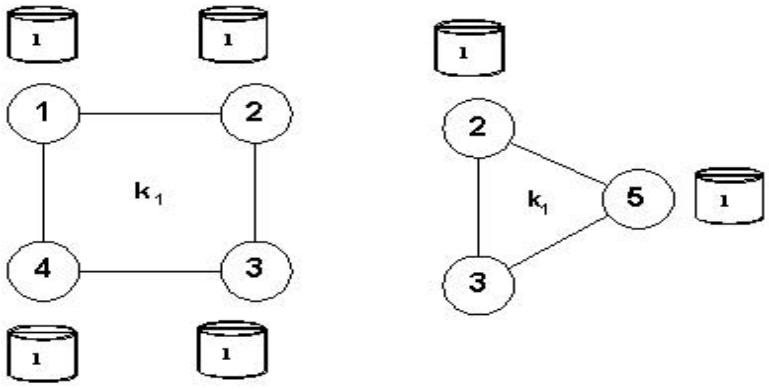


Figura 4.16: Solução Final

os dois equipamentos do anel $k = 2$ e foi necessário alocar apenas mais um equipamento do tipo 1 no anel $k = 1$. Com isto teremos o preço para atendermos a todas as demandas da rede sendo quatro vezes o preço do equipamento alocado no anel $k = 1$, associado ao ciclo $C_1$ mais duas vezes o preço do equipamento alocado no anel $k = 1$, associado ao ciclo $C_2$. Como não possuimos mais nenhuma possibilidade de troca, concluimos que a configuração do ótimo local $\hat{x}$, será a descrita na figura 4.16. Esta será a nossa solução ótima local para este exemplo e com isto concluimos uma iteração do GRASP-PAURT.

# Capítulo 5

# Resultados Computacionais

## 5.1 Gerador de Dados Iniciais

Para executarmos o procedimento GRASP-PAURT precisamos de dados de entrada, tais como o grafo que representa a rede e as demandas que deverão ser alocadas. Para isto, implementamos um gerador de grafos e um gerador de demandas. Estes algoritmos se encontram no apêndice C.

## 5.2 Análise dos Resultados

Podemos observar na tabela 5.2, o número de nós (N), a densidade de arestas no grafo (E), o número de demandas (D), o número de ciclos do grafo analisado (L), o valor ótimo obtido pelo problema 2.1 (quando possível) ($Z^*$), a solução encontrado pelo GRASP-PAURT ($\bar{Z}$), o número de iterações ($Nit$) utilizadas e o tempo total de $Nit$ iterações.

Com o objetivo de comparar os resultados entre a resolução de um método exato e o GRASP-PAURT, consideramos um pequeno problema-teste.

Tomaremos o grafo representado na figura 5.1 que possue 5 nós, com sua matriz de adjacência, a matriz de demanda representadas na figura 5.2, tipos de equipamentos com seus respectivos preços e capacidades representados na tabela 5.1.

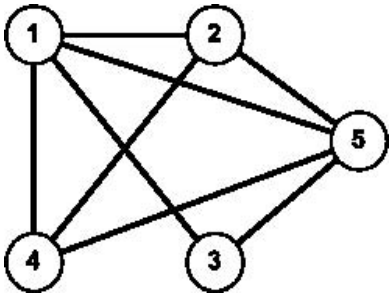


Figura 5.1: Grafo G

| i | j | demanda |
|---|---|---|
| 5 | 2 | 765 |
| 5 | 4 | 828 |
| 2 | 5 | 268 |
| 3 | 5 | 594 |
| 4 | 5 | 634 |

Figura 5.2: Tabela de demandas da rede

Para função tendência randômica, $\alpha_1 = 1$ e $\alpha_2 = 0,5$, a melhor solução encontrada selecionou o ciclo suporte apresentado na figura 5.3, construindo o anel 8, figura 5.4, no qual foram instalados equipamentos do tipo 2.

Utilizando função tendência linear, $\alpha_1 = 1$ e $\alpha_2 = 0,5$, a melhor solução encontrada foi a mesma solução encontrada anteriormente, isto é, foi selecionado o mesmo ciclo suporte e o anel construido (anel 9) utilizou o mesmo tipo de equipamento.

Em uma segunda etapa com função tendência randômica, utilizando $\alpha_1 = 1$ e $\alpha_2 = 0,3$ encontramos a mesma solução apresentada anteriormente, ou seja, foi selecionado o

Tabela 5.1: ADM's

| ADM | Capacidade | Preço ($10^6$ R$) |
|---|---|---|
| ADM - 1 | 1008 Mbps | 250,00 |
| ADM - 2 | 4032 Mbps | 450,00 |

