



**UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**



THIAGO HENRIQUE FREIRE DE OLIVEIRA

**ALGORITMOS HÍBRIDOS PARA O PROBLEMA DE
ATRIBUIÇÃO DE LOCALIDADES A ANÉIS EM REDES
SONET/SDH**

**MOSSORÓ - RN
2015**

THIAGO HENRIQUE FREIRE DE OLIVEIRA

**ALGORITMOS HÍBRIDOS PARA O PROBLEMA DE
ATRIBUIÇÃO DE LOCALIDADES A ANÉIS EM REDES
SONET/SDH**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação - associação ampla entre a Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semi-Árido, para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof^o Dr. Dario José Aloise

Coorientador: Prof^o Dr. Christophe Duhamel

**MOSSORÓ - RN
2015**

**Catálogo da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.**

Oliveira, Thiago Henrique Freire de
Algoritmos Híbridos Para O Problema De atribuição De Localidades A Anéis
Em Redes Sonet/SDH. / Thiago Henrique Freire de Oliveira . – Mossoró, RN,
2015.

62 f.

Orientador: Prof. Dr. Dario José Aloise

Dissertação (Mestrado em Ciência da Computação) Universidade do
Estado do Rio Grande do Norte. Universidade Federal Rural do Semi-
Árido. Programa de Pós-Graduação em Ciência da Computação.

1. Ciencia da Computação - Dissertação. 2. Algoritmos Híbridos. 3.
Redes Sonet/SDH. 4. Projeto de redes. I. Aloise, Dario José .
II. Universidade do Estado do Rio Grande do Norte. III. Universidade
Federal Rural do Semi-Árido. IV. Título.

UERN/BC

CDD 004

Bibliotecário: Sebastião Lopes Galvão Neto – CRB 15-486

THIAGO HENRIQUE FREIRE DE OLIVEIRA

ALGORITMOS HÍBRIDOS PARA O PROBLEMA DE ATRIBUIÇÃO DE LOCALIDADES

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação – associação ampla entre a Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semi-Árido, para a obtenção do título de Mestre em Ciência da Computação.

APROVADA EM: ____/____/____

BANCA EXAMNINADORA



Dr. Dario José Aloise, UERN
Orientador



Dr. Christophe Duhamel, UBP
Coorientador



Dra. Andréa Cynthia Santos, UTT
Membro Externo



Dr. Carlos Heitor Pereira Liberalino,
UERN Membro Interno



Dr. Luiz Amorim Carlos, UFRN
Membro Externo

Dedico este trabalho as minhas duas mães, Maria
Aldênia Barbosa Freire e Ione Maria Alcolea!

AGRADECIMENTOS

Os agradecimentos prestados neste trabalho, são dedicadas a pessoas que contribuíram de alguma forma para a construção do mesmo.

Agradeço principalmente a Deus por tudo que ele tem feito em minha vida e as pessoas que ele tem colocado em meu destino.

As minhas duas mães, Maria Aldênia Barbosa Freire e Ione Maria Alcolea pois são elas que me dão força todos os dias, nunca vi duas pessoas tão batalhadoras em toda a minha vida.

Ao meu orientador, conselheiro e amigo Professor Dario José Aloise por tudo que ele tem feito, seja ensinando, reclamando, aconselhando, cobrando ou qualquer outra coisa. Continue assim professor, o sr. é o meu exemplo, não vou nem quantificar o que aprendi com o sr., pois não teria como.

Ao Professor Christophe por toda sua sabedoria, vontade de ensinar e paciência comigo, é claro. O sr. pode não saber, mas voltei da França como fã do sr., uma das pessoas mais inteligentes que já vi, obrigado por tudo.

A Professora Andréa Cynthia por tudo o que me ensinou também, pela paciência e pela presteza quando cheguei a França e PRINCIPALMENTE pelas reclamações, pois eu sei que precisava, na verdade sei que precisaria ter levado mais, poucas pessoas vi ter tão pulso firme quanto a sra..

A todos os professores da UERN e UFRS que compartilharam conhecimento com vontade e respeito.

A sra. Rita por me auxiliar e ajudar quando precisei ir, uma pessoa muito divertida e alegre.

A todos os meus colegas do NEaD, que em momentos difíceis me ajudaram com apoio e compreensão, principalmente nos atrasos.

Por último e não menos importante, a todos os meus amigos, sejam os de antes, durante ou após o mestrado, obrigado pelas conversas, distrações e alegrias e agradeço também é claro a minha namorada por todo o apoio nas horas mais difíceis a qual passei.

"Impor a disciplina a si mesmo é uma das grandes vitórias do homem, contra si mesmo."

Textos judaicos.

RESUMO

O problema de partição em um grafo $G = (V, E)$ consiste em obter V , o conjunto de vértices, como a união de subconjuntos ou partições disjuntas. Neste estudo, nós propomos um procedimento *Split* para minimizar o número de partições para uma determinada sequência de nós, considerando as restrições de capacidade locais e globais das partições. O procedimento proposto utiliza programação dinâmica e é usualmente aplicado para problemas de roteamento de veículos. Neste trabalho, o procedimento é aplicado ao Problema de Atribuição de Localidades a Anéis em redes SONET. O PALAS é NP-difícil e tem aplicações em projeto de redes de fibra ótica. Duas estratégias foram aplicadas ao PALAS usando programação dinâmica: na primeira estratégia, um número de sequências aleatórias é gerado e particionado; no segundo, o procedimento tem sido usado como um decoder para a heurística BRKGA (Biased Random Key Genetic Algorithm). Estas estratégias apresentaram resultados computacionais superiores aos encontrados até o momento na literatura.

Palavras-chave: PALAS, Programação dinâmica, Heurísticas, BRKGA, projeto de redes.

ABSTRACT

The graph partition problem in a graph $G = (V, E)$ is to obtain V , the set of vertices, as the union of disjoint subsets or partitions. In this study, we propose a Split procedure to minimize the number of partitions for a given sequence of nodes, considering local and global capacity constraints on the partitions. The proposed procedure uses dynamic programming and this procedure is usually applied to vehicle routing problems. The procedure relies on dynamic programming and it is applied here to the SONET Ring Assignment Problem. SRAP is a NP-hard problem and finds applications for designing optical fiber networks. Two strategies have been applied to SRAP using the dynamic programming: in the first strategy, a number of random sequences are generated and partitioned; in the second one, the procedure has been used as a decoder in a BRKGA (*Biased Random Key Genetic Algorithm*) heuristic. These strategies showed computational results superior those found in the literature to date.

Key-words: PALAS, Dynamic programming, Heuristics, BRKGA, Network design.

LISTA DE FIGURAS

Figura 1 – Rede de telecomunicações em grafos	16
Figura 2 – Três níveis da rede	17
Figura 3 – Capacidade de sobrevivência com uma topologia em anel	22
Figura 4 – Rede SONET/SDH com topologia em anel	22
Figura 5 – Sobreposição de subproblemas, através do exemplo do problema da multiplicação de cadeia de matrizes. Cada nó representa parâmetros i e j	32
Figura 6 – Etapas dos algoritmo genético.	35
Figura 7 – Exemplo de um crossover.	37
Figura 8 – Criação de uma nova população.	39
Figura 9 – Criação de uma nova população.	40
Figura 10 – Movimento realizado por Vizinhaça N_a	44
Figura 11 – Movimento realizado por Vizinhaça N_b	44
Figura 12 – Esquema dos métodos de resolução	46

LISTA DE TABELAS

Tabela 1 – SONET/SDH Designações e Larguras de Banda	19
Tabela 2 – Resultados para as instâncias C1	50
Tabela 3 – Resultados para as instâncias C2	53
Tabela 4 – Resultados para as instâncias C3	58
Tabela 5 – Resultados do estado da arte para os conjuntos de instâncias C1, C2 e C3.	58

LISTA DE ALGORITMOS

1	Pseudo-código para determinar anéis para o SRAP.	42
2	Pseudocódigo do Multi-Start para o PALAS.	45

LISTA DE ABREVIATURAS E SIGLAS

ADM Multiplexador Add/Drop, página 20

AF Anel Federal, página 21

ANSI American National Standards Institute, página 18

BRKGA Biased Random-Keys Genetic Algorithm, página 38

ITU International Telecommunication Union, página 18

OC Optical Carrier, página 18

PALAS Problema de Atribuição de Localidades a Anéis em redes SONET/SDH, página 16

PDH Plesiochronous Digital Hierarchy, página 18

RKGA Random-Keys Genetic Algorithm, página 38

SDH Synchronous Digital Hierarchy, página 18

SONET Synchronous Optical NETwork, página 18

STM Synchronous Transport Module, página 18

STS Synchronous Transport Signal, página 18

TSP Travelling Salesman Problem, página 41

VRP Vehicle Routing Problem, página 41

SUMÁRIO

1	INTRODUÇÃO	14
2	REDES DE TELECOMUNICAÇÕES E O PADRÃO SONET/SDH .	16
2.1	Visão básica da estrutura de uma rede de telecomunicações	16
2.2	O Padrão SONET/SDH	18
2.2.1	O Sinal SONET/SDH	19
2.2.2	Elementos da Rede SONET	19
2.2.3	Topologia do Padrão SONET/SDH	21
3	DESCRIÇÃO DO PROBLEMA E MODELAGEM MATEMÁTICA .	23
3.1	Descrição do Problema	23
3.2	Modelo matemático	24
4	TRABALHOS RELACIONADOS	26
5	ABORDAGENS EXATAS E HEURÍSTICAS	30
5.1	Programação Dinâmica	30
5.1.1	Elementos da programação dinâmica	31
5.1.2	Estratégias	32
5.2	Algoritmos Evolutivos	33
5.2.1	Algoritmos Genéticos	34
5.2.1.1	Indivíduos	34
5.2.1.2	População	35
5.2.1.3	Inicialização	36
5.2.1.4	Avaliação	36
5.2.1.5	Seleção	36
5.2.1.6	Reprodução	37
5.2.1.7	Mutação	37
5.2.1.8	Atualização	38
5.2.1.9	Finalização	38
5.2.2	Algoritmos Genéticos com Chaves Aleatórias	38
5.2.3	Algoritmos Genéticos com Chaves Aleatórias Viciadas	39
6	BRKGA E MULTI-START COM BUSCA LOCAL	41
6.1	Procedimento Split	41
6.2	Busca Local	43
6.2.1	Vizinhança N_a	43

6.2.2	Vizinhança N_b	44
6.3	Multi-Start e BRKGA	45
7	RESULTADOS	47
7.1	Resultados das metaheurísticas	47
8	CONSIDERAÇÕES FINAIS	59
	REFERÊNCIAS	60

1 INTRODUÇÃO

A necessidade de interação entre as pessoas e a facilidade de acesso à informação é algo que não tem limites na sociedade atual e tudo isto é possibilitado pelas telecomunicações. Dentre os principais serviços promovidos pelas redes de telecomunicações estão: transferência de dados, de voz, de imagens, transações bancárias, compras, presença virtual em outros lugares, aquisição passagens aéreas e vários outros.

Os serviços que surgem e que são agregados, despertam o interesse da população e atraem novos usuários. Com aumento na demanda de usuários, a infraestrutura de atendimento tem de melhorar para suportar a nova demanda e com isso surgem os problemas de planejamento de novas redes de telecomunicações que têm se tornado cada vez mais complexos, mesmo com a alta qualidade dos equipamentos que são lançados.

Com a notoriedade dos problemas de rede de telecomunicações e a vontade das empresas em conquistarem cada vez mais, muito se tem investido em recursos tecnológicos e pesquisas, atraindo assim muitos pesquisadores, principalmente do campo da Pesquisa Operacional. O problema de planejamento de uma rede de telecomunicações pode ser formulado como um problema de otimização combinatória e representado através de um grafo.

Este trabalho tem como problema a ser resolvido, um problema de planejamento de redes de telecomunicações. O planejamento de uma rede de telecomunicações consiste em determinar adequadamente todas os seus pontos de conexões entre conjuntos de localidades, de forma a atender a todas as suas restrições no menor custo possível. Das restrições, a principal é a questão de sobrevivência da rede em caso de alguma falha. A capacidade de sobrevivência de uma rede é a capacidade da mesma permanecer ativa em caso de alguma falha, seja ela qual for. Para atender esse requisito, faz-se necessário o uso da topologia em anel e da tecnologia SONET (*Synchronous Optical NETWORK*).

O problema citado acima, é conhecido na literatura como o Problema de Atribuição de Localidades a Anéis em Redes SONET/SDH (PALAS). Trata-se de um problema de otimização combinatória da classe NP-difícil (Goldschmidt, O.; Laugier, A.; Olinick, E. V., 2003).

O PALAS é um problema que tem origem na fase de planejamento da rede de telecomunicações, podendo ser modelado por um grafo não-orientado $G = (V, E)$, onde V é o conjunto de n clientes e E representa o conjunto de demandas entre os clientes. O problema de de partições em grafos consiste em dividir V em conjuntos disjuntos $V = V_1 \cup V_2 \cup \dots \cup V_l$, os quais representam anéis. Um anel é formado por um conjunto de nós $i \in V$, cuja capacidade total $Q \in \mathbb{N}^*$ do anel limitada. O PALAS, portanto, consiste em definir partições de V (cada partição corresponde a um anel, chamado de anel local),

respeitando a capacidade Q , dos anéis locais, com o objetivo de minimizar a quantidade de anéis. Após a construção das partições, um anel central (anel federal), conecta todos os anéis locais, a definirem a topologia da rede, desde que respeitada a capacidade Q .

Os componentes que integram a rede são substancialmente caros, especialmente um, o DXC (*Digital Cross Connect*), aparelho responsável por interligar todos os anéis locais. O custo desse equipamento pode se tornar tão alto, frente as demais, que pode levar o problema apenas ao de minimizar a utilização dos mesmos. Existem outras abordagens para o PALAS onde não se usa o DXC, quando são substituídos por mais ADMs (*Add-Drop Multiplexer*). Os ADMs são equipamentos responsáveis por interligar os níveis mais inferiores de uma rede de telecomunicações, formando centros locais de rede, sendo assim anéis locais.

Sendo um problema NP-difícil, algoritmos exatos não possuem tempo computacional viável, por isso problemas dessa classe são resolvidos por métodos heurísticos, que embora não garantam solução ótima, podem oferecer soluções tão próximas do ótima ou até ótima, alcançada em tempo computacional viável. Muitos métodos já foram utilizados neste problema e cada vez mais continuam a aparecer novas formas de resolução pela integração de métodos heurísticos, conhecidos também como metas-heurísticas híbridas ou simples.

Devido a complexidade do PALAS, este trabalho tem como objetivo propor o uso de dois algoritmos para a resolução do do mesmo. O primeiro algoritmo usado é o BRKGA (*Biased Random-Key Genetic Algorithm*) enquanto que o segundo envolve Programação Dinâmica. O BRKGA é um algoritmo genético modificado, cuja modificação caracteriza-se pelo uso do elitismo na seleção de indivíduos para o cruzamento, onde sempre um dos indivíduos será levado para a população elite. Já a Programação Dinâmica contribuirá para validar uma solução de forma rápida e eficaz por meio de uma técnica conhecida como *Split*, proposta por (Beasley, J. E., 1983), comumente aplicado a problemas de roteamento de veículos, (Afsar, H. M.; Prins, C.; Santos, A. C., 2014; Prins, C.; Lacomme, P.; Prodhon, C., 2014).

O trabalho possui a seguinte estrutura: no Capítulo 2, será apresentado e descrito o Problema de Atribuição de Localidades a Anéis em Redes SONET/SDH, assim como a formulação matemática utilizada na resolução do mesmo. O capítulo 3 apresenta os trabalhos que serviram de base para esta dissertação, onde os trabalhos selecionados tiveram significativa importância na literatura do problema. No Capítulo 4, os algoritmos *Biased Random-Key Genetic Algorithm* e a abordagem por meio de Programação Dinâmica, o *Split*, serão descritos. No Capítulo 5 serão abordadas as estratégias utilizadas para a resolução do problema, desde as metaheurísticas utilizadas até a o processo de busca local. O Capítulo 6 apresenta os testes e os resultados em comparação com os trabalhos da literatura. No Capítulo 7 apresenta-se considerações finais e as sugestões para trabalhos futuros.