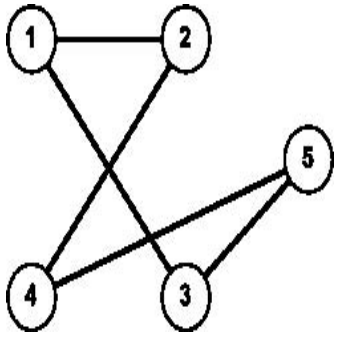


Figura 5.3: Ciclo suporte

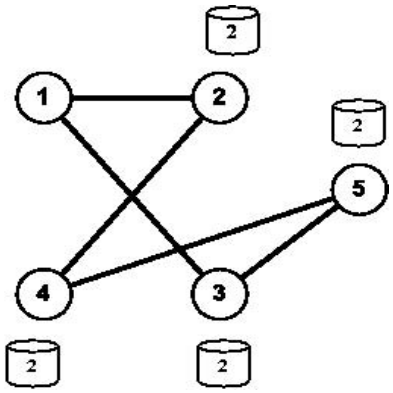


Figura 5.4: Anel

mesmo ciclo suporte e o anel construido (anel 6) utilizou o mesmo tipo de equipamento.

Com a função tendência linear, utilizando $\alpha_1 = 1$ e $\alpha_2 = 0,3$ encontramos novamente a mesma solução apresentada anteriormente,isto é, foi selecionado o mesmo ciclo suporte e o anel construido (anel 9) utilizou o mesmo tipo de equipamento.

Observe que tanto no processo com $\alpha_1 = 1$ e $\alpha_2 = 0,5$ como aquele com $\alpha_1 = 1$ e $\alpha_2 = 0,3$, foi selecionado o ciclo suporte apresentado na figura 5.3, isso se deve ao fato de estarmos utilizando $\alpha_1 = 1$ e a mesma semente nos dois exemplos, o que não acontece na segunda etapa onde os $\alpha_2$ são diferentes.

Além disto os resultado obtidos para a função tendência linear obteve tempo computacional menor, isto se deve ao fato de estarmos beneficiando os elementos que tendem a melhorar a solução, enquanto que com função randômica os elemento tem a mesma

Tabela 5.2: Problema Teste (5 nós)

| $\alpha_1 = 1$ e $\alpha_2 = 0,5$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bias | $N$ | $E$ | $D$ | $L$ | $Z^*$ | $T_1$ | $\bar{Z}$ | $Nit$ |
| Rand. | 5 | 80 % | 5 | 9 | 2000 | 1h 2 min 33 seg | 2000 | 100 |
| Lin. | 5 | 80 % | 5 | 9 | 2000 | 55 min 59 seg | 2000 | 100 |
| $\alpha_1 = 1$ e $\alpha_2 = 0,3$ | | | | | | | | |
| Bias | $N$ | $E$ | $D$ | $L$ | $Z^*$ | $T_2$ | $\bar{Z}$ | $Nit$ |
| Rand. | 5 | 80 % | 5 | 9 | 2000 | 1h 23 min 51 seg | 2000 | 100 |
| Lin. | 5 | 80 % | 5 | 9 | 2000 | 1h 11 min 23 seg | 2000 | 100 |

probabilidade de ser selecionados dificultando assim a obtenção da solução $\hat{X}$.

Note que para $\alpha_2 = 0,3$ o tempo computacional é maior do que quando utilizamos $\alpha_2 = 0,5$. Isso se deve ao fato que para $\alpha_2 = 0,3$ a $LRC$ será composta por elementos com qualidade pior do que a $LRC$ montada quando utilizamos $\alpha_2 = 0,5$.

# Capítulo 6

# Conclusões e Trabalhos Futuros

Problemas de otimização combinatória aparecem frequentemente em vários setores da economia. Como exemplo podemos citar o planejamento de redes de telecomunicação, onde o desenvolvimento de novas técnicas, novos equipamentos e o aumento da busca dos serviços traz a necessidade de reotimizar (reavaliarmos) a rede.

Grande parte dos problemas de otimização combinatória são intratáveis por natureza ou são grandes o suficiente para tornar inviável o uso de algoritmos exatos. Nesses casos métodos heurísticos são usualmente empregados. Esses métodos frequentemente obtêm a solução ótima para um problema, porém não há nenhuma garantia de otimalidade desta solução.

Neste trabalho consideramos o problema planejamento de anéis unidirecionais em rede de telecomunicações (PAURT) que consiste em definir os anéis que comporão a rede de forma a atender a demanda a um mínimo custo, ou seja, consiste em determinar ciclos suporte sobre a rede onde serão construídos anéis, assim como os nós destes ciclos onde serão instalados equipamentos de transmissão.

Para resolver o PAURT, propomos um procedimento GRASP, no qual a fase de construção da solução divide-se em duas sub-fases. Na primera, denominada sub-fase de construção Ciclo, é gerado um ciclo suporte para um dos anéis que comporão a solução. Na segunda, denominada sub-fase de construção Anéis, os equipamentos que comporão este anel, assim como os nós onde serão instalados, são determinados. Finalmente desenvolvemos um procedimento de busca local, no qual um ótimo local para o problema é

procurado na vizinhança de uma solução corrente.

Realizamos testes computacionais com o algoritmo proposto, variando os parâmetros que definem o tamanho da lista restrita de candidatos (LRC), nas duas sub-fases de construção. Resolvemos inicialmente uma instância pequena do problema (5 nós) para comprovar a qualidade dos resultados obtidos. Ao compará-los com a solução ótima desta instância, obtidas através da solução do modelo de programação inteira apresentado para o problema (problema 2.1), observamos que a solução ótima foi obtida em todos as aplicações do GRASP implementadas.

Observamos que quando escolhemos $\alpha$ próximo ao guloso o tempo computacional tende a diminuir. Com $\alpha$ próximo ao guloso a lista restrita de candidatos ($LRC$) conterá elementos com melhor qualidade, isto é, que aceleram a obteção da solução no procedimento de construção, consequentemente diminuindo o tempo computacional.

Planejamos dar continuidade a este trabalho, desenvolvendo e testando novos procedimentos de busca local, funções gulosas alternativas para a fase de construção e finalmente tornando a implementação do procedimento mais robusta de forma a permitir a resolução de problemas de maior porte. Este trabalho já foi de fato inicializado com a adaptação do nosso código à versão 14 do pacote XPRESS.

Trabalhos posteriores nesta linha podem vir a utilizar $\alpha$ aleatório, reativo e funções gulosas diferentes comparando os resultados obtidos, como também utilizar outros métodos como *Path Relinking*, acrescentar novos dados ao problema como roteadores, regeneradores, utilizar um método de busca local a nível de ciclos ou mesmo em dois níveis.

# Referências Bibliográficas


[1] Aiex, Renata Machado; *Uma investigação Experimental da Distribuição de Probabilidade do Tempo de Solução em Heurísticas GRASP e sua Aplicação na Análise de Implementações Paralelas*, Tese de D.Sc., PUC, Rio de Janeiro, RJ, Brasil, 2002.

[2] Brito, José André de M.; *Modelo de Otimização para Dimensionamento de uma Rede de Telecomunicações*, Tese de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1999.

[3] Feo, Thomas; Resende, Maurício; *A probabilistic heuristic for a computationally dificult set covering problem* Operations Research Letters, 8, pp. 67-71, 1989.

[4] Feo, Thomas; Resende, Maurício; *Greedy Randomized Adaptive Search Procedure* Journal of Global Optimization, 6, pp. 109-133, 1995.

[5] Festa, Paola; Resende, Maurício G.C.; *GRASP : An Annotated bibliography*, AT&T Labs Research Technical Report, Jan, 2001.

[6] Glover, Fred; Hanafi, Said; *Tabu Search and Finite Convergence*, Discrete Applied Mathematics, Setember, 1999.

[7] Lin; Kernighan; *Randon multi-start local search*, 1973.

[8] Hard; Shogan; *Semi-greedy heuristics*, 1977.

[9] Maculan, Nelson; Passini, Marcos M.; Brito, José André M.; Lisser, Abdel; *Column Generation Method for Network Design*, Kluwer Academic Publishers, Netherlands, 2000.

[10] Passini, Marcos de Mendonça; *Geração de Colunas em Programação Inteira Aplicada à Síntese de Redes*, Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2001.

[11] Pitsoulis, Leonidas S.; Resende, Maurício G.C.; *Greedy Randomized Adaptive Search Procedure*, AT & T Labs Research Technical Report, Jan, 2001.

[12] Resende, Maurício G.C.; *Greedy Randomized Adaptive Search Procedure (GRASP)*, AT &T Labs Research Technical Report, Dez, 1998.

[13] Resende, Maurício G.C.; *A bibliography of GRASP*, AT &T Labs Research Technical Report, Jan, 2001.

[14] Resende, Maurício G.C.; Ribeiro, Celso C.; *Greedy Randomized Adaptive Search Procedure* , AT&T Labs Research Technical Report, Set, 2001.

[15] Resende, Maurício G.C.; *Combinatorial Optimization in Telecomunications*, AT&T Labs Research Technical Report, Jul, 2001.

[16] Reeves, Colin R.; *Modern Heuristic Techniques for Combinatorial Problems* ; Jonh Wiley & Sons inc, New York, Toronto.

[17] Szwarcfiter, Jayme Luiz; *Grafor e Algoritmos Computacionais*, Editora Campus, pp. 472-484, 1984.

[18] Szwarcfiter, Jayme Luiz; *On optiomal near-optimal algorithms for some computational graph problems* Claremont tower, University of NewCastle upon tyne, 1975.

[19] Szwarcfiter, Jayme Luiz; Lauer, Peter E.; *A Search Strategy for the Elementary Cycles of a Directed Graph*, BIT 16, pp. 192-204, 1975.

[20] Trindade, Marcos B.; Nakamura, Roberto Y.; Sousa Filho, Argemiro O.; *Planejamento de Anéis Síncronos Unidirecionais: Modelo de Otimização e um Estudo de Caso*, CPqD/Telebrás, 2000.