2 REDES DE TELECOMUNICAÇÕES E O PADRÃO SONET/SDH

O problema do SONET/SDH consiste em projetar uma rede de telecomunicações, para isso o projeto é subdividido em duas fases, a saber: o projeto físico, que pode fazer uso da topologia em anel (será mostrada mais adiante) para a formação de subconjuntos de clientes - os anéis da rede; o projeto lógico que será responsável por estabelecer conexões entre os clientes.

A seguir, na seção 2.1 é mostrada a visão básica da estrutura de uma rede de telecomunicações. Na seção 2.2 é descrito o Padrão SONET/SDH. Na seção 2.3 é mostrado o sinal do padrão SONET/SDH. Na seção 2.4 são descritos todos os elementos presentes no PALAS. Na seção 2.5, apresenta-se a contendo uma descrição e motivação da topologia usada.

2.1 VISÃO BÁSICA DA ESTRUTURA DE UMA REDE DE TELECOMUNICAÇÕES

Basicamente uma rede de telecomunicações pode ser representada através de grafos. Através desta representação, os vértices passam a ser os clientes e as arestas são os cabos que realiza a ligação entre elas, onde cada aresta possui um valor associado para representar a demandas de tráfego simétrico como representado na Figura 1. Construir uma rede totalmente conectada é uma solução não real, devido a enorme quantidade de cabos necessários para transmitir todas as demandas entre os clientes (Soriano, P. et al., 1998).

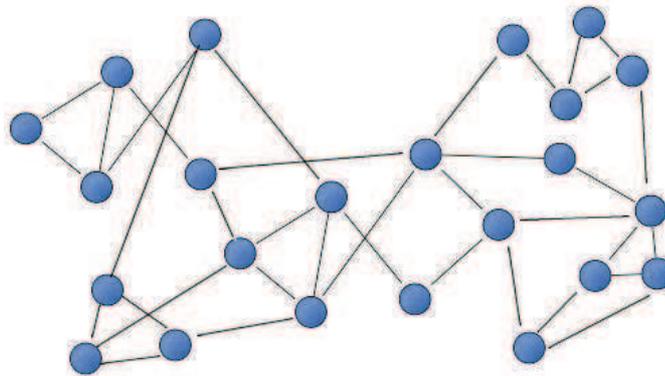


Figura 1 – Rede de telecomunicações em grafos

Uma rede de telecomunicações não é apenas uma rede, é uma hierarquia de redes em três níveis, como mostrado na Figura 2:

1. No primeiro nível, os usuários ou distribuidores de um serviço a um determinado local, são aglomerados e interligados por meio de um *hub*, formando assim um conjunto, chamado de rede acesso.
2. No segundo nível ou rede local, cada rede de acesso formada no primeiro nível representa uma localidade para a formação de um conjunto maior, onde todo o tráfego é coletado como em uma rede local e enviado para a rede backbone.
3. No terceiro nível ou rede backbone, ocorre a transmissão de tráfego entre as redes locais, bem como entre o tráfego internacional.

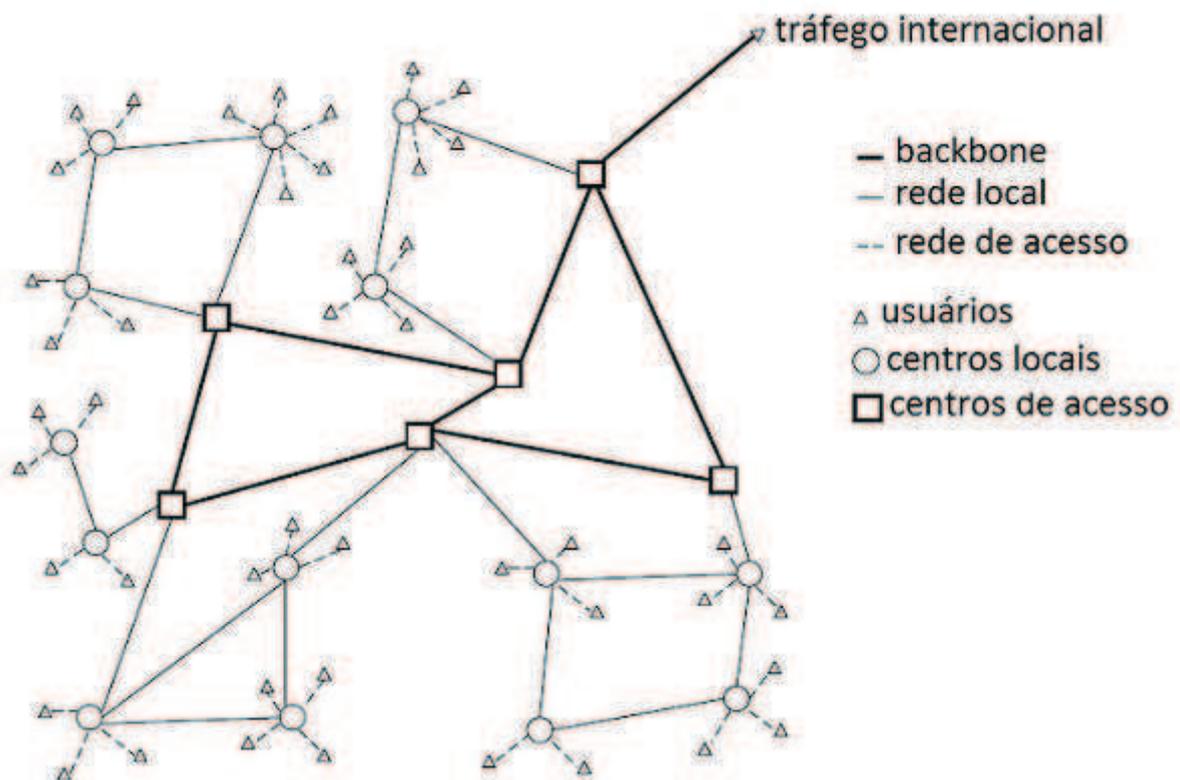


Figura 2 – Três níveis da rede

Em relação aos possíveis problemas que podem ocorrer em uma estrutura de rede de telecomunicações, os principais referem-se a restrição de demanda de tráfego e a capacidade de sobrevivência da rede, em caso de alguma falha em determinado local ou acidental rompimento de uma fibra de transmissão de dados. Esses possíveis problemas afetam operações de todos os níveis da rede, então, com uma estrutura bem desenvolvida e preparada para as falhas citadas, os danos serão bastante reduzidos em um possível acidente na rede (Vasseur, J. P.; Pickavet, M.; Demeester, P., 2004).

2.2 O PADRÃO SONET/SDH

O SONET (*Synchronous Optical NETWORK*), traduzido para o português como Rede Ótica Síncrona, é um padrão para transporte de comunicações ótimas que foi formulado pela ECSCA (textitExchange Carriers Standards Association - Associação de Padrões de Intercâmbio de Portadoras) para o ANSI (*American National Standards Institute - Instituto Nacional Americano de Padronização*), responsável por definir padrões da indústria nos Estados Unidos. Apesar de ser um padrão americano, o SONET também é usado no Canadá. Já o SDH (*Synchronous Digital Hierarchy*), traduzido para o português como Hierarquia Digital Síncrona, é o equivalente ao SONET para o mundo e foi formulado pela ITU (*International Telecommunication Union - União Internacional de Telecomunicações*). Devido a essa desproporcionalidade de uso, o SONET acaba sendo considerado como uma variação do SDH, mesmo tendo sido criado antes.

O padrão SONET/SDH define níveis de portadora ótica (*Optical Carrier - OC*) e seu equivalente elétrico (*Synchronous Transport Signal - STS*, no SONET, e *Synchronous Transport Module - STM*, no SDH) para a hierarquia de transmissão baseada em fibra ótica. Como diferença do seu antecessor, o PDH (*Plesiochronous Digital Hierarchy*), ele possui as suas taxas de transmissão sincronizadas por toda a rede por meio de relógios atômicos. Devido a esta sincronização, há redução de necessidade de *buffers* entre os elementos na rede. Os *buffers* são memórias utilizadas para armazenar temporariamente os dados de saída ou de chegada de um elemento na rede, trabalhando como um controlador de dados e estabelecendo um equilíbrio quando a diferença entre as taxas de dados que são recebidos em relação aos que podem ser processados (Perros, H. G., 2005).

O SONET traz maiores flexibilidades de configuração e disponibilidade de banda, proporcionando vantagens significativas em relação aos antigos sistemas de telecomunicações, tais como: redução dos requisitos dos equipamentos e aumento da confiabilidade da rede, possibilidade de compatibilidade de equipamentos de fabricantes diferentes (anteriormente incompatíveis), maior escalabilidade de arquitetura possibilitando acomodar futuras aplicações.

Estima-se que o modelo completo do SONET proporcione a infraestrutura de transporte adequada para as telecomunicações por pelo menos as próximas duas ou três décadas (Perros, H. G., 2005).

2.2.1 O Sinal SONET/SDH

O Padrão SONET estabelece uma tecnologia que possibilita a transmissão de sinais de diferentes capacidades através de uma hierarquia ótica síncrona e flexível, que é obtida por meio de um esquema de multiplexação por intercalação de *bytes*. Isto simplifica a multiplexação e oferece um gerenciamento de ponta a ponta para as redes.

No processo de multiplexação do SONET/SDH, o primeiro passo é a geração do sinal de nível mais baixo ou sinal base. Originalmente no SONET esse sinal é o STS-1 (*Synchronous Transport Signal level - 1*, em português é Sinal Síncrono de Transporte nível 1), que opera a 51,84 Mb/s. Os sinais de níveis mais altos são os múltiplos de STS-1, originando a família STS-N (ver Tabela 1), ou seja, um sinal STS-2 é composto por dois STS-1 juntos. A Tabela 1 também apresenta o equivalente ótico para cada sinal STS-N, chamado de OC-N (OC - *Optical Carrier*, Portadora Ótica).

Na Tabela 1 são apresentados os sinais do SONET, do SDH e sua equivalência, seguido pela largura de banda da carga útil (informação transportada) e a taxa total da linha de transmissão (Pioro, M.; Medhi, D, 2004).

Nível de Portadora Ótica para o SONET	Formato do Quadro para o SONET	Formato do Quadro para o SDH	Largura de Banda "Carga Útil" (Payload) (kbit/s)	Taxa de Linha (kbit/s)
OC-1	STS-1	STM-0	48.960	51.840
OC-3	STS-3	STM-1	150.336	155.520
OC-12	STS-12	STM-4	601.344	622.080
OC-24	STS-24	STM-8	1.202.688	1.244.160
OC-48	STS-48	STM-16	2.405.376	2.488.320
OC-96	STS-96	STM-32	4.810.752	4.976.640
OC-192	STS-192	STM-64	9.621.504	9.953.280
OC-768	STS-768	STM-256	38.486.016	39.813.120
OC-1536	STS-1536	STM-512	76.972.032	79.626.120
OC-3072	STS-3072	STM-1024	153.944.064	159.252.240

Tabela 1 – SONET/SDH Designações e Larguras de Banda

2.2.2 Elementos da Rede SONET

Uma rede SONET deve conter pelo menos os seguintes elementos:

Regenerador: necessário para regenerar o nível de um sinal de volta para o nível original. A regeneração se torna necessária quando um sinal é propagado por longas distâncias e devido a isso, tem quedas em seu nível.

Multiplexador Terminal: atua no primeiro nível da rede concentrando sinais digitais de baixo nível. Em sua forma mais simples de utilização, envolve dois multiplexadores terminais ligados por fibra ótica com ou sem presença de regenerador. Forma mais simples de implementação de uma rede SONET.

Multiplexador Add/Drop (ADM): é um multiplexador que agrega e segregava o tráfego da rede, fazendo com que opere apenas nos níveis necessários, atuando no segundo nível da rede. Um multiplexador/demultiplexador de um estágio pode multiplexar várias entradas em um sinal OC-N. Em um local que possua um ADM, somente os sinais que precisam ser acessados são retirados ou inseridos. O tráfego restante persiste através do elemento de rede sem requerer nenhum equipamento especial para isso ou outro processamento do sinal.

Tomando por base aplicações rurais como exemplo, um ADM pode ser utilizado em um local terminal ou qualquer localização intermediária para consolidar o tráfego de diferentes localizações muito distantes. A utilização de muitos ADM's pode ser configurada como um anel com capacidade de regeneração, o anel SONET/SDH. O SONET/SDH permite retirar e repetir (*drop and repeat*) o sinal que está sendo transmitido (capacidade chave em aplicações de telefonia e TV a cabo), o sinal chega a um nó, é retirado para ser utilizado e então é repetido e enviado para o próximo nó. Em caso da conexão entre os nós não puder ser realizada, o sinal é repetido e passado através de uma rota alternativa para o nó de destino.

Em aplicações de distribuição com vários nós, um canal de transporte pode transportar tráfego eficientemente entre múltiplos nós de distribuição. Quando transportando vídeo, por exemplo, cada canal de programação é entregue no nó e repetido para ser entregue nos próximos e subsequentes nós. A banda (canais de programação de TV) não precisa ser totalmente entregue em todos os nós, pois canais que não terminam em um nó pode atravessar sem intervenção física deste nó.

O ADM proporciona interfaces entre os diferentes sinais de rede e os sinais SONET. Na localidade ao qual pertence, o ADM pode retirar sinais de taxa baixa para serem transportados em diferentes equipamentos ou pode adicionar sinais de taxa baixa dentro de sinais de STS-N de taxa mais alta. O resto do tráfego simplesmente passa direto. Embora elementos de redes sejam compatíveis no nível de portadora ótica, eles podem diferir em características de fabricante para fabricante. O padrão SONET/SDH não restringe fabricantes a um único tipo de produto, nem requer que eles fabriquem todos os tipos. Ou seja, podem existir ADMs diferentes (com acesso a níveis de sinais diferentes).

Digital Cross Connect (DXC): um DXC SONET/SDH aceita várias taxas de

portadoras óticas, acessa os sinais STS-1 e efetua mudanças neste nível, sendo idealmente utilizado como um *hub* SONET/SDH. Um DXC opera de forma semelhante ao ADM e a maior diferença entre os dois é que o DXC pode ser usado para interconectar um número maior de sinais STS-1. O DXC pode ser usado para agregar e segregar sinais STS-1 ou para o gerenciamento de tráfego de banda muito larga (*broadband*). Como exemplo, pode ser utilizado para segregar tráfego de alta largura de banda de tráfego de baixa largura de banda e enviá-los separadamente através de dispositivos de alta largura de banda (como vídeo) e de baixa largura de banda (como voz) (Vasseur, J. P.; Pickavet, M.; Demeester, P., 2004; Perros, H. G., 2005; Pioro, M.; Medhi, D., 2004).

2.2.3 Topologia do Padrão SONET/SDH

Existem várias topologias de rede disponíveis para serem utilizadas, a escolha depende da necessidade visualizada pelo projetista. No caso do padrão SONET/SDH, as topologias base são: ponto a ponto, ponto multiponto, barramento e anel.

Mesmo existindo várias topologias para uma rede SONET/SDH, a anel é tipicamente utilizada. Ela consiste em um conjunto de clientes conectados através de um ou mais anéis. Como exemplo, cada cliente está vinculado a um único anel e envia, recebe e transmite sinais por meio de um dispositivo ADM. Vários ADMs podem ser dispostos em uma configuração anelar, possibilitando tanto para o tráfego unidirecional quanto bidirecional. A topologia em anel tem como principal vantagem, a capacidade de sobrevivência da rede em caso de alguma falha ou rompimento de algum cabo de fibra ótica. Em caso de rompimento de um cabo de fibra ótica ou falha, os multiplexadores enviam automaticamente os serviços afetados por um caminho alternativo. A Figura 3 mostra o uso de um caminho alternativo para o caso de falha ou rompimento de uma fibra ótica, no qual o fluxo nos dois sentidos passa a ser ativado. Sendo assim, não haverá interrupção dos serviços de comunicação (Fortz, B; Soriano, P.; Wynants, C., 2003).