[21] Vieira, Carlos Eduardo Costa; *Metaheurística Aplicada ao Problema de Alocação de Canal em Sistemas de Telecomunicações Móveis*, Tese de M.Sc., Instituto Militar de Engenharia, Rio de Janeiro, RJ, Brasil, 2001.

[22] Viana, Geraldo Valdisio R; *Meta-Heurísticas e Programação paralela em Otimização Combinatória*, EUFC, 1998.

# Apêndice A

# Definições de Grafos

Um **grafo** $G(V, E)$ é um conjunto finito não-vazio $V$ e um conjunto $E$ de pares não-ordenados de elementos distintos de $V$. Os elementos de $V$ são os vértices e os de $E$ são as arestas de $G$ respectivamente. Cada aresta $e \in E$ será denotada pelo par de vértices $e = (v, w)$. Neste caso, os vértices $v$, $w$ são os extremos da aresta $e$, sendo denominados adjacentes. A aresta $e$ é dita incidente a ambos $v$, $w$. Duas arestas que possuem um extremo comum são chamadas de adjacentes. Utilizaremos a notação $n = |V|$ e $m = |E|$ para determinar o tamanho do conjunto $V$ e do conjunto $E$ respectivamente.

Um grafo pode ser visualizado através de uma representação geométrica, na qual seus vértices correspondem a pontos distintos do plano em posições arbitrárias, enquanto que a cada aresta $(v, w)$ é associada uma linha arbitrária unindo os pontos correspondentes a $v$, $w$.

## A.1 Propriedades

### A.1.1 Grafo Simétrico

**Grafo simétrico** é quando podemos afirmar que para todo par de vértice $i$ e $j$, existir os arco $(i, j)$ e $(j, i)$.

### A.1.2 Caminho

Uma sequência de vértices $v_1$, ... , $v_k$ tal que $(v_j, v_{j+1}) \in E$, $1 \leq j < |k-1|$, é denominado **caminho** de $v_1$ a $v_k$.

### A.1.3 Ciclo

Um **ciclo** é um caminho $v_1$, ... , $v_k$, $v_{k+1}$ sendo $v_1 = v_{k+1}$ e $k \geq 3$.

### A.1.4 Conexo

Um grafo $G(V, E)$ é denominado conexo quando existe caminho entre cada par de vértice de $G$. Caso contrário é desconexo.

# Apêndice B

# Gerador de Grafos

O gerador de grafos construirá um grafo com o número de nós e com o grau de densidade escolhido pelo usuário, onde a densidade do grafo determinará o número de arestas que o grafo conterá.

Para o problema PAURT é necessário que a representação da rede seja um grafo conexo e que contenha pelo menos um ciclo. Para isto, este deverá conter um número mínimo de $n$ arestas ($n$ é o número de nós do grafo).

Para tal definimos o número de aresta de um grafo nao direcionado como $\frac{n(n-1)}{2}$ e estabelecemos a densidade da matriz de adjacência como sendo $\frac{n}{\frac{n(n-1)}{2}} = \frac{2}{n-1}$. Como desejamos que um número mínimo de arestas não seja muito baixo multiplicamos o valor encontrado por dois, tendo assim o número mínimo de aresta para os grafos que serão construidos como $\frac{4}{n}$. A matriz de adjacência gerada pelo algoritmo é uma matriz simétrica.

## B.1 Algoritmo Gerador de Grafo

O pseudo-código da figura 2.1 ilustra o algoritmo básico do gerador de grafo que representará a rede, onde *numvert* é o número de vértices e *dens* é a densidade do grafo.

Determinamos para o gerador de grafos uma densidade mínima (4 dividido pelo número de vertices do grafo) que garante que estaremos construindo um grafo conexo, com pelo menos um ciclo.

Seleciona-se um valor aleatóriamente entre $[0, 1]$ para o primeiro vértice $v_i$ e um segundo valor para o segundo vértice e $v_j$ onde $v_i \neq v_j$. Associasse valor um a posição $(v_i, v_j)$ na matriz de adjacência. Este procedimento é executado até que tenhamos preenchido Narestas posições na matriz de adjacência.

## B.2 Gerador de Demandas

O gerador de demandas nos fornecerá as demandas que deverão ser atendidas na rede, levando em conta os ciclos do grafo construido pelo gerador de grafo.

A idéia básica para construção da matriz de demandas, parte de dois nós que contém uma aresta $(v_i, v_j)$ e esta contido em um ciclo. O par $(v_i, v_j)$ será associado a uma demanda, isto é, um valor aleatorio escolhido no intervalo $[Min, Max]$, onde $Max$ e $Min$ correspondem a maior e a menor demandas desejadas. Observe que a matriz de demanda gerada não é uma matriz simétrica.