Como forma de realizar comunicação entre os anéis, um dispositivo DXC, liga os anéis locais a um anel especial, denominado de Anel Federal (AF).

O tráfego em um anel local é limitado pela largura de banda dos ADMs utilizados. Essa limitação deve ser suficiente para acomodar a soma das demandas de todas os clientes pertencentes ao anel local. Já no caso do anel federal, a quantidade máxima de tráfego é limitada pela largura de banda dos DXCs utilizados e essa capacidade deve acomodar a soma das demandas dos anéis. Os valores de largura de banda dos ADMs e DXCs correspondem às taxas de *bits* dos sinais SONET (ver Tabela 1).

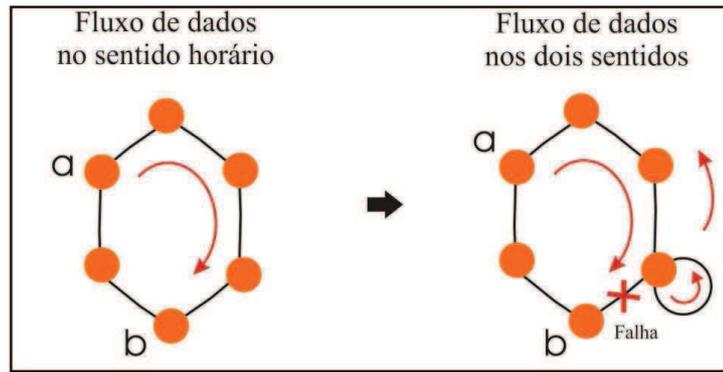


Figura 3 – Capacidade de sobrevivência com uma topologia em anel

Na Figura 4 a seguir, é mostrada uma rede SONET/SDH utilizando topologia em anel. Cada ADM conecta um cliente a rede através de um anel local. O tráfego entre os anéis locais é feito através dos DXSs que compõem o anel federal.

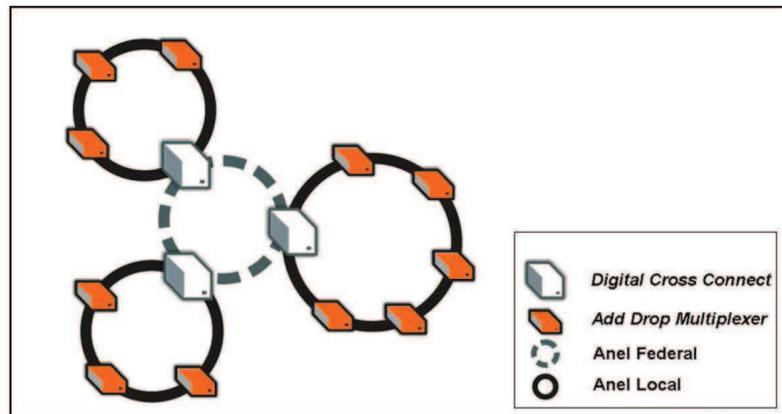


Figura 4 – Rede SONET/SDH com topologia em anel

3 DESCRIÇÃO DO PROBLEMA E MODELAGEM MATEMÁTICA

Neste capítulo a primeira seção segue com uma descrição formal do PALAS e suas restrições. Ainda neste capítulo há uma seção com o modelo matemático utilizado para tratar o PALAS.

3.1 DESCRIÇÃO DO PROBLEMA

O problema PALAS surge na fase de planejamento físico de uma rede *backbone*. Sendo o PALAS modelado por um grafo não orientado $G = (V, E)$, onde V é o conjunto de n vértices e E é o conjunto de m arestas. Portanto, o problema de partições em grafos consiste em dividir V em conjuntos disjuntos tais que $V = V_1 \cup V_2 \cup \dots \cup V_l$ (referência Souza, 2002) enquanto que a aresta $[i, j] \in E$ representa a demanda d_{ij} correspondente. O anel, é um conjunto de clientes $i \in V$, cuja capacidade total $Q \in \mathbb{N}^*$ é limitada, este anel é denominado de anel local. O PALAS tem por objetivo obter uma partição de menor tamanho sobre V respeitada sua capacidade Q onde cada elemento da partição representa um anel local. Um outro anel, conhecido como anel federal e que também possui capacidade Q é responsável por interligar todos os anéis locais (Arighieri, R.; Dell'Amico, M, 2005; Goldschmidt, O.; Laugir, A.; Olinick, E. V., 2003; Macambira, E. M.; Maculan, N.; de Souza, C. C., 2006).

De acordo com (Goldschmidt, O.; Laugir, A.; Olinick, E. V., 2003) o PALAS é NP-difícil. A formação do anel federal ocorre pela ligação dos anéis locais, através dos aparelhos DXC, onde cada aparelho destes representa um anel local. Pelo alto custo dos DXC, o problema trata em minimizar o número de anéis locais. As restrições impostas pelo problema são as seguintes:

1. Cada cliente deve ser atribuído a um único anel.
2. Demanda entre cada par de clientes deve respeitar a capacidade do anel.
3. A ligação entre cada anel representado por um DXC ocorre por meio de um anel federal.

É importante lembrar que uma rede SONET/SDH também pode possuir outras topologias, mas o fato do agrupamento formando anéis, onde cada um deles é representado por um DXC e somado a capacidade de sobrevivência de uma rede com a topologia em anel, torna-se a escolha mais viável.

3.2 MODELO MATEMÁTICO

Diversas formulações matemáticas compactas e estendidas foram propostas para o SRAP na literatura, (Goldschmidt, O.; Laugir, A.; Olinick, E. V., 2003; Macambira, E. M.; Maculan, N.; de Souza, C. C., 2006). No nosso conhecimento, a formulação compacta de melhor desempenho foi proposta por (Macambira, E. M.; Maculan, N.; de Souza, C. C., 2006). A formulação (1) - (12) utiliza o grafo $G = (V, E)$ previamente definido, demandas de tráfego $d_{ij} > 0$ entre cada par de vértice $i, j \in V$. O tráfego total de demandas é dado por $D = \sum_{[i,j] \in E} d_{ij}$. $Q \in \mathbb{N}^*$ corresponde a capacidade total (banda passante) dos anéis locais e federal. O modelo matemático utiliza variáveis y^r que determinam se o anel r é usado ($y^r = 1$), ou não ($y^r = 0$) e as variáveis x_i^r estabelecem se um vértice i é alocado no anel r ($x_i^r = 1$), ou não ($x_i^r = 0$). Variáveis de decisão auxiliares z_{ij}^r indicam se pelo menos uma das extremidades da aresta $[i, j]$ pertence ao anel i ($z_{ij}^r = 1$), ou não ($z_{ij}^r = 0$).

$$\max \sum_{r=1}^n \sum_{i \in V} n x_i^r - \sum_{r=1}^n y^r \quad (1)$$

$$\sum_{[i,j] \in E} d_{ij} z_{ij}^r \leq Q \quad \forall i = 1 \dots n \quad (2)$$

$$\sum_{r=1}^n \sum_{[i,j] \in E} d_{ij} z_{ij}^r \leq D + Q \quad (3)$$

$$\sum_{r=1}^n x_i^r \leq 1 \quad \forall i \in V \quad (4)$$

$$x_i^r \leq y^r \quad \forall i \in V, \forall r = 1 \dots n \quad (5)$$

$$z_{ij}^r \geq x_i^r \quad \forall [i, j] \in E, \forall r = 1 \dots n \quad (6)$$

$$z_{ij}^r \geq x_j^r \quad \forall [i, j] \in E, \forall r = 1 \dots n \quad (7)$$

$$z_{ij}^r \leq x_i^r + x_j^r \quad \forall [i, j] \in E, \forall r = 1 \dots n \quad (8)$$

$$y^r \leq y^{r-1} \quad \forall r = 2 \dots n \quad (9)$$

$$x_i^r \in \{0, 1\} \quad \forall i \in V, \forall r = 1 \dots n \quad (10)$$

$$y^r \in \{0,1\} \quad \forall r = 1 \dots n \quad (11)$$

$$z_{ij}^r \in \{0,1\} \quad \forall [i,j] \in E, \forall r = 1 \dots n \quad (12)$$

A função objetivo (1) estabelece a maximização do número de clientes incluídos nos anéis e a minimização da quantidade de anéis. As restrições (2) e (3) asseguram, respectivamente, o respeito da capacidade dos anéis locais e federal. As inequações (4) garantem que cada vértice seja alocado a no máximo um anel, enquanto que as restrições (5) evitam que vértices sejam alocados a anéis que não existem. Variáveis x e z são conectadas através das restrições (6), (7) e (8). As inequações (9) reduzem o fenômeno de simetria das variáveis y , evitando que um anel seja aberto, se seu predecessor encontra-se fechado. A definição das variáveis é realizada de (10) a (12).

4 TRABALHOS RELACIONADOS

O presente trabalho traz uma ampla revisão bibliográfica e com o fim suprir o conhecimento necessário ao problema, entre eles podem ser citados: (Lee, Y et al., 2000; Goldschmidt, O.; Laugir, A.; Olinick, E. V., 2003; Oliveira, W., 2010; Soares, W., 2008; Girão, A. C. S., 2008; Silva, A.C.G. et al., 2009; Bastos, L. D. O.; Ochi, L. S.; Macambira, E. M., 2005a; Bastos, L. D. O.; Ochi, L. S.; Macambira, E. M., 2005b; Bastos, L. O., 2005). Em todos eles foram utilizadas as instâncias da literatura conhecidas como: GLO, AD, LSHK e RTNR, referenciadas respectivamente como C1, C2, C3 e C4.

Em (Arighieri, R.; Dell'Amico, M, 2005) foram utilizadas quatro funções objetivo Z_1 , Z_2 , Z_3 e Z_4 , sendo três delas existentes na literatura e a quarta introduzida por eles, tendo como objetivo reduzir a geração de soluções inviáveis dentro de um conjunto de soluções. Isso permite que algumas soluções com números iguais de anéis, onde foram dadas como inviáveis, sejam possíveis de serem aproveitadas, utilizando a baixa carga desses anéis como segundo fator avaliativo.

Os mesmos autores também utilizaram quatro procedimentos heurísticos para a elaboração de uma melhor solução inicial, sendo os dois primeiros procedimentos de bin-packing como *Best-Fit Decreasing* (BFD), *Next-Fit* (NF) e os outros dois são o *Clique-BF* e uma versão modificada do *Cycle-BF*, onde é relaxada a condição de achar um clique completo.

Com uma boa solução inicial formulada, eles fizeram uso de seis metaheurísticas para a comparação de resultados. As metaheurísticas são: Busca Tabu Simples, duas diferentes implementações de Path-Relinking embutidas na Busca Tabu Simples (BTS), eXploring Busca Tabu, Scatter Search (SS) e Diversification by Multi-Neighborhood (DMN).

As duas variações de Path-Relinking são chamadas de *PR1* e *PR2*. Na *PR1* é gerado o menor caminho entre a solução atual e a melhor solução, dado um determinado conjunto limite de movimentos para alcançar o objetivo. Na *PR2* é feito o mesmo processo e em adicional, as soluções geradas pela *PR1* passam a ser do conjunto elite de soluções. Tudo isso com a finalidade de gerar melhores soluções iniciais para a Busca Tabu Simples.

A técnica DMN faz uso de mais de uma vizinhança, onde as vizinhanças a mais são utilizadas para a diversificação de soluções. Uma vizinhança *N1* move um cliente de um anel para outro e a vizinhança *N2* realiza uma troca entre dois clientes de anéis distintos em dois movimentos, onde ele move um cliente para outro anel provavelmente se tornando inviável e no segundo movimento ele restauraria essa viabilidade.

O trabalho de (Macambira, E. M.; Maculan, N.; de Souza, C. C., 2006) trata o problema do SONET/SDH fazendo uso de um método híbrido para resolução do mesmo através do Branch-and-Price (B&P). Primeiramente são comparadas quatro

formulações matemáticas, sendo elas resolvidas pelo Branch-and-Bound (B&B), duas pertencentes a literatura do problema e outras duas foram criadas pelos autores. Após a comparação, por exaustivos testes, é selecionada a melhor entre elas e comparada com outra formulação proposta através do B&P.

As formulações criadas por Macambira e Maculan para a resolução via B&B são as *Packing Model with Less Symmetry* (SM) e o *Compact Model* (CM). O (SM) é fortemente baseado em uma formulação já existente na literatura, o *Packing Model* (PM), que também é baseado em outro modelo, o (GM), onde a diferença consiste na relaxação das variáveis. A diferença entre as duas formulações se resume em uma redução de simetria. Esta simetria faz referência a anéis r com representação inteira $\binom{n}{r}r!$, fazendo com que os anéis sejam indistinguíveis, pois são enumerados, e consequentemente prejudique o desempenho do algoritmo.

A formulação (CM) é originada da formulação de Goldschmidt, no qual ele segue o princípio de reduzir as variáveis sem diminuir o número de restrições e mantendo a função objetivo. Ainda no (CM), ele aproveita outra restrição de outro modelo da literatura, onde adiciona uma restrição que venha a diminuir a simetria.

Outra formulação proposta pelo autores é a *Set-Partitioning Model* (SP). Nesta formulação o problema original é dividido em vários níveis e uma restrição de mochila é adicionada para representar a restrição de capacidade de um anel. A alocação dos anéis federais é definida como problema mestre, a alocação nos anéis locais como problema mestre reduzido e a restrição da mochila como subproblema escravo.

Na implementação do B&P, o *Set-Partitioning* (SP) e as estratégias dos autores se baseiam em: conjunto inicial de colunas, seleção de vértices, regras de branch, detecção de taling off e a inclusão ou não de várias colunas.

O problema mestre trata da seleção da melhor combinação de anéis e faz uso de duas restrições: cobertura de vértices e capacidade do anel federal. Já o problema escravo objetiva a redução dos custos dos anéis.

Este trabalho foi significativamente importante, tendo este fornecido o melhor modelo matemático para a resolução do PALAS e também foi capaz de comprovar a inviabilidade de várias instâncias.

Para criar limites primários e otimizar o processamento do B&P, o GRASP é executado em toda inicialização do B&P para cada instância. O GRASP tem por objetivo fornecer um melhor conjunto de soluções iniciais, na fase de construção da solução, modificando sua função objetivo para penalizar soluções inviáveis e seu processo de construção é baseado na heurística *edge-based*. Na etapa de busca local, uma vizinhança é responsável pela troca de um cliente de um anel por outro enquanto na segunda vizinhança um par de clientes é trocado entre dois anéis distintos.

Em (Bastos, L. D. O.; Ochi, L. S., 2008), a resolução do problema do SONET foi por meio de Algoritmos Genéticos (AG), e, para intensificar ainda mais a busca por

melhores resultados, eles também utilizaram AG com Path-Relinking Evolucionário (PRE).

A geração da população inicial é feita por dois processos de heurísticas construtivas, a *Random Node-Base* (RNB) e *Relative Neighborhood* (RN). Ambos os procedimentos geram soluções iniciais selecionando aleatoriamente os clientes para compor uma solução. Cada heurística é executada n vezes até atingir o tamanho da população desejada, onde cada processo gerará metade da população. No AG proposto, o cromossomo é um vetor positivo onde cada gene representa um cliente e informando em qual anel encontra-se o cliente.

Partindo para uma melhor solução inicial, Bastos e Ochi utilizaram o Path-Relinking para transformar soluções ótimas locais S_1 em soluções elite S_2 . Nas execuções do PR no problema do SONET/SDH, foi observado que o PR teria muito mais chance de sucesso, se o par de soluções possuísse a mesma quantidade de anéis. Por outro lado, quanto maior o caminho entre as soluções, mais chances de se obter um ótimo local totalmente diferente, dando diversidade ao conjunto soluções.

Para garantir o mesmo número de anéis nas soluções S_1 e S_2 , eles elaboraram três condições onde comparam o número de anéis das soluções S_1 e S_2 e, reduzem a quantidade de anéis da solução com maior número, igualando o número de anéis das soluções.

Já o PRE começa atuando sobre as soluções elite originadas da melhor população criada pelo AG. Então o PRE combina cada solução elite com as soluções geradas pelo PR, formando um novo conjunto de soluções elite. Todo esse processo se repetirá até que o conjunto de soluções geradas não tenha melhora se comparado ao último conjunto de soluções.

O trabalho de (Bernardino, E. M. et al., 2012) resolve as duas abordagens do problema do SONET/SDH usando um algoritmo híbrido de Honey Bees Mating Optimisation (HBMO) e compara os resultados presentes na literatura, em especial com outro algoritmo feito pelos mesmo autores em (Bernardino, E. M. et al., 2010). Para as funções fitness, foram utilizadas as já presentes na literatura e criadas por (Pelleau, M.; Van Hentenryck, P.; Truchet, C., 2009).

Para geração da população inicial, (Bernardino, E. M. et al., 2012) optaram por utilizar um método determinístico, utilizando um algoritmo também feito por eles, o HSS. Com a população gerada, se faz necessária a avaliação das soluções. Uma função objetivo básica Z_0 conta o número de anéis para a abordagem baseada em DXC e o número de ADMs para a outra abordagem. A função Z_5 minimiza a função Z_0 , reduzindo o número de anéis. Caso a solução gerada ultrapasse a carga do anel, ela é penalizada, sendo preferível uma solução com maior número de anéis do que uma inviável. Esse processo de avaliação é a etapa mais custosa do algoritmo.

Com as soluções avaliadas, inicia-se o processo de acasalamento. Neste processo,

a rainha (melhor solução) usa uma regra probabilística determinada por uma equação para a escolha do zangão (solução aleatória). Sendo sucesso, a rainha armazena o esperma do zangão e continua o processo de acasalamento até o limite de iterações.

O processo de acasalamento cria novas soluções baseadas na combinação da melhor solução com uma aleatória, sendo essas soluções são conhecidas por prole. A prole criada é melhorada por duas etapas de busca local. No caso do DCS, a primeira etapa de busca local é feita em toda a vizinhança, onde um cliente aleatório pertencente a um anel, é jogado para o anel com menor quantidade de tráfego. A solução não sendo boa, é tentado com dois clientes. Se ainda não for melhor, um cliente qualquer pertencente ao anel com maior tráfego é jogado para o anel com menor tráfego. Isso tenta equilibrar a carga dos anéis sem criar um novo. A segunda etapa ocorre em cima da solução melhorada pela primeira, sendo que busca local ocorre apenas em parte da vizinhança e usa dois métodos. No primeiro método ocorre a mesma troca de clientes através de anéis, porém a diferença é que os dois anéis são aleatórios. No segundo método ocorre a troca de dois clientes simultaneamente entre dois anéis.

Na abordagem por ADMs, a primeira etapa ocorre com trocas de arestas entre partições diferentes na tentativa de equilibrar as partições. Na segunda etapa existem quatro métodos, um por troca de clientes em anéis aleatórios, o segundo ocorre através da troca simultânea de dois clientes entre dois anéis e outros dois métodos. Nos demais, o primeiro seleciona aleatoriamente um cliente da partição com maior tráfego e o atribui a outro anel qualquer de outra partição, enquanto que no segundo um cliente aleatório que não pertence a partição com menor tráfego é atribuído a ela.

Após todos esses processos, a prole criada é melhorada e comparada com a rainha, se alguém da prole for melhor que a rainha, a rainha é substituída e os remanescentes da prole serão os novos zangões e novamente ocorrerá o processo de acasalamento e melhoria. O algoritmo para quando se atinge o número limite de acasalamentos.

Em relação a resultados, o trabalho (Bernardino, E. M. et al., 2012) foi o que obteve os melhores resultados em questão de tempo de processamento e qualidade de solução.

5 ABORDAGENS EXATAS E HEURÍSTICAS

A otimização combinatória e seus problemas difíceis de serem resolvidos, nos obriga a aceitar soluções aproximadas. Problemas computacionalmente difíceis são conhecidos como NP-difícil e o tempo para obter uma resposta exata pode ser praticamente inviável, mesmo com o uso de computação paralela, através dos poderosos clusters.

Grande parte dos problemas de otimização combinatória de relevância prática, são NP-difícil e considerados intratáveis. Mesmo que até hoje não exista prova concreta de que a classe NP-difícil possa ser tratada por algoritmos de tempo polinomial, existe fortes evidências que eles possam ser apenas resolvidos por algoritmos aproximativos. Problemas como: caixeiro viajante, bin packing, coloração mínima de grafos, escalonamento de tarefas, fluxo em redes e diversos outros (Ausiello, G. et al., 2003).

Nesse capítulo serão mostradas duas abordagens de resolução utilizadas no PALAS, por algoritmos evolutivos e por programação dinâmica.

5.1 PROGRAMAÇÃO DINÂMICA

A programação dinâmica é baseada no método dividir para conquistar que consiste em resolver um problema através da combinação de soluções dos seus subproblemas. É um método exato utilizado para a resolução de problemas de programação inteira envolvendo decisões sequenciais, onde uma decisão depende apenas do estado atual do sistema.

Algoritmos com esta abordagem, particionam o problema em subproblemas disjuntos, resolvem os subproblemas através de recursividade e, em seguida combina as soluções dos subproblemas a fim de resolver o problema original. Para podermos aplicar a programação, deve haver sobreposição de subproblemas, ocorre quando os subproblemas dividem subsubproblemas. Ao comparar o método da divisão e conquista com a programação dinâmica, é visível a vantagem da última. Na divisão e conquista os subproblemas são resolvidos repetidamente, sendo assim um esforço desnecessário, diferente do que ocorre na programação dinâmica, onde cada subproblema é resolvido apenas uma vez e o seu resultado é armazenado em uma tabela, evitando assim um novo cálculo (Cormen, T. H. et al., 2009; Levitin, A., 2011).

Desenvolver um algoritmo de programação dinâmica envolve os seguintes passos:

1. Caracterizar a subestrutura de uma solução ótima.

2. Definir recursivamente uma solução ótima, sobreposição de subproblemas.
3. Armazenar o valor de uma solução de um subproblema.
4. Construir uma solução ótima a partir das soluções já computadas.

5.1.1 Elementos da programação dinâmica

Para aplicação da programação dinâmica em problemas de otimização é necessário que existam os elementos fundamentais, sendo eles:

- **Subestrutura ótima:** Um problema de otimização que possui uma subestrutura ótima, também conhecido como o **princípio de otimalidade** de Bellman, já se torna um forte indício para o uso da programação dinâmica, caso contrário, a programação dinâmica pode ser ineficiente ou a modelagem do problema pode está errada. Uma subestrutura ótima é quando a solução ótima para o problema contém soluções ótimas dos subproblemas. Deve-se garantir que a solução ótima do problema original usa as soluções ótimas dos subproblemas.

Uma subestrutura ótima varia entre o domínio do problema de duas formas:

1. A quantidade de subproblemas usados para a solução ótima do problema;
2. A quantidade de escolhas disponíveis na determinação de quais subproblemas serão usados na solução ótima.

Em geral, a programação dinâmica usa a subestrutura ótima com estratégia *bottom-up*, determinando assim as soluções ótimas dos subproblemas para só então determinar a solução ótima do problema (Skiena, S. S.; Revilla, M. A., 2003).

- **Sobreposição de subproblemas:** Neste caso o espaço de subproblemas deve ser pequeno no sentido de que a recursividade deve resolver os subproblemas existentes, ao invés de gerar novos subproblemas, diferente do que ocorre em divisão em conquista, onde cada novo problemas é gerado a cada recursão ou ao chamar novamente o problema. Na figura 5 existe um exemplo gráfico da sobreposição de subproblemas, através do exemplo da multiplicação de cadeia de matrizes. Geralmente o número de subproblemas é dado por um polinômio do tamanho da entrada do problema. Quando um algoritmo recursivo visitar o mesmo problema várias vezes, é que ele possui subproblemas para serem resolvidos, este processo é o que dá origem ao nome deste elemento, sobreposição de subproblemas. A solução desses subproblemas ficam armazenada para uma eventual consulta posterior.

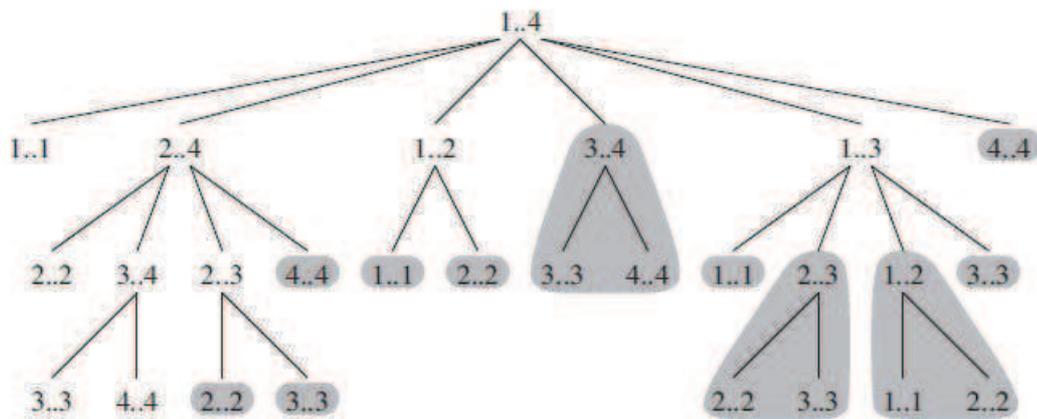


Figura 5 – Sobreposição de subproblemas, através do exemplo do problema da multiplicação de cadeia de matrizes. Cada nó representa parâmetros i e j .

- **Memoization:** Trata-se de uma tabela contendo as soluções dos subproblemas e uma estrutura de controle para preencher a tabela. Para cada solução de um subproblema existe uma entrada correspondente na tabela. Quando um subproblema ainda não tem sua solução computada na tabela, usa-se um valor especial para determinar que aquele subproblema ainda não tem uma solução computada. Esse armazenamento da solução serve para evitar um novo cálculo de uma solução, sendo necessário apenas utilizar o valor computado na tabela para o determinado subproblema (Cormen, T. H. et al., 2009).

5.1.2 Estratégias

A programação dinâmica funciona através do uso de memória, pois como trabalha de forma recursiva, um subproblema será visitado mais de uma vez. Assim, a solução de um subproblema é armazenada em um vetor, outra estrutura de dados necessária para o problema, isso evita um novo cálculo do subproblema, necessitando apenas consultar o valor armazenado. Esse método gera um *trade-off* entre tempo e memória, pois a programação pode exigir mais pela memorização, mas será de maior eficiência em relação ao tempo gasto, pois não precisará recalcular nada, somente consultar. Através do uso da memória, a programação dinâmica é capaz de transformar um problema de tempo exponencial em tempo polinomial. Sua execução é de tempo polinomial quando a entrada for de tamanho polinomial. Para efeitos comparativos, um algoritmo recursivo possui complexidade assintótica de $\Omega(2^n)$, enquanto que um algoritmo com programação dinâmica tem complexidade assintótica de $O(n^2)$.

- **Top-down:** Neste tipo de abordagem o procedimento recursivo funciona de forma

natural, seguindo do problema original até a menor porção do subproblema. Ao chegar no subproblema, uma consulta na tabela é feita a fim de verificar se o mesmo já foi resolvido, caso tenha sido, apenas utiliza o resultado, caso não, realiza a computação do subproblema.

- **Bottom-up:** Esta abordagem necessita ter uma noção do tamanho do problema e dos seus subproblemas. Qualquer resolução de um subproblema especial, deve depender apenas de resolver subproblemas menores. O procedimento necessita de uma classificação dos subproblemas para que possam ser resolvidos de forma ascendente e uma vez que precisarmos resolver os de hierarquia maior.

Em ambas as abordagens a complexidade assintótica em relação ao tempo é igual. Em casos especiais a abordagem *top-down* não necessita resolver todos os subproblemas, diferente do que ocorre na abordagem *bottom-up*, onde o algoritmo percorre todos os subproblemas. Em relação a qual abordagem utilizar, afirma-se que a segunda possui fatores mais constantes, sobrecarregando menos nas chamadas de procedimentos (Skiena, S. S.; Revilla, M. A., 2003).

5.2 ALGORITMOS EVOLUTIVOS

Essa classe de algoritmos é baseada no princípio da seleção natural, proposta por Darwin em 1858. Este princípio trás a ideia da evolução das espécies, onde as mesmas sofrem várias mudanças genéticas, decorrente das adaptações necessárias para sobreviver (Darwin, C., 1999).

O neo-darwinismo traz alguns preceitos tidos como básicos para o processo evolutivo:

- Indivíduos, sejam eles da mesma ou de diferentes espécies, disputam por recursos limitados no meio ambiente;
- Alguns indivíduos têm vantagem sobre outros em determinados ambientes, isso devido a características que os favorecem;
- Indivíduos mais adaptados ou com vantagens em determinadas situações, têm maior probabilidade de sobreviverem e conseqüentemente se reproduzirem;
- No processo reprodutivo, os filhos sempre irão herdar características de seus pais, tendenciando os filhos a nascerem mais adaptados. Essas características são passadas por gerações subsequentes;

- No decorrer do processo evolutivo, as características mais necessárias para adaptações tendem a serem propagadas nas espécies, lhes dando um maior grau de adaptação.
- O propagar das características pelas espécies possibilita o ocorrer de um fenômeno conhecido como mutação. Tal fenômeno poderia ser prejudicial ou benéfico; prejudicial no sentido de que esta mutação poderia lhe acarretar em um defeito genético, e benéfico no sentido da possibilidade de trazer mais uma característica que o favoreça na adaptação.

5.2.1 Algoritmos Genéticos

Os algoritmos genéticos, baseados no processo de seleção natural de Darwin, se destinam à resolução de uma gama de problemas, sendo em especial os problemas de otimização. Por sua robustez e facilidade de alteração/adaptação para resolver um problema, tornou-se uma técnica fortemente estudada e utilizada em diversas áreas do campo da otimização.

De acordo com (Holland, J. H., 1992) os algoritmos genéticos resolvem um problema seguindo etapas baseadas no processo evolutivo que podem ser vistas na figura 6 . De forma objetiva, os algoritmos genéticos criam uma população que representa as possíveis respostas para o problema em questão, sendo cada indivíduo dessa população submetido ao processo evolutivo.

5.2.1.1 Indivíduos

Tratando-se de algoritmos genéticos, os indivíduos são conhecidos como *chromossomos* e os seus elementos ou características como *genes*. Entretanto, o valor de um indivíduo é o seu *fenótipo* que é o resultado da decodificação dos genes. Cada indivíduo tem um grau de adaptação, calculado por uma função objetivo. Também existe um grau de aptidão ou *fitness* representada na equação (1), atribuído a cada indivíduo e responsável por dizer o nível de adaptação dele em relação aos demais (Goldberg, D. E., 1989).

$$fA(x) = \frac{fO(x)}{\sum_{i=1}^n fO(i)} \quad (1)$$



Figura 6 – Etapas dos algoritmo genético.

5.2.1.2 População

O processo evolutivo sobre o qual a população está sujeita, se dá pelo uso da seleção, através de sua aptidão, e também por cruzamento e mutação. Uma geração representa a quantidade de vezes que a população passou pela execução do processo constituído por seleção, reprodução, mutação e atualização (Davis, L. D., 1991).

Cada geração tem o seu grau de convergência, representando a adaptação da atual geração em relação a sua anterior para alcançar os valores ótimos. A diversidade é uma característica de uma geração, pois é ela que amplia o espaço de busca, quanto maior a diversidade, maior o espaço de busca. Outro fator importante de cada população, são os indivíduos elites, possuem uma maior adaptação em relação aos demais.