## B.3 Algoritmo Gerador de Demandas

O pseudo-código da figura 2.2 ilustra o algoritmo básico do gerador de demandas que estará representando o fluxo que estará sendo transmitido do nó $i$ para o nó $j$ na rede, onde $numvert$ é o número de vértices, $Ndens$ é o número de demandas que deverão ser alocadas, $Max$ assume o valor da capacidade do menor equipamento, $MatrizAdj$ é a matriz de adjacência do grafo e Ciclos é a lista de ciclos encontrados no grafo construido pelo gerador de grafos.

Este algoritmo seleciona aleatoriamente um valor no intervalo $[0, 1]$ e calcula com base em $dem = (dem * Max) + Min$ uma demanda no intervalo $[Min, Max]$.

Seleciona-se um valor aleatóriamente entre $[0, 1]$ para o primeiro vértice $v_i$ e um segundo valor para o segundo vértice e $v_j$ onde $v_i \neq v_j$. Estes procedimentos são executados até que tenhamos obtido as $Max$ demandas.

```
Procedimento Gerador de Grafo (numvert, dens)
 1  lim = 4/numvert;
 2   Se (dens ≥ lim) faça
 3     Narestas = (dens/100) * ((numvert * (numvert - 1))/2) ;
 4   Seleciona aleatoriamente v_i ;
 5   numvert = numvert - 1
 6   Enquanto (numvert > 0) faça
 7     Seleciona aleatoriamente v2 (v2 ≠ v_i, C = {v_i});
 8     Se (matriz[v_i][v2] = 0) faça
 9       matriz[v_i][v2] = 1;
10       matriz[v_i][v1] = 1;
11       numvert = numvert - 1;
12       Narestas = Narestas - 1;
13       v_i = v2;
14     fim
15   fim
16   Enquanto (Narestas > 0);
17    Seleciona aleatoriamente v2 (v2 ≠ v_i);
18    Se matriz[v1][v2] = 0 faça
19      matriz[v1][v2] = 1;
20      matriz[v2][v1] = 1;
21      Narestas = Narestas - 1;
22      v1 = v2;
23    fim
24   fim
fim Gerador de Grafo;
```

Figura 2.1: pseudo código Gerador de Grafo

**Procedimento** `Gerador de Demandas` (`numvert`, `Ndem`,`Ciclos`, `Max`)

1 **Enquanto** ($Ndem > 0$) **faça**

2 `Seleciona aleatoriamente` $dem$ ;

3 `dem = (dem * Max) + Min;`

4 `Seleciona aleatoriamente` $v_i$ `e` $v_j$;

5 **Se** ( $((v_i, v_j) \in$ Ciclo) e $(v_i \neq v_j)$ e $(Matrizdem[v_i][v_j] = 0)$) **faça**

6 $Matrizdem[v_i][v_j] = dem$;

7 $Ndem = Ndem - 1$;

8 **fim**

9 **fim**

**fim** `Gerador de Demandas`;

Figura 2.2: pseudo código Gerador de Demandas

# Apêndice C

# Geração de Ciclos

Para gerar os ciclos, implementamos o algoritmo proposto por *Szwarcfiter* [19], o qual parte de um grafo para encontrar todos os ciclos nele contidos. Como o problema de rede está associado a um grafo não direcionado adaptamos este algoritmo de acordo com o procedimento posto em [18].

Para implementação do algoritmo que gera todos os ciclos, precisamos compreender antes as estruturas de dados utilizados para representar o grafo.

## C.1 Estruturas de dados utilizadas na representação de um Grafo

Podemos representar um grafo $G = (V, E)$ na forma matricial ou por listas. Consideremos, $n = |V|$ e $m = |E|$.

Iniciaremos pela matriz de adjacência $A = (a_{ij})$ que é uma matriz $n$ x $n$ tal que:

$$a_{ij} = \begin{cases} 1 & \text{se } (i,j) \in E \\ 0 & \text{caso contrário} \end{cases}$$

Observamos que $A$ é simétrica para um grafo **não direcionado** e o número de $1's$ é igual a $2m$ , pois cada aresta $(i, j)$ dá origem a dois $1's$ em $A$, $a_{ij}$ e $a_{ji}$.

Uma matriz de adjacência nao é única pois podemos permutar as linhas e colunas, o

que não altera o grafo.

Note que existe $n^2$ informações binárias, o que significa um espaço de armazenamento da ordem de $O(n^2)$.

Uma das estruturas de lista usada para configurar adjacência em um grafo $G(V, E)$ é feita da seguinte forma:

$A(v) \setminus v \in V$ é o conjunto de $n$ listas, uma para cada $v \in V$. Cada lista $A(v)$ é denominada lista de adjacência do vértice $v$, e contém os vértices $w$ adjacentes a $v$ em $G$, ou seja, $A(v) = \{w|(v, w) \in E\}$

**Exemplo:**

A figura 3.1 mostra um grafo não direcionado com seis (6) vértices. Cada posição de $A(v)$ representa um vértice e esta possui seis (6) posicões. Observe que a posição 3 da lista não possui nenhuma lista de vértices adjacêntes e que aponta para o fim da lista, isto ocorre pois o vértice 3 é um vértice isolado, isto é que não possui conexão com os outros vértices do grafo.