5.2.1.3 Inicialização

A etapa de inicialização constitui-se basicamente da geração da população inicial, visando uma maior biodiversidade, algo fundamental para ampliar o espaço de busca. Existem algumas formas de gerar a população inicial, embora o meio mais comum seja uma inicialização randômica. Outra forma de gerar seria por meio de algoritmos de busca heurística com caráter gerador de população inicial (Geyer-Schulz, A., 1996; Goldberg, D. E., 1989).

Entre os mais usados, o método randômico uniforme consiste em modificar os genes do cromossomo por um elemento do conjunto de alelos¹.

5.2.1.4 Avaliação

Cada indivíduo da população deverá ser avaliado para determinar o seu grau de adaptação, ou seja, verificar o quão boa é uma resposta para o problema que esta sendo tratado.

Calcular com precisão o grau de adaptação dos indivíduos tende a ser uma tarefa complexa, somado ao fato da extrema repetição em ter que fazer isso para cada indivíduo, o processo pode ser muito custoso. Uma solução para esses casos, é o de usar funções não determinísticas, onde será avaliado apenas uma amostragem das características dos indivíduos (Geyer-Schulz, A., 1996).

5.2.1.5 Seleção

Etapa em que se avalia a adaptação dos indivíduos e julga se suas características serão propagadas por gerações, podendo ser um processo:

- **Dirigido:** A seleção não seria algo aleatório, operando de um modo determinístico. Pois um indivíduo só é capaz de sobreviver em determinado ambiente e nele se reproduzir se e somente se as suas características lhe permitirem tal adaptação.
- **Cumulativo:** São os benefícios passados de uma geração para outra. Sendo assim a seleção de indivíduos cada vez mais preparados ajudam a obter organismos mais complexos e melhores possíveis soluções.

¹ Alelos é também um termo da biologia e se refere ao conjunto de valores a qual um gene poderá assumir.

Ainda tratando-se da etapa de seleção, existe um método chamado de *scaling* que é atuante sobre o valor de adaptação de cada indivíduo, cuja finalidade é evitar que o cruzamento dos que são selecionados convirja ao ótimo prematuramente (Goldberg, D. E., 1989; Wall, M. G., 2000; Whitley, D., 1994).

Entre os métodos de seleção um dos mais usados é a seleção por giro de roleta (*roulette wheel selection*). Esse método calcula o somatório da adequação da população total, sorteia um valor i pertencente ao intervalo $[0;total]$, selecionando o indivíduo da faixa da roleta onde i esteja localizado.

5.2.1.6 Reprodução

Seguinte a etapa de seleção, na reprodução os indivíduos seguem com uma probabilidade pré-estabelecida pelo processo de cruzamento (*crossover*), combinando parte dos genes dos pais para geração de filhos.

A escolha de indivíduos para o cruzamento tanto pode ser aleatória, como pode ser através de métodos celetistas. O par de indivíduos selecionados para o cruzamento possuem a representação do cromossomo através de vetores, onde cada posição representa os genes (Geyer-Schulz, A., 1996).

Com base na representação por meio de vetores, os operadores de cruzamento são pontos de cruzamento dos genes dos pais onde originarão os filhos. Uma representação básica pode ser vista na figura 6.

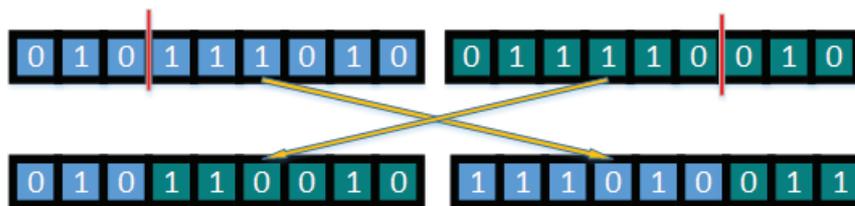


Figura 7 – Exemplo de um crossover.

5.2.1.7 Mutação

A mutação ocorre nos indivíduos resultantes do processo de cruzamento e com uma probabilidade então pré-determinada de efetuar algum tipo de alteração em sua estrutura. A importância deste operador é a diversidade que ele pode proporcionar, isto se bem escolhido (Goldberg, D. E., 1989).

5.2.1.8 Atualização

Os indivíduos originários dos processos de cruzamento e mutação serão inseridos na população. Pelo fato da população possuir tamanho fixo, os novos indivíduos ficarão no lugar dos que lhe deram origem. Outra alternativa seria a substituição apenas em caso de serem mais aptos, possuírem melhor valor *fitness* (Davis, L. D., 1991).

5.2.1.9 Finalização

A finalização é apenas um teste de condição para saber se o algoritmo genético conseguiu atingir a condição de parada. Os critérios podem ser desde o número de gerações criadas até o grau de convergência/homogeneidade da atual população.

5.2.2 Algoritmos Genéticos com Chaves Aleatórias

Algoritmos genéticos com chaves aleatórias (RKGA - *Random-Keys Genetic Algorithm*) é uma derivação dos clássicos algoritmos genéticos. Eles utilizam uma população de cromossomos que por sua vez possuem representação através de vetores reais com valores no intervalo $[0, 1]$. Cada elemento contido no vetor é denominado de *chave* e seu valor é gerado aleatoriamente na população inicial. O fator adaptativo do cromossomo é definido através do custo da solução fornecida por uma heurística que recebe o vetor de chaves do cromossomo como dados de entrada e através disso fornece uma solução viável para o problema.

Esse tipo de algoritmo genético foi proposto por (Bean, J. C., 1994). Dados dois cromossomos c_1 e c_2 , é gerado um novo cromossomo c_{new} , admitindo-se que o cromossomo c_1 seja mais apto que o c_2 , o valor de cada uma das chaves de c_{new} é herdado de forma aleatória da chave correspondente em c_1 com probabilidade p_{rec} e também da chave correspondente em c_2 com probabilidade $1 - p_{rec}$, onde p_{rec} é um parametro que é tipicamente fixado em 0,7. O algoritmo dispensa o uso de operadores tradicionais de mutação, tendo a cada geração os seus piores cromossomos na população eliminados e substituídos por novos cromossomos que foram gerados aleatoriamente.

Em cada nova geração, a população atual é particionada em 3 (três) conjuntos distintos, que são: *Elite*, *Não-elite* e *Mutante*. Cromossomos mais aptos, ou seja, com melhor *fitness*, são mantidos no conjunto *Elite*, os que são menos aptos no conjunto *Mutante* e por consequência o restante dos cromossomos são mantidos no conjunto

Não-elite. Com relação ao tamanho da população e a cardinalidade dos conjuntos, são parâmetros que devem ser ajustados.

Como pode ser visto na figura 8, os cromossomos do conjunto *Elite* são mantidos, sendo copiados para a próxima geração, enquanto que os que estão no conjunto *Mutante* serão substituídos por novos cromossomos gerados aleatoriamente. Já os cromossomos no conjunto *Não-elite* são substituídos por novos cromossomos que serão obtidos através do cruzamento de dois cromossomos escolhidos aleatoriamente.

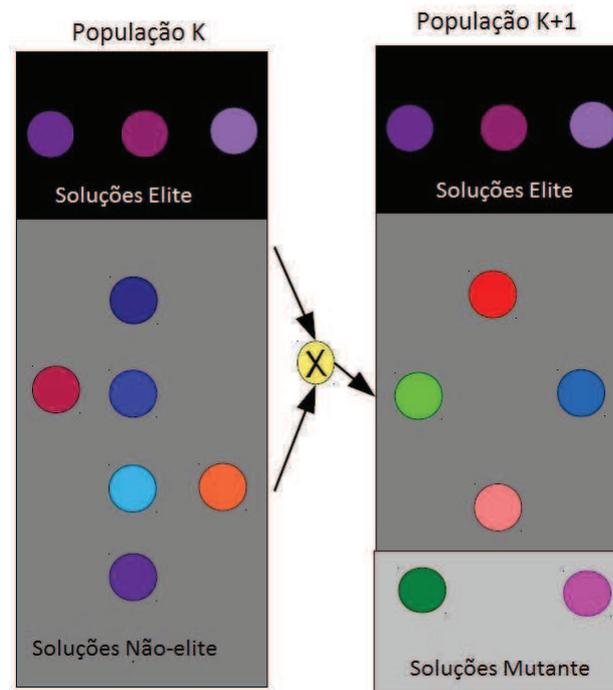


Figura 8 – Criação de uma nova população.

5.2.3 Algoritmos Genéticos com Chaves Aleatórias Viciadas

O algoritmo genético com chaves aleatórias viciadas (BRKGA - *Biased Random-Key Genetic Algorithm*) deriva do RKGA, onde representam uma solução do problema de otimização através de um vetor de chaves aleatórias (Gonçalves, J. F.; Resende, M. G. C., 2011). Uma chave aleatória é apenas um número real que é gerado aleatoriamente em um intervalo contínuo $[0, 1)$. Essas chaves aleatórias são decodificadas através de um decodificador, que no caso do decodificador de (Bean, J. C., 1994), basicamente ordena os elementos do vetor de chaves, gerando uma permutação que corresponde aos índices dos elementos ordenados.

Um RKGA evolui uma população de p vetores de chaves aplicando o princípio darwinista, onde os indivíduos mais fortes de uma população possuem mais chance de

encontrar um parceiro e conseqüentemente com isso, perpetuar seu material genético. Após gerada a k -ésima geração, os p vetores da população são particionados em um conjunto pequeno de $p = 2$ vetores que correspondem às melhores soluções (este conjunto se chama elite) e um outro conjunto com o restante da população (conjunto não-elite). Na hora de gerar a população, os vetores elite são copiados, sem mudança, para a população da $k + 1$ -ésima geração, assim também como os vetores mutantes, que possuem o mesmo papel de um algoritmo genético clássico, evitar que a população fique presa em um ótimo local.

No RKGA o darwinismo é utilizado apenas com o elitismo. Diferente do que ocorre no RKGA, o BRKGA se diferencia em etapas que os pais são selecionados para o cruzamento e como o cruzamento é implementado. Enquanto que no RKGA ambos os pais são escolhidos da população inteira, no BRKGA um pai sempre será escolhido do conjunto elite e o outro poderá ser escolhido do conjunto não-elite ou da população inteira, isso depende de como for parametrizado a escolha do segundo pai, representação do processo na figura 9.

Ambos os algoritmos fazem a combinação dos pais p_a e p_b com o cruzamento uniforme parametrizado para gerar o filho p_c . Entretanto, enquanto que no RKGA cada pai pode ser o p_a ou o p_b , no BRKGA o p_a sempre é o pai elite $1/p_e$, enquanto o p_b pode ou não ser o não-elite $1/(p - p_e)$ como pode ser visto na Figura x. Apesar de parecer uma pequena diferença entre o BRKGA e o RKGA, essas mudanças quase sempre fazem com que o BRKGA seja superior ao RKGA (Resende, M. G. C., 2012; Gonçalves, J. F.; Resende, M. G. C.; Toso, R. F., 2012).

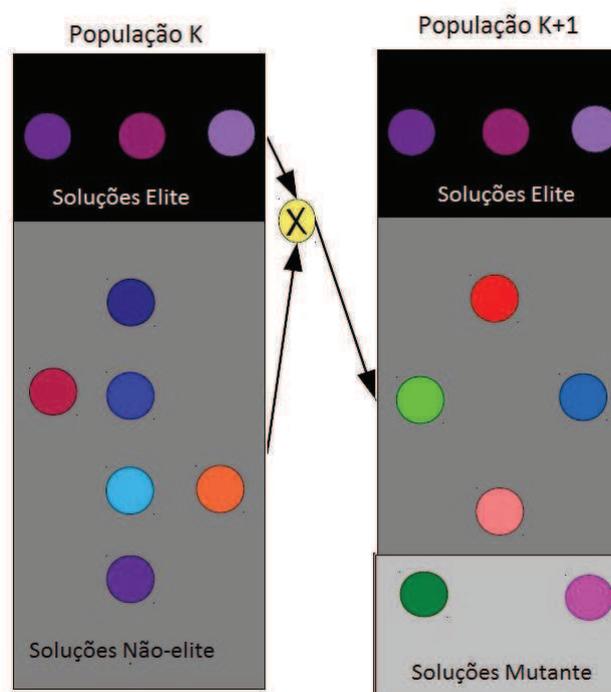


Figura 9 – Criação de uma nova população.

6 BRKGA E MULTI-START COM BUSCA LOCAL

O corrente capítulo é composto pela descrição dos métodos de resolução e como foram utilizados nesse trabalho de dissertação. Entre os métodos estão o multi-start (que faz uso da programação dinâmica) e o BRKGA (também utiliza programação dinâmica, mas como decodificador).

6.1 PROCEDIMENTO SPLIT

O *Split* consiste em uma técnica de decomposição do problema em subproblemas independentes com decisões multiestágio. Então, a ideia utilizada neste trabalho teve por base o princípio “*route first cluster second*” proposto por (Beasley, J. E., 1983) para o problema de roteamento de veículos (VRP - *Vehicle Routing Problem*). O procedimento *Split* consiste em gerar uma solução viável para o VRP a partir de uma solução para o caixeiro viajante (TSP - *Travelling Salesman Problem*). Considerando o grafo G , definido previamente, o TSP consiste em definir uma rota que sai de um vértice $i \in V$, passa por todos os nós $j \in V \setminus i$ uma única vez e retorna a i , de modo a otimizar o custo total do trajeto. Um algoritmo polinomial é proposto para definir os pontos de corte na solução do TSP de modo a torná-la viável para o VRP. O mais interessante é que o algoritmo é capaz de determinar os pontos de cortes ótimos de modo a obter um ótimo local para o VRP, considerando a ordem em que os nós aparecem na solução do TSP. Assim, neste trabalho, utiliza-se este princípio para determinar soluções para o PALAS, considerando a capacidade dos anéis e a relação dos anéis com o anel federal.

Seja o grafo auxiliar $H = (N, A)$ obtido a partir de uma sequência S dos vértices de V . Assim, N contém $n + 1$ pontos de corte de uma sequência S . O arco $(i, j) \in A$ corresponde a uma subsequência viável de S entre as posições i e j . O problema consiste em calcular um caminho válido induzido por H , do nó 0 até o nó n , de modo a minimizar a quantidade de arcos utilizados. Um caminho de 0 até n é válido se repete a restrição de capacidade dos anéis e do anel federal. O problema torna-se um problema de caminho mais curto com restrição de capacidade. O grafo H é acíclico. Assim, usar um algoritmo de programação dinâmica que calcula rótulos permite encontrar uma solução ótima, se existe uma, para a sequência inicial fornecida.

Quando um vértice é alocado a um anel no problema PALAS, referencia-se aqui como anel gerado ou aberto. Um rótulo l corresponde a um caminho parcial de 0 até um nó j . Sendo definido por $l = (p_l, n_l, t_l)$, onde p_l é o início do arco (p_l, j) que gerou l , n_l é a quantidade de anéis já abertos e t_l é o tráfego interno dos anéis abertos. Cada nó $i \in H$ tem uma lista L_i de rótulos. Inicialmente, $L_i = \emptyset$ para todos os nós $i > 0$. O nó 0

recebe o rótulo inicial $l_0 = (0,0,0)$. Os rótulos são propagados sob os arcos. Dado um rótulo $l = (p_l, n_l, t_l) \in L_i$ e um arco $(i, j) \in A$; a propagação ocorre da seguinte forma: um rótulo $l' = (i, n_l + 1, t_l + t(i, j))$ é gerado no nó j . O valor $t(i, j)$ corresponde ao tráfego da subsequência $S_i \dots S_{j-1}$ para os nós posteriores $S_j \dots S_n$. Se $t_l + t(i, j) > B$, a capacidade do anel federal é violada e o rótulo é descartado. O rótulo l' entra na lista L_j , se não tem um rótulo $l'' \in L_j$ que domina l' ($l'' < l'$). A condição para que um rótulo seja dominado é dada na Equação (13):

$$l_1 = (p_1, n_1, t_1) < l_2 = (p_2, n_2, t_2) \Leftrightarrow (n_1 = n_2) \text{ e } (t_1 < t_2) \quad (13)$$

As listas L_i são classificadas em ordem crescente de acordo com o valor n_l dos rótulos $l \in L_i$. Um pseudo-código para o algoritmo de cálculo de partições a partir de uma sequência fornecida é apresentado no Algoritmo (1).