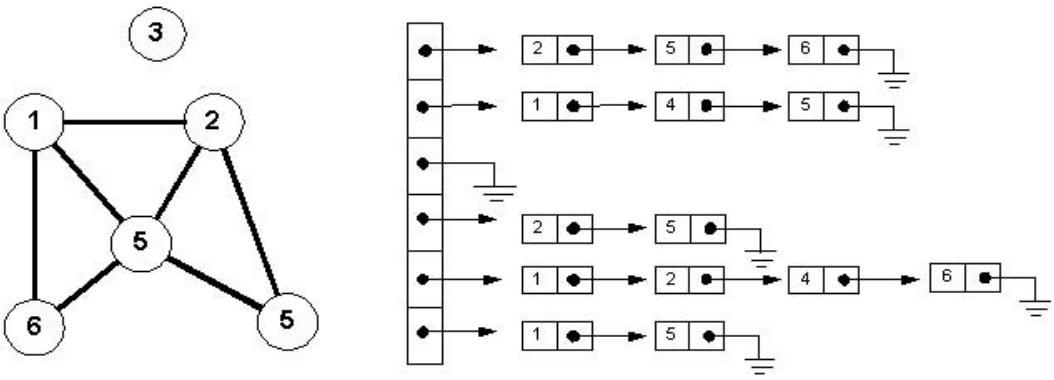


Figura 3.1: Estrutura de adjacência

Observe que cada aresta $(v, w)$ dá origem a duas entradas na estrutura de adjacências, correspondentes a $v \in A(w)$ e $w \in A(v)$. Logo a estrutura $A$ consiste de $n$ listas com um total de $2m$ elementos.

Generalizando teremos que a cada elemento $w$ de uma lista de adjacência $A(v)$ associa-se um ponteiro, o qual informa o próximo elemento após $w$ em $A(v)$, se houver. Além disso, é necessário também a utilização de um vetor $p$, de tamanho $n$, tal que $p(v)$ indique

o ponteiro inicial da lista $A(v)$. Assim sendo, se $A(v)$ for vazia, $p(v)$ é vazio. Pode-se concluir então que o espaço utilizado por uma estrutura da adjacência é $O(n + m)$, ou seja, linear com o tamanho de $G$.

A estrutura de adjacência pode ser interpretada como uma matriz de adjacência, representada sob a forma de matriz esparsa, isto é, na qual os zeros não estão presentes, neste caso $j \in A(i)$ corresponderia a $a_{ij} = 1$ na matriz de adjacência. Maiores detalhes em [17].

## C.2 Algoritmo de geração de ciclos

Para que tenhamos um algoritmo que resolva o problema de geração de ciclos em um grafo **não direcionado** precisamos de alguns dados iniciais e que estejam representados de forma adequada.

Dados iniciais necessários:

- Matriz de Adjacência do grafo não direcionado: $Matriz_{Adj}[x][y]$;
- Número de nós da rede;

*Szwarcfiter* [19] propoe um algoritmo com um procedimento eficiente para detectar ciclos em um grafo qualquer.

Onde a idéia principal é partindo de um nó inicial $v$ do grafo, construir um caminho utilizando informações da Matriz de Adjacência, isto é, buscamos um nó $w$ adjacente a $v$ que será inserido no caminho. Este processo se repete sucessivamente até que seja formado um ciclo, ou seja, paramos quando inserimos um nó que já pertence ao caminho. Observe que o ciclo formado não conterá necessariamente todos os nós do caminho.

Como exemplo consideraremos o grafo representado na figura 3.2. Vamos supor que partimos do nó 4, construimos o caminho, $4 - 1 - 2 - 6 - 3 - 2$.

A medida que construimos o caminho colocamos o vértice em uma lista (pilha) e o marcamos em uma lista de vértices. Quando verificamos que o vértice que vai ser inserido na pilha está marcado na lista de vértices, podemos afirmar que encontramos um ciclo.

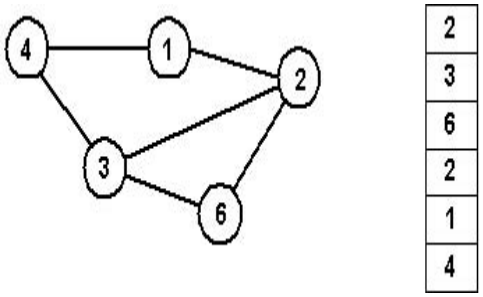


Figura 3.2: Exemplo 3

Ao detectarmos a formação de um ciclo, inserimos este em uma lista de ciclos e para isto guardamos o último elemento (elemento do topo) em uma variável temporária (*temp*) e retiramos cada elemento da pilha comparando com o valor de *temp*. Paramos quando o valor de *temp* é igual ao do elemento retirado da pilha, garantimos assim que retiramos todo o ciclo da pilha.

Neste exemplo podemos notar que apesar do caminho possuir os vértices $4-1-2-6-3-2$, o ciclo encontrado é formado pelos vértices $2-6-3-2$, isto é, não contém todos os vértices do caminho.