Algorithm 1 Pseudo-código para determinar anéis para o SRAP.

Entrada: grafo $H = (N, A)$, sequência S

Saída: número mínimo de anéis associados a S

```

1: // Inicialização
2:  $L_i \leftarrow \{\}, \forall i = 1 \dots n$ 
3:  $L_0 \leftarrow \{(0,0,0)\}$ 
4: // Pontos de corte
5: for  $i = 0 \dots n - 1$  do
6:   if  $L_i \neq \emptyset$  then
7:     // Percorrer os arcos  $(i, j) \in A$ 
8:     for  $j = i + 1 \dots n$  do
9:       seja  $S' = S_{i+1} \dots S_j$  o novo anel
10:      calcular  $I(S')$  e  $E(S')$ , tráfegos interno e externo de  $S'$ 
11:      if  $I(S') + E(S') \leq Q$  then
12:        // Propagação de rótulos de  $L_i$ 
13:        for all  $l = (p_l, n_l, t_l) \in L_i$  do
14:          calcular  $E^+(S')$ , tráfego externo de  $S'$  com os nós  $S_{j+1} \dots S_n$ 
15:          if  $t_l + E^+(S') \leq Q$  then
16:             $l' = (i, n_l + 1, t_l + E^+(S'))$ 
17:            atualizar  $L_j$  com  $l'$ 
18:          end if
19:        end for
20:      end if
21:    end for
22:  end if
23: end for
24: if  $L_n = \emptyset$  then
25:   retornar  $n$ 
26: else
27:   retornar  $\arg \min_{l=(p_l, n_l, t_l) \in L_n} \{n_l\}$ 
28: end if

```

O PALAS tem uma vasta literatura e vários pesquisadores já trabalharam com ele. Trabalhos anteriores (Arighieri, R.; Dell'Amico, M, 2001; Bastos, L. D. O.; Ochi, L. S.; Macambira, E. M., 2005b; Macambira, E. M.; Maculan, N.; de Souza, C. C., 2006; Bernardino, E. M. et al., 2012) já ressaltaram a importância de permitir soluções inviáveis durante a busca local. Isto permite uma exploração mais eficiente do espaço de solução, ainda mais se tratando das restrições de capacidade que são muito limitadas. Além disso, alguns casos clássicos são conhecidos por serem inviáveis, enquanto outros, mesmo que viáveis, possuem restrições de capacidade muito limitadas. Com o procedimento do *Split*, quanto mais limitada for a restrição de capacidade, menor será a chance de encontrar uma solução viável a partir de uma sequência aleatória. O *Split* permite uma inviabilidade parcial, sendo apenas a restrição de capacidade do anel federal relaxada, mantendo a do anel local. Com isso as soluções calculadas pelo *Split* são viáveis em relação as restrições de capacidade dos anéis locais, podendo apenas violar as do anel federal $t_l > Q$. Este algoritmo é polinomial uma vez que cada arco de H é examinado uma única vez e a avaliação de $I(S')$, $E(S')$ e $E^+(S')$ tem complexidade assintótica de pior caso $O(m)$.

6.2 BUSCA LOCAL

Neste trabalho, a busca local trabalha com dois tipos de soluções: viáveis (respeita as restrições de capacidade dos anéis locais e federal) e as soluções inviáveis (a restrição de capacidade dos anéis locais é respeitada, mas a do anel federal não) (Goldschmidt, O.; Laugir, A.; Olinick, E. V., 2003).

A busca local aplicada neste trabalho utiliza os dois tipos de solução, para as viáveis, apenas a redução do número de anéis é visada, buscando o ótimo. Para as soluções inviáveis, a adequação a restrição de capacidade do anel federal é o objetivo, isto sem violar as dos anéis locais. Para isto, foram utilizadas duas estruturas de vizinhança, N_a e N_b .

6.2.1 Vizinhança N_a

Esta estrutura objetiva a minimização do número de anéis locais. Uma operação desta estrutura consiste em: retirar um cliente V do anel r em que está e alocá-lo a outro. Cada cliente é testado em todos os anéis, buscando a melhor configuração de anéis e esse teste é feito com todos os clientes, um exemplo pode ser visto na figura 10.

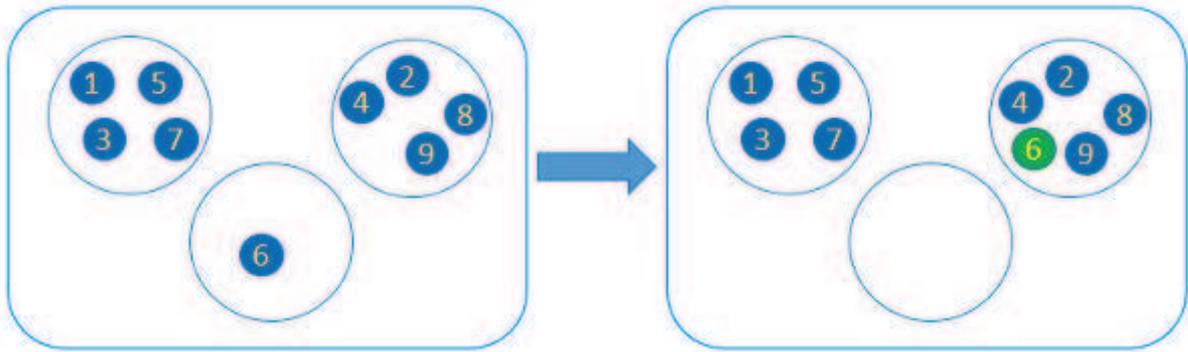


Figura 10 – Movimento realizado por Vizinhaça N_a .

6.2.2 Vizinhaça N_b

A estrutura de vizinhaça N_b busca a minimização do tráfego federal. Uma operação desta estrutura consiste em um *swap* (troca) de clientes, onde um cliente V de um anel r é trocado com outro cliente V de outro anel r . Cada cliente é testado no lugar de todos os outros clientes que estão em anéis distintos. Dentre as soluções inviáveis existem dois casos: já possuem o mesmo número de anéis de uma solução ótima, sendo caracterizada por inviável apenas pela violação de restrição do anel federal e outro caso é quando além de possuir a violação do anel federal, a solução possui um ou mais anéis que a solução ótima. Para o primeiro caso, apenas ajusta a solução e para o segundo caso, a solução atual pode ser melhorada no sentido de tráfego federal e facilitar a redução do número de anéis através da estrutura de vizinhaça N_a , a figura 11 ilustra o movimento.

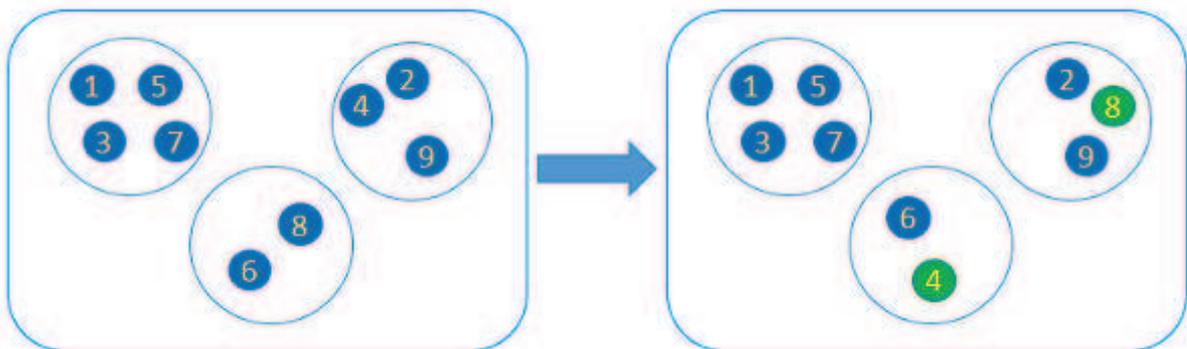


Figura 11 – Movimento realizado por Vizinhaça N_b .

6.3 MULTI-START E BRKGA

O multi-start é composto da técnica do *Split* e da busca local (*VND - Variable Neighborhood Descent*) (Mladenović, N.; Hansen, P., 1997). O multi-start opera n_m iterações, em cada iteração é gerada aleatoriamente uma solução inicial, com um vetor de tamanho n , sendo o número de clientes. Com a solução gerada, ela é avaliada pelo *Split* para determinar se ela é ou não viável, lembrando que existe uma relaxação na restrição de capacidade do anel federal, unicamente nele, mantendo os anéis locais intactos.

Avaliada a solução, a próxima etapa é o VND. São duas estruturas de vizinhança: N_1 que realiza apenas uma operação, a de envio de um cliente de um anel para outro, buscando assim a redução do número de anéis, já a N_2 realiza duas operações consecutivas, sendo responsável por tentar reaver a viabilidade da solução, pois ela realiza a troca entre 1 (um) par de clientes de anéis distintos, visando minimizar o tráfego do anel federal e possibilitar viabilidade quando possível. O pseudocódigo do Multi-Start para o PALAS pode ser visto no Algoritmo (2).

Algorithm 2 Pseudocódigo do Multi-Start para o PALAS.

Entrada: grafo $G = (V, E)$, capacidade Q

Saída: número mínimo de anéis associados a S

```

1: // Inicialização
2:  $S \leftarrow \emptyset$ 
3: // Iterações Multi-Start
4:  $L_i \leftarrow \{\}, \forall i = 0 \dots n_m$ 
5: for all  $i = 0 \dots n_m$  do
6:   // Solução inicial  $S_0$  usando Split na sequência aleatória
7:   // Aplicação do VND
8:   if  $S' < S_{temp}$  then
9:      $S_{temp} \leftarrow S'$ 
10:  end if
11:  // Atualiza
12:  if  $S_0 \leftarrow S_{temp}$  then
13:     $S_{temp} \leftarrow S'$ 
14:    if  $S_0 < S$  then
15:       $S \leftarrow S_0$ 
16:    end if
17:  end if
18: end for
19: retornar  $S$ 

```

No BRKGA se faz necessário um processo de decodagem. A solução inicial também é gerada aleatoriamente, onde o cromossomo (vetor de chaves aleatórias) possui dois valores, os das chaves aleatórias que é responsável pela ordenação de forma crescente do cromossomo e os índices que neste caso são os clientes que compõem a

solução. A função de decodagem é realizada pelo *Split*, que avalia a viabilidade da solução para que o BRKGA realize a otimização.

Um esquema geral dos dois procedimentos também podem ser vistos na figura 12.

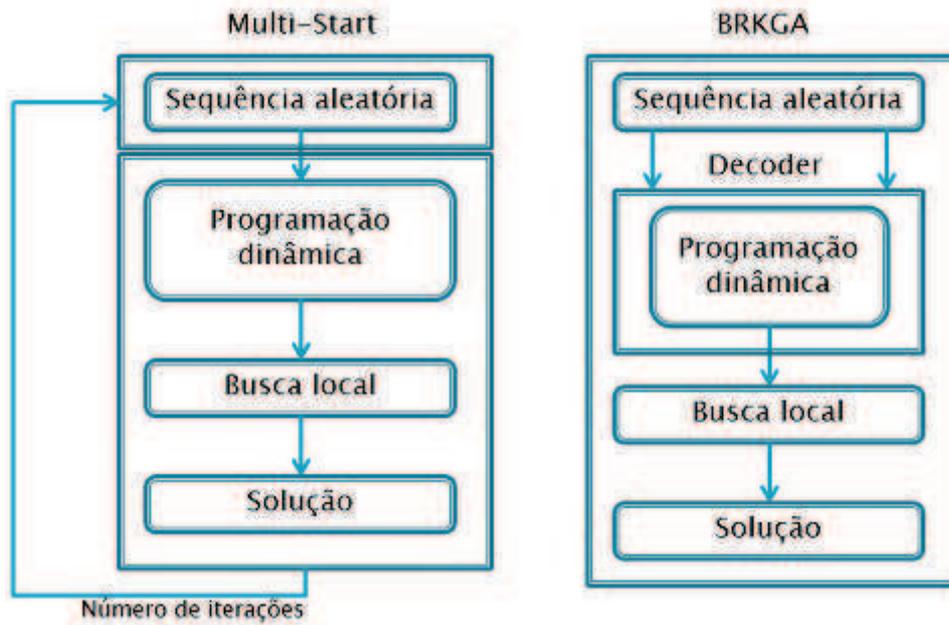


Figura 12 – Esquema dos métodos de resolução

7 RESULTADOS

Os algoritmos propostos neste trabalho foram desenvolvidos na linguagem de programação C++ e utilizando a IDE Visual Studio 2013. Os testes foram executados em uma máquina com o sistema operacional Windows 7 64 bits, com 8 Gb de memória RAM com frequência de 1333 Mhz e processador Intel Core i7 de 2.4 Ghz e com 4 núcleos.

7.1 RESULTADOS DAS METAHEURÍSTICAS

Para este trabalho foram utilizados 3 (três) conjuntos de instâncias C1, C2, C3.

As instâncias do grupo C1 são chamadas de GLO, pois foram propostas por (Goldschmidt, O. et al., 2003). Elas são divididas em dois grupos: 80 instâncias geométricas, mostrando uma maior demanda entre vizinhos e 80 instâncias aleatórias. Cada um dos grupos é subdividido em outros dois grupos de instâncias, sendo 40 instâncias com uma baixa capacidade do anel federal 155 Mbps e as outras 40 instâncias com uma maior capacidade de tráfego do anel federal, sendo 622 Mbps. O conjunto C1 tem tamanhos $|V|$ 15, 25, 30, 50. Das 160 instâncias, 42 foram provadas inviáveis por (Arighieri, R.; Dell'Amico, M, 2001) por meio do CPLEX 8.0.

Instâncias pertencentes ao grupo C2, também chamadas de AD, pois foram propostas por (Arighieri, R.; Dell'Amico, M, 2001) foram divididas em instâncias geométricas e randômicas e também possuem a subdivisão em relação capacidade do anel federal, também sendo de 155 Mbps e 622 Mbps. Totalizando 230 instâncias, onde 42 delas são instâncias modificadas aleatoriamente do conjunto C1, dadas por inviáveis. A quantidade de $|V|$ no grupo permanece a mesma.

O grupo de instâncias C3, conhecido por LSHK foi introduzido por (Lee, Y et al., 2000). Foram geradas 40 instâncias com capacidade de tráfego do anel federal limitado em 48 Mbps. O conjunto C3 possui $|V|$ 20, 25, 30 e $|E|$ de 30, 35. Esse conjunto tem apenas 2 instâncias conhecidas por viáveis para o problema do PALAS.

O método do Multi-Start teve o parâmetro de iterações n_m configurado para 1 (um) mil. Sendo geradas 1000 soluções aleatórias. O Multi-Start deixou a desejar em soluções iniciais, em casos de instâncias mais fáceis, ela conseguiu obter até soluções ótimas, porém em instâncias de maior complexidade ela não se mostrou eficaz. Devido ao relaxamento, pode-se permitir soluções não viáveis, o que ocorre em praticamente todos os casos em instâncias com maior número de $|V|$ ou com menor capacidade de

tráfego do anel federal, ficando a responsabilidade de tornar uma solução inviável em uma solução viável a cargo unicamente da busca local. Com o relaxamento as soluções encontradas sempre continham um ou mais anéis locais a mais que a solução ótima. Com a vizinhança N_1 que tentava minimizar o número de anéis respeitando a capacidade real do anel federal, foi obtido o êxito em encontrar soluções ótimas. Outro caso que ocorreu diversas vezes, foi o fato da solução não viável possuir o mesmo número de anéis da solução viável, nestes casos a vizinhança N_2 conseguiu reverter o caso e torna-lá uma solução viável, pois ela apenas troca clientes em anéis diferentes e testa se o tráfego está dentro do limite real.