### C.2.1 Algoritmo Ciclos

Para implementação deste algoritmo utiliza-se uma pilha a qual chamaremos de *pilhano* (*stack*) e esta terá como finalidade guardar os nós que poderão vir a formar um ciclo, isto é, a pilha guarda na posição $j$, o número do $j$-ésimo nó que entrou no caminho. Chamamos atenção para o fato que o ciclo formado, não necessariamente será constituído com o primeiro nó (nó inicial) a ser inserido na pilha até o nó que se encontra no topo da pilha.

Utilizamos um vetor que chamaremos de *marca* (*mark*). Cada posição de marca estará associada a um vértice do grafo e será marcada cada vez que o nó $j$ entrar na *pilhano*.

$$marca = \begin{cases} 1\ (\textit{verdadeiro}), & \text{se o nó já está na pilha} \\ 0\ (\textit{falso}), & \text{caso contrário} \end{cases}$$

Fazemos uso de um o vetor *posicao* (*position*) e de um vetor *alcance* (*reach*). O vetor *posicao* é usado para guardar a posição do nó $v$ na pilha *pilhano* e tem tamanho determinado pelo número de vértices do grafo. O vetor *alcance*, é um vetor lógico, que estará marcado caso o vértice $v$ já tiver sido explorado.

$$alcance\ [v] = \begin{cases} 1\ (verdadeiro), & \text{se alcançamos o vértice} \\ 0\ (falso), & \text{caso contrário} \end{cases}$$

As variáveis $t$, $f$, $q$, $g$ são variáveis auxiliares utilizadas no processo de detectar ciclos. Onde $t$ determina onde é o topo da lista, $q$ guarda a posição do último elemento inserido, $f$ e $g$ são usados para marcar se formamos um ciclo ou não, porém $g$ é usado dentro do processo de recursão.

O procedimento pode ser resumido em duas possibilidades quando analisamos se o vértice $w$ poderá entrar na *pilhano*, isto é, $w$ está marcado no vetor *marca* ou $w$ não está marcado no vetor *marca*.

Se $w$ estiver marcado no vetor *marca* e se existir três ou mais vértices acima de $w$ então achamos um ciclo, caso contrário nenhum ciclo foi formado.

Se $w$ ainda não estiver marcado no vetor *marca* e se existir três ou mais vértices acima de $w$ marcamos $w$ no vetor *marca*, associamos a variável f valor falso (nao formamos ciclo), inserimos $w$ na pilha, associamos à variável t o topo e $posicao[w]$ recebe a posição de $w$ na pilha, caso contrário nenhum ciclo foi formado.

Por exemplo, supondo o início do processo. Inicializamos $alcance[v] = 0$, $marca[v] = 0$, $\forall v$. Se tomarmos como vértice inicial o nó $v$ e $w_1$ o vértice adjacênte a $v$, $w_2$ o vértice adjacênte a $w_1$, ... , $w_i$ adjacênte a $w_k$.

Ao selecionarmos um vértice inicial $v$ e este não estiver marcado no vetor *marca*, marcamos $v$ no vetor *marca*, inserimos $v$ na *pilhano* e atualizamos a variável $t$ com a posição do topo. Como $alcance[v] = 0$, associamos à variável $q$ a posição de $v$ na *pilhano*. Repetiremos este procedimento para o vértice $w_1$ adjacênte a $v$.

Por estarmos trabalhando com um grafo não direcionado, ao analisarmos os vértices adjacêntes a $w_1$, iremos analisar novamente $v$, porém este já estará marcado no vetor *marca*, neste caso verificamos se a posição de $v$ é menor que q. Como só possuímos

dois vértices na *pilhano* a diferenca da posição destes dois elementos será igual a um, indicando que não formamos um ciclo. Chamaremos então o procedimento *nenhumciclo* que deletará essa aresta da matriz de adjacência ($MatrizAdj[w_1, v]$) e marcaremos no vetor $B$ ($B[w_1, v]$). Repetiremos o mesmo processo, analisando outros vértices adjacêntes a $w_1$.

Ao testamos se um vértirce $w_i$ pode ser inserido na pilha ($w_i$ adjacênte a um vértice $w_k$ que já se encontra na *pilhano*) e este já estiver marcado em *marca*, se verificarmos que a posição de $w_i$ é menor que $q$ e a diferença entre $w_k$ e $w_i$ é diferente de um, teremos formado um ciclo com os vértices entre $w_k$ e $w_i$. Utilizando o procedimento *saidaciclo* guardamos este ciclo, associamos à variável $f$ o valor um e buscamos um novo vértice para ser inserido na *pilhano*.