De maneira geral, o Multi-Start se mostrou eficaz em relação a obter a solução ótima para os conjuntos de instâncias C1 e C2, em relação a C3 ele não foi tão eficaz assim, visto que a solução obtida possuía um anel a mais que a solução ótima. Se mostrou menos eficiente em relação a tempo, se comparado com a metaheurística mais bem sucedida até o momento, a *Honey Bee Mating Optimization* proposta por (Silva, A.C.G. et al., 2009).

O BRKGA teve os seguintes parâmetros: população $p = 100$, 50 gerações (iterações), população elite $p_e = 20$, população mutante $p_m = 10$, fator de herança de um pai elite $p_r = 0,7$. Dadas essas configurações, o BRKGA conseguiu obter soluções com qualidade igual ao Multi-Start, porém em questão de eficiência, obteve um menor desempenho no fator tempo, sempre demorando mais para encontrar uma solução ótima se comparado ao Multi-Start. Com o mesmo método de solução inicial, isso também prejudicou o BRKGA, pois com uma solução inicial aleatória e com possibilidade de ter péssima qualidade, tendencialmente necessitaria de mais gerações para obter uma solução ótima, caso não fosse capaz de ainda obter a solução ótima, o resto do processo caberia a busca local, encontrar o ótimo. Mais uma vez a relaxação se mostrou imprescindível para ter soluções com que se possa trabalhar mesmo que inviáveis.

Com exceção do conjunto C3, o Multi-Start e o BRKGA conseguiram obter soluções ótimas para todas as instâncias, sendo diferenciados pelo tempo de execução, onde o Multi-Start possui vantagem. Com relação ao conjunto C3, o BRKGA conseguiu obter a solução ótima para apenas uma das duas instâncias consideradas viáveis, igualmente ao Multi-Start, mas no caso onde não conseguiram alcançar o ótimo, o BRKGA apresentou um GAP menor.

Os resultados das metaheurísticas utilizadas neste trabalho mostram o tempo (segundos:milissegundos) e o *Gap* que mostrar a diferença do número de anéis, da solução ótima para a encontrada. As tabelas 2, 3 e 4 mostram os resultados obtidos respectivamente para os conjuntos de instâncias C1, C2 e C3.

Na tabela 5 segue uma comparação dos métodos utilizados neste trabalho com alguns existentes no estado da arte, como os trabalhos de (Bernardino, E. M. et al., 2012; Oliveira, W., 2010). A tabela possui a quantidade de soluções ótimas encontradas em

relação a cada conjunto de instâncias utilizado neste trabalho.

Tabela 2 – Resultados para as instâncias C1

Instância		Multi-Start			BRKGA		
Nome	Solução ótima	Solução encontrada	Tempo(s)	Gap	Solução encontrada	Tempo(s)	Gap
GL_15_1	3	3	0,001	0	3	0,02	0
GL_15_4	3	3	0,001	0	3	0,03	0
GL_15_7	3	3	0,003	0	3	0,02	0
GL_15_9	3	3	0,001	0	3	0,03	0
GL_25_1	4	4	0,002	0	4	0,23	0
GL_25_3	3	3	0,022	0	3	0,26	0
GL_25_4	4	4	0,004	0	4	0,22	0
GL_25_7	3	3	0,01	0	3	0,3	0
GL_25_8	4	4	0,012	0	4	0,19	0
GL_30_1	4	4	0,004	0	4	0,5	0
GL_30_2	4	4	0,006	0	4	0,45	0
GL_30_3	4	4	0,08	0	4	0,53	0
GL_30_4	4	4	0,005	0	4	0,58	0
GL_30_5	3	3	0,005	0	3	0,62	0
GL_30_6	4	4	0,005	0	4	0,52	0
GL_30_7	4	4	0,006	0	4	0,51	0
GL_30_8	4	4	0,006	0	4	0,48	0
GL_30_9	4	4	0,005	0	4	0,54	0
GL_50_1	5	5	0,614	0	5	2,93	0
GL_50_4	6	6	0,07	0	6	2,91	0
GL_50_7	5	5	1,282	0	5	3,22	0
GL_50_8	5	5	0,429	0	5	3,08	0
GL_50_9	4	4	1,011	0	4	4,49	0
GH_15_1	3	3	0,001	0	3	0,04	0
GH_15_2	3	3	0,001	0	3	0,04	0
GH_15_3	2	2	0,031	0	2	0,04	0
GH_15_6	2	2	0,037	0	2	0,05	0
GH_15_7	2	2	0,001	0	2	0,06	0
GH_15_8	3	3	0,001	0	3	0,03	0
GH_15_9	3	3	0,002	0	3	0,04	0
GH_15_10	2	2	0,079	0	2	0,05	0
GH_25_1	3	3	0,008	0	3	0,41	0
GH_25_2	2	2	0,144	0	2	0,45	0
GH_25_3	2	2	0,21	0	2	0,44	0
GH_25_4	3	3	0,005	0	3	0,28	0
GH_25_7	4	4	0,001	0	4	0,24	0
GH_25_8	3	3	0,008	0	3	0,37	0
GH_25_9	3	3	0,001	0	3	0,34	0
GH_30_1	3	3	0,002	0	3	0,82	0
GH_30_2	4	4	0,002	0	4	0,59	0
GH_30_3	4	4	0,002	0	4	0,59	0
GH_30_4	3	3	0,002	0	3	0,69	0
GH_30_5	4	4	0,003	0	4	0,62	0

Instância		Multi-Start			BRKGA		
Nome	Solução ótima	Solução encontrada	Tempo(s)	Gap	Solução encontrada	Tempo(s)	Gap
GH_30_9	4	4	0,004	0	4	0,66	0
GH_30_10	3	3	0,001	0	3	0,9	0
GH_50_1	5	5	0,024	0	5	3,31	0
GH_50_4	4	4	1,011	0	4	3,86	0
GH_50_5	5	5	0,022	0	5	4,01	0
GH_50_6	5	5	0,011	0	5	3,28	0
GH_50_7	5	5	0,017	0	5	3,39	0
RL_15_1	3	3	0,001	0	3	0,05	0
RL_15_3	2	2	0,097	0	2	0,06	0
RL_15_4	3	3	0,013	0	3	0,04	0
RL_15_6	3	3	0,001	0	3	0,05	0
RL_15_8	3	3	0,001	0	3	0,04	0
RL_15_9	3	3	0,001	0	3	0,04	0
RL_15_10	3	3	0,008	0	3	0,04	0
RL_25_1	4	4	0,019	0	4	0,33	0
RL_25_2	3	3	0,007	0	3	0,4	0
RL_25_4	4	4	0,002	0	4	0,33	0
RL_25_5	4	4	0,002	0	4	0,37	0
RL_25_6	4	4	0,002	0	4	0,34	0
RL_25_7	4	4	0,002	0	4	0,33	0
RL_25_8	3	3	0,003	0	3	0,44	0
RL_25_9	4	4	0,001	0	4	0,29	0
RL_25_10	4	4	0,001	0	4	0,56	0
RL_30_1	3	3	0,032	0	3	0,84	0
RL_30_3	4	4	0,013	0	4	0,48	0
RL_30_4	4	4	0,01	0	4	0,46	0
RL_30_5	4	4	0,001	0	4	0,61	0
RL_30_6	4	4	0,026	0	4	0,48	0
RL_30_7	3	3	0,502	0	3	0,67	0
RL_30_8	4	4	0,004	0	4	0,64	0
RL_30_10	4	4	0,092	0	4	0,56	0
RL_50_1	5	5	0,021	0	5	4,35	0
RL_50_3	4	4	1,401	0	4	5,46	0
RL_50_4	4	4	1,111	0	4	4,57	0
RL_50_5	4	4	0,374	0	4	4,53	0
RL_50_6	4	4	0,047	0	4	4,73	0
RL_50_7	5	5	0,01	0	5	4,96	0
RL_50_10	4	4	0,118	0	4	4,08	0
RH_15_1	3	3	0,001	0	3	0,03	0
RH_15_3	3	3	0,001	0	3	0,03	0
RH_15_4	2	2	0,098	0	2	0,05	0
RH_15_5	3	3	0,001	0	3	0,04	0
RH_15_6	3	3	0,001	0	3	0,04	0
RH_15_7	2	2	0,089	0	2	0,04	0

Instância		Multi-Start			BRKGA		
Nome	Solução ótima	Solução encontrada	Tempo(s)	Gap	Solução encontrada	Tempo(s)	Gap
RH_15_8	2	2	0,099	0	2	0,04	0
RH_15_9	3	3	0,001	0	3	0,04	0
RH_25_1	3	3	0,001	0	3	0,3	0
RH_25_2	3	3	0,001	0	3	0,42	0
RH_25_3	3	3	0,001	0	3	0,37	0
RH_25_4	3	3	0,018	0	3	0,27	0
RH_25_5	4	4	0,019	0	4	0,25	0
RH_25_6	3	3	0,001	0	3	0,29	0
RH_25_7	3	3	0,001	0	3	0,4	0
RH_25_8	4	4	0,001	0	4	0,26	0
RH_25_9	2	2	0,441	0	2	0,55	0
RH_25_10	3	3	0,361	0	3	0,42	0
RH_30_1	3	3	0,009	0	3	1,07	0
RH_30_2	4	4	0,001	0	4	0,6	0
RH_30_3	3	3	0,008	0	3	0,99	0
RH_30_4	3	3	0,002	0	3	0,84	0
RH_30_6	4	4	0,002	0	4	0,59	0
RH_30_7	4	4	0,001	0	4	0,62	0
RH_30_8	3	3	0,044	0	3	0,75	0
RH_30_9	3	3	0,009	0	3	0,71	0
RH_30_10	3	3	0,688	0	3	0,63	0
RH_50_1	4	4	0,02	0	4	4,97	0
RH_50_2	3	3	3,182	0	3	7,69	0
RH_50_3	4	4	0,092	0	4	4,63	0
RH_50_4	4	4	0,189	0	4	4,79	0
RH_50_5	3	3	2,615	0	3	11,67	0
RH_50_6	3	3	2,878	0	3	13	0
RH_50_7	5	5	0,044	0	5	8,01	0
RH_50_8	4	4	0,087	0	4	10,25	0
RH_50_9	4	4	0,154	0	4	9,48	0
RH_50_10	4	4	0,191	0	4	4,54	0
			0,174			1,517	

Tabela 3 – Resultados para as instâncias C2

Instância		Multi-Start			BRKGA		
Nome	Solução ótima	Solução encontrada	Tempo(s)	Gap	Solução encontrada	Tempo(s)	Gap
GL_15_3.1	4	4	0,001	0	4	0,01	0
GL_15_3.2	4	4	0,001	0	4	0,01	0
GL_15_3.3	4	4	0,001	0	4	0,01	0
GL_15_3.4	4	4	0,001	0	4	0,01	0
GL_15_3.5	4	4	0,001	0	4	0,01	0
GL_15_3.6	4	4	0,001	0	4	0,01	0
GL_15_3.7	4	4	0,001	0	4	0,01	0
GL_15_3.8	4	4	0,002	0	4	0,01	0
GL_15_3.9	4	4	0,001	0	4	0,01	0
GL_15_3.10	4	4	0,001	0	4	0,01	0
GL_15_6.1	3	3	0,001	0	3	0,01	0
GL_15_6.2	3	3	0,001	0	3	0,01	0
GL_15_6.3	3	3	0,001	0	3	0,01	0
GL_15_6.4	3	3	0,001	0	3	0,01	0
GL_15_6.5	3	3	0,001	0	3	0,01	0
GL_15_6.6	3	3	0,001	0	3	0,01	0
GL_15_6.7	3	3	0,001	0	3	0,01	0
GL_15_6.8	3	3	0,001	0	3	0,01	0
GL_15_6.9	3	3	0,001	0	3	0,01	0
GL_15_6.10	3	3	0,001	0	3	0,01	0
GL_25_9.1	4	4	0,002	0	4	0,11	0
GL_25_9.2	4	4	0,002	0	4	0,12	0
GL_25_9.3	4	4	0,002	0	4	0,1	0
GL_25_9.4	4	4	0,002	0	4	0,12	0
GL_25_9.5	4	4	0,001	0	4	0,11	0
GL_25_9.6	4	4	0,001	0	4	0,13	0
GL_25_9.7	4	4	0,001	0	4	0,1	0
GL_25_9.8	4	4	0,001	0	4	0,11	0
GL_25_9.9	4	4	0,001	0	4	0,1	0
GL_25_9.10	4	4	0,002	0	4	0,11	0
GL_25_10.1	5	5	0,001	0	5	0,07	0
GL_25_10.2	5	5	0,001	0	5	0,07	0
GL_25_10.3	5	5	0,001	0	5	0,08	0
GL_25_10.4	5	5	0,001	0	5	0,07	0
GL_25_10.5	5	5	0,001	0	5	0,08	0
GL_25_10.6	5	5	0,001	0	5	0,08	0
GL_25_10.7	5	5	0,001	0	5	0,08	0
GL_25_10.8	5	5	0,001	0	5	0,08	0
GL_25_10.9	5	5	0,002	0	5	0,08	0
GL_25_10.10	5	5	0,001	0	5	0,07	0
GL_30_10.1	5	5	0,001	0	5	0,16	0
GL_30_10.2	5	5	0,001	0	5	0,18	0
GL_30_10.3	5	5	0,001	0	5	0,16	0

Instância		Multi-Start			BRKGA		
Nome	Solução ótima	Solução encontrada	Tempo(s)	Gap	Solução encontrada	Tempo(s)	Gap
GL_30_10.4	5	5	0,007	0	5	0,17	0
GL_30_10.5	5	5	0,001	0	5	0,16	0
GL_30_10.6	5	5	0,002	0	5	0,16	0
GL_30_10.7	5	5	0,002	0	5	0,16	0
GL_30_10.8	5	5	0,001	0	5	0,17	0
GL_30_10.9	5	5	0,001	0	5	0,16	0
GL_30_10.10	5	5	0,003	0	5	0,16	0
GL_50_6.1	6	6	0,004	0	6	4,82	0
GL_50_6.2	6	6	0,017	0	6	4,77	0
GL_50_6.3	6	6	0,015	0	6	4,13	0
GL_50_6.4	6	6	0,077	0	6	4,2	0
GL_50_6.5	6	6	0,011	0	6	4,21	0
GL_50_6.6	6	6	0,007	0	6	4,74	0
GL_50_6.7	6	6	0,066	0	6	4,4	0
GL_50_6.8	6	6	0,004	0	6	4,42	0
GL_50_6.9	6	6	0,003	0	6	3,97	0
GL_50_6.10	6	6	0,004	0	6	4,42	0
GL_50_10.1	5	5	0,01	0	5	6	0
GL_50_10.2	5	5	0,471	0	5	5,35	0
GL_50_10.3	6	6	0,004	0	6	5,61	0
GL_50_10.4	5	5	0,397	0	5	5,25	0
GL_50_10.5	5	5	0,299	0	5	5,81	0
GL_50_10.6	5	5	0,304	0	5	5,07	0
GL_50_10.7	6	6	0,007	0	6	5,57	0
GL_50_10.8	5	5	0,44	0	5	4,02	0
GL_50_10.9	6	6	0,004	0	6	4,29	0
GL_50_10.10	6	6	0,006	0	6	5,41	0
GH_25_5.1	4	4	0,006	0	4	0,13	0
GH_25_5.2	4	4	0,001	0	4	0,14	0
GH_25_5.3	4	4	0,001	0	4	0,15	0
GH_25_5.4	4	4	0,007	0	4	0,13	0
GH_25_5.5	4	4	0,008	0	4	0,15	0
GH_25_5.6	4	4	0,001	0	4	0,13	0
GH_25_5.7	4	4	0,022	0	4	0,12	0
GH_25_5.8	4	4	0,007	0	4	0,14	0
GH_25_5.9	4	4	0,047	0	4	0,14	0
GH_25_5.10	4	4	0,009	0	4	0,14	0
GH_30_8.1	4	4	0,023	0	4	0,3	0
GH_30_8.2	4	4	0,003	0	4	0,28	0
GH_30_8.3	4	4	0,021	0	4	0,36	0
GH_30_8.4	4	4	0,008	0	4	0,31	0