Quando chegamos ao vértice máximo ($v_{max}$), deletamos o último vértice que entrou na pilha, utilizando o procedimento *desmarca* desmarcamos este do vetor *marca* e reconstituímos a matriz de adjacência utilizando o auxílio da matriz $B$, isto é, se o vértice $w_i$ é o vértice que se encontra no topo da *nopilha*, para todo $0 \leq x \leq v_{max}$, se $B[w_i][x] = 1$ então $MatrizAdj[w_i][x] = 1$. Em seguida $B[w_i][x]$ receberá valor zero, para todo $0 \leq x \leq v_{max}$.

Por último o vetor *alcance* $[w_i]$ receberá valor 1, indicando que já exploramos todos os caminhos possíveis partindo do vértice $w_i$ e analisaremos o vértice que agora se encontra no topo, testando todas as possibilidades de inserirmos outros vértices. Quando não existir mais possibilidades, repetiremos o processo que deleta o vértice que se encontrar no topo. Observe que testaremos novamente se o vértice $w_i$ pode entrar no caminho, porém este não entrará pois já está marcado no vetor *alcance*. Estes procedimentos serão repetidos até que deletemos o primeiro vértice a entrar na pilha ($v$).

Os procedimentos *inserenapilha*, *deletadapilha*, *ProcuraListaciclos* e *InsereListadeciclos* são procedimentos comuns envolvendo pilha e lista ordenada onde $v$ é o vértice do topo da pilha.

Os procedimentos *Nenhumciclo*, *desmarca*, *saidaciclo* são chamados dentro do procedimento *ciclo*, apresentado na figura 3.3, e serão descritos abaixo.

**Nenhumciclo**

O procedimento *nenhumciclo* é chamado quando um vértice $w$ adjacente a $v$ não nos

```
Procedimento Ciclo (v, q, f )
 1  marca[v] = 1 ;
 2  f = 0 ;
 3  Inserepilha(v) ;
 4  t = topo ;
 5  posição[v] = t − 1 ;
 6  Se alcance[v] = 0 faça
 7    q = t − 1 ;
 8  Para w = 0, ..., numvertices faça
 9   Se MatrizAdj[v][w] = 1 faça
10    Se marca[w] = 0  faça
11      Ciclo(w, q, g) ;
12      Se g = 1
13        f = 1 ;
14      Senão Nenhumciclo(v, w) ;
15    Senão faça
16      Se posição[w] ≤ q faça
17        dif = posição[v]−posição[w] ;
18       Se dif ≠ 1
19         Saidaciclo(w, v) ;
20       f = 1 ;
21      Senão Nenhumciclo(v, w) ;
22    fim
23   fim
24  fim
25   Deletapilha() ;
26   Se f = 1 faça
27     Desmarca(v) ;
28   alcance[v] = 1 ;
29   posição[v] = numvertices ;
fim Ciclo;
```

Figura 3.3: pseudo codigo Ciclo

```
procedimento Nenhumciclo (x, y)
   1    B[y][x] = 1 ;
   2    MatrizAdj[x][y] = 0 ;
fim Nenhumciclo;
```

Figura 3.4: pseudo codigo nenhumciclo

```
procedimento Desmarca (x)
   1   marca[x] = 0;
   2    Para y = 0, ..., numvertices faça
   3      Se B[x][y] ≠ 0 faça
   4        MatrizAdj[y] [x] = 1 ;
   5        Se marca[y]   faça
   6          desmarca[y] ;
   7      fim
   8    fim
   9    Para y = 0, ..., numvertices faça
  10      B[x][y] = 0 ;
fim Desmarca;
```

Figura 3.5: pseudo codigo Desmarca

leva a um novo ciclo. Deleta-se o vértice $w$ da matriz de adjacência, o qual é passado pelo procedimento *ciclo*, e o marcamos na matriz $B$, para podermos restaurar a matriz de adjacência em uma nova iteração, como podemos ver no pseudo código apresentado na figura 3.4).

**Desmaca**

O procedimento *desmarca* é chamado quando ao inserirmos o vértice $w_i$ e detectamos a formação de um ciclo. Este desmarca o vértice $w_i$ e restaura a matriz de adjacência com base na matriz $B$, como podemos ver no pseudo código apresentado na figura (3.5).

**Saidaciclo**

Este procedimento é chamado quando detectamos a formação de um ciclo.

```
procedimento Saidaciclo (w, v)
  1  Para k = 0, ..., numvertices faça
  2    candidato[k] = 0;
  3  Para k = position[w], ..., position[v] faça
  4    candidato[pilhano[k]] = 1;
  5  Se ProcuraListaciclos (candidato)
  6    retorna;
  7  InsereListadeciclos(candidato,numv);
end Saidaciclo;
```

Figura 3.6: pseudo codigo Saidaciclo

Toma-se o vértice $v$, onde $v$ é o vértice que se encontra no topo da pilha, em seguida chama-se o procedimento *ProcuraListaCiclos* para verificar se o ciclo formado já existe na lista de ciclos, caso contrário chama-se o procedimento *InsereListadeCiclos* que o guarda, como podemos ver no pseudo código apresentado na figura (3.6).

Maiores detalhes podem ser encontrados em [19].