Instância		Multi-Start			BRKGA		
Nome	Solução ótima	Solução encontrada	Tempo(s)	Gap	Solução encontrada	Tempo(s)	Gap
GH_30_8.5	4	4	0,01	0	4	0,36	0
GH_30_8.6	4	4	0,011	0	4	0,3	0
GH_30_8.7	4	4	0,017	0	4	0,33	0
GH_30_8.8	4	4	0,003	0	4	0,36	0
GH_30_8.9	4	4	0,007	0	4	0,3	0
GH_30_8.10	4	4	0,002	0	4	0,3	0
GH_50_2.1	6	6	0,117	0	6	1,16	0
GH_50_2.2	6	6	0,07	0	6	1,19	0
GH_50_2.3	6	6	0,011	0	6	1,15	0
GH_50_2.4	6	6	0,065	0	6	1,25	0
GH_50_2.5	6	6	0,078	0	6	1,14	0
GH_50_2.6	6	6	0,066	0	6	1,18	0
GH_50_2.7	6	6	0,09	0	6	1,27	0
GH_50_2.8	6	6	0,021	0	6	1,15	0
GH_50_2.9	6	6	0,033	0	6	1,14	0
GH_50_2.10	6	6	0,017	0	6	1,15	0
GH_50_3.1	5	5	0,044	0	5	1,51	0
GH_50_3.2	5	5	0,039	0	5	1,49	0
GH_50_3.3	5	5	0,091	0	5	1,7	0
GH_50_3.4	5	5	2,218	0	5	1,59	0
GH_50_3.5	5	5	1,873	0	5	1,53	0
GH_50_3.6	5	5	1,769	0	5	1,5	0
GH_50_3.7	5	5	0,151	0	5	1,52	0
GH_50_3.8	5	5	0,028	0	5	1,66	0
GH_50_3.9	5	5	0,027	0	5	1,62	0
GH_50_3.10	6	6	0,047	0	6	1,54	0
GH_50_8.1	6	6	0,363	0	6	1,28	0
GH_50_8.2	6	6	0,032	0	6	1,32	0
GH_50_8.3	6	6	0,097	0	6	1,29	0
GH_50_8.4	6	6	0,021	0	6	1,27	0
GH_50_8.5	6	6	0,078	0	6	1,33	0
GH_50_8.6	6	6	0,154	0	6	1,27	0
GH_50_8.7	6	6	0,101	0	6	1,33	0
GH_50_8.8	6	6	0,093	0	6	1,26	0
GH_50_8.9	6	6	0,331	0	6	1,3	0
GH_50_8.10	6	6	0,943	0	6	1,33	0
GH_50_9.1	5	5	0,062	0	5	1,59	0
GH_50_9.2	5	5	0,057	0	5	1,62	0
GH_50_9.3	5	5	0,288	0	5	1,54	0
GH_50_9.4	5	5	1,915	0	5	1,53	0
GH_50_9.5	5	5	0,047	0	5	1,55	0
GH_50_9.6	5	5	0,38	0	5	1,84	0
GH_50_9.7	5	5	2,1	0	5	1,69	0
GH_50_9.8	5	5	0,078	0	5	1,67	0
GH_50_9.9	5	5	0,018	0	5	1,81	0
GH_50_9.10	5	5	0,07	0	5	1,63	0
GH_50_10.1	5	5	0,117	0	5	1,76	0

Instância		Multi-Start			BRKGA		
Nome	Solução ótima	Solução encontrada	Tempo(s)	Gap	Solução encontrada	Tempo(s)	Gap
GH_50_10.2	5	5	0,031	0	5	1,58	0
GH_50_10.3	5	5	0,077	0	5	1,58	0
GH_50_10.4	5	5	0,022	0	5	1,53	0
GH_50_10.5	5	5	1,749	0	5	1,58	0
GH_50_10.6	5	5	0,064	0	5	1,6	0
GH_50_10.7	5	5	1,989	0	5	1,57	0
GH_50_10.8	5	5	2,187	0	5	1,54	0
GH_50_10.9	5	5	0,099	0	5	1,5	0
GH_50_10.10	5	5	0,113	0	5	1,5	0
RL_15_2.1	3	3	0,001	0	3	0,03	0
RL_15_2.2	4	4	0,001	0	4	0,02	0
RL_15_2.3	3	3	0,001	0	3	0,03	0
RL_15_2.4	3	3	0,001	0	3	0,03	0
RL_15_2.5	3	3	0,001	0	3	0,03	0
RL_15_2.6	3	3	0,001	0	3	0,03	0
RL_15_2.7	3	3	0,001	0	3	0,03	0
RL_15_2.8	3	3	0,001	0	3	0,03	0
RL_15_2.9	4	4	0,002	0	4	0,03	0
RL_15_2.10	3	3	0,002	0	3	0,02	0
RL_15_5.1	3	3	0,001	0	3	0,03	0
RL_15_5.2	3	3	0,001	0	3	0,03	0
RL_15_5.3	3	3	0,001	0	3	0,03	0
RL_15_5.4	3	3	0,002	0	3	0,03	0
RL_15_5.5	3	3	0,001	0	3	0,04	0
RL_15_5.6	3	3	0,002	0	3	0,04	0
RL_15_5.7	3	3	0,001	0	3	0,05	0
RL_15_5.8	3	3	0,002	0	3	0,05	0
RL_15_5.9	3	3	0,001	0	3	0,05	0
RL_15_5.10	3	3	0,002	0	3	0,03	0
RL_25_3.1	4	4	0,002	0	4	0,37	0
RL_25_3.2	4	4	0,007	0	4	0,37	0
RL_25_3.3	4	4	0,01	0	4	0,36	0
RL_25_3.4	4	4	0,021	0	4	0,38	0
RL_25_3.5	4	4	0,006	0	4	0,36	0
RL_25_3.6	4	4	0,002	0	4	0,38	0
RL_25_3.7	4	4	0,002	0	4	0,37	0
RL_25_3.8	4	4	0,005	0	4	0,4	0
RL_25_3.9	4	4	0,004	0	4	0,39	0
RL_25_3.10	4	4	0,008	0	4	0,38	0

Instância		Multi-Start			BRKGA		
Nome	Solução ótima	Solução encontrada	Tempo(s)	Gap	Solução encontrada	Tempo(s)	Gap
RL_30_2.1	4	4	0,009	0	4	0,81	0
RL_30_2.2	4	4	0,021	0	4	0,73	0
RL_30_2.3	4	4	0,098	0	4	0,77	0
RL_30_2.4	4	4	0,055	0	4	0,75	0
RL_30_2.5	4	4	0,023	0	4	0,71	0
RL_30_2.6	4	4	0,033	0	4	0,78	0
RL_30_2.7	4	4	0,071	0	4	0,73	0
RL_30_2.8	4	4	0,097	0	4	0,79	0
RL_30_2.9	4	4	0,008	0	4	0,77	0
RL_30_2.10	4	4	0,02	0	4	0,79	0
RL_30_9.1	4	4	0,053	0	4	0,84	0
RL_30_9.2	4	4	0,01	0	4	0,77	0
RL_30_9.3	4	4	0,002	0	4	0,73	0
RL_30_9.4	4	4	0,099	0	4	0,83	0
RL_30_9.5	4	4	0,011	0	4	0,85	0
RL_30_9.6	4	4	0,288	0	4	0,77	0
RL_30_9.7	4	4	0,072	0	4	0,74	0
RL_30_9.8	4	4	0,022	0	4	0,8	0
RL_30_9.9	4	4	0,028	0	4	0,86	0
RL_30_9.10	4	4	0,154	0	4	0,73	0
RL_50_2.1	5	5	0,096	0	5	5,77	0
RL_50_2.2	5	5	0,07	0	5	5,68	0
RL_50_2.3	5	5	1,958	0	5	5,79	0
RL_50_2.4	5	5	2,094	0	5	4,85	0
RL_50_2.5	5	5	0,367	0	5	4,53	0
RL_50_2.6	5	5	0,069	0	5	3,61	0
RL_50_2.7	5	5	0,121	0	5	3,66	0
RL_50_2.8	5	5	0,309	0	5	3,49	0
RL_50_2.9	5	5	3,1	0	5	3,76	0
RL_50_2.10	5	5	0,069	0	5	5,97	0
RL_50_8.1	5	5	0,155	0	5	5,28	0
RL_50_8.2	5	5	0,671	0	5	5,6	0
RL_50_8.3	5	5	0,998	0	5	5,47	0
RL_50_8.4	5	5	0,7	0	5	5,16	0
RL_50_8.5	6	6	0,502	0	6	5,99	0
RL_50_8.6	5	5	0,41	0	5	5,6	0
RL_50_8.7	5	5	0,601	0	5	5,47	0
RL_50_8.8	5	5	2,144	0	5	4,28	0
RL_50_8.9	5	5	1,413	0	5	4,07	0
RL_50_8.10	5	5	0,1	0	5	5,47	0
RH_15_10.1	3	3	0,001	0	3	0,03	0
RH_15_10.2	3	3	0,001	0	3	0,03	0
RH_15_10.3	3	3	0,001	0	3	0,03	0

Instância		Multi-Start			BRKGA		
Nome	Solução ótima	Solução encontrada	Tempo(s)	Gap	Solução encontrada	Tempo(s)	Gap
RH_15_10.4	3	3	0,002	0	3	0,03	0
RH_15_10.5	3	3	0,071	0	3	0,03	0
RH_15_10.6	3	3	0,001	0	3	0,03	0
RH_15_10.7	3	3	0,001	0	3	0,03	0
RH_15_10.8	3	3	0,001	0	3	0,03	0
RH_15_10.9	3	3	0,001	0	3	0,03	0
RH_15_10.10	3	3	0,001	0	3	0,03	0
RH_30_5.1	4	4	0,001	0	4	0,65	0
RH_30_5.2	4	4	0,001	0	4	0,63	0
RH_30_5.3	4	4	0,001	0	4	0,63	0
RH_30_5.4	4	4	0,001	0	4	0,58	0
RH_30_5.5	4	4	0,003	0	4	0,63	0
RH_30_5.6	4	4	0,002	0	4	0,63	0
RH_30_5.7	4	4	0,001	0	4	0,62	0
RH_30_5.8	4	4	0,002	0	4	0,64	0
RH_30_5.9	4	4	0,004	0	4	0,61	0
RH_30_5.10	4	4	0,003	0	4	0,57	0
			0,175	0		1,319	0

Tabela 4 – Resultados para as instâncias C3

Instância		Multi-Start			BRKGA		
Nome	Solução ótima	Solução encontrada	Tempo(s)	Gap	Solução encontrada	Tempo(s)	Gap
LSHL_25_35_02	6	8	0,021	2	7	0,101	1
LSHL_25_35_08	6	6	0,015	0	6	0,128	0
			0,018	0		0,165	0

Tabela 5 – Resultados do estado da arte para os conjuntos de instâncias C1, C2 e C3.

		Multi-Start	BRKGA	HBMO	MA+VB
Conjunto de instâncias	Instâncias	Instâncias viáveis	Soluções ótimas	Soluções ótimas	Soluções ótimas
C1	160	118	100%	100%	100%
C2	230	230	100%	100%	99,57%
C3	40	2	50%	50%	-

8 CONSIDERAÇÕES FINAIS

O trabalho propôs uma nova técnica para o problema do PALAS, a qual se mostrou promissora ao ser combinada com metaheurísticas, conseguindo atingir soluções de boa qualidade e em tempo computacional, podendo ser expandido para outros problemas de partição de grafos. Com as instâncias trabalhadas ele conseguiu achar o ótimo em todas, com exceção do conjunto C3. O método proposto aqui pode ser aplicado em combinação com diversas heurísticas e metaheurísticas.

O trabalho ainda pode ser muito melhorado, principalmente com a aplicação de algum método heurístico para a construção da solução inicial, pois partindo-se de boas soluções iniciais, tendencialmente exigirá menos trabalho computacional por parte das metaheurísticas, aproximando também de uma possível solução ótima para o problema.

Para trabalhos futuros fica a proposta de um bom método para construção de uma solução inicial ou trabalhar em cima de estruturas de vizinhança para conseguir resolver de forma ótima as outras instâncias não trabalhadas.

A construção de uma solução inicial pode ser embasada por meio da construção de bons anéis. Isto pode ser feito seguindo características dos anéis de soluções ótimas, pois eles possuem um padrão, ainda mais em situações onde as restrições de capacidade são bem limitadas. Um método heurístico que pode ser aplicado para uma solução inicial do PALAS, é o *binpacking*, associando um anel a um *bin*.

REFERÊNCIAS

- Afsar, H. M.; Prins, C.; Santos, A. C. Exact and heuristic algorithms for solving the generalized vehicle routing problem with flexible fleet size. *International Transactions in Operational Research*, v. 21, p. 153–175, 2014.
- Arighieri, R.; Dell’Amico, M. *Solution of the sonet ring assignment problem with capacity constraints*. [S.l.], 2001.
- Arighieri, R.; Dell’Amico, M. *Comparing metaheuristic algorithms for SONET network design problems*. [S.l.], 2005.
- Ausiello, G. et al. *Complexity and Approximation Combinatorial Optimization Problems and Their Approximability Properties*. [S.l.]: Springer, 2003.
- Bastos, L. D. O.; Ochi, L. S. A genetic algorithm with evolutionary path-relinking for the Sonet Ring Assignment Problem. In: *International Conference on Engineering Optimization - EngOpt2008*. [S.l.: s.n.], 2008.
- Bastos, L. D. O.; Ochi, L. S.; Macambira, E. M. A relative neighbourhood GRASP for the SONET Ring Assignment Problem. In: *Network Optimization Conference*. [S.l.: s.n.], 2005.
- Bastos, L. D. O.; Ochi, L. S.; Macambira, E. M. GRASP with path-relinking for the SONET ring assignment problem. In: *5th International Conference on Hybrid Intelligent Systems (HIS2005), co-sponsored by IEEE Systems, Man, and Cybernetics Society*. [S.l.: s.n.], 2005.
- Bastos, L. O. *Soluções Heurísticas para o Problema de Atribuição de Localidades a Anéis em Redes SONET*. Dissertação (Mestrado) — Universidade Federal Fluminense, 2005.
- Bean, J. C. Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, v. 6, p. 154–160, 1994.
- Beasley, J. E. Route-first cluster-second methods for vehicle routing. *Omega*, v. 11, p. 403–408, 1983.
- Bernardino, E. M. et al. Solving SONET problems using a hybrid scatter search algorithm. In: *Computational Intelligence, IJCCI 2010*. [S.l.]: Springer, 2010.
- Bernardino, E. M. et al. Solving large-scale SONET network design problems using bee-inspired algorithms. *Optical Switching and Networking*, v. 9, p. 97–117, 2012.
- Cormen, T. H. et al. *Introduction to Algorithms*. 3^o. ed. [S.l.]: The MIT Press, 2009.
- Darwin, C. *The Origin of Species*. [S.l.]: Bantam Classics, 1999.
- Davis, L. D. *Handbook of genetic algorithms*. [S.l.]: Van Nostrand Reinhold, 1991.
- Fortz, B; Soriano, P.; Wynants, C. A tabu search algorithm for self-healing ring network design. *European Journal of Operational Research*, v. 151, p. 280–295, 2003.
- Geyer-Schulz, A. *Fuzzy Rule-Based Expert Systems and Genetic Machine Learning (Studies in Fuzziness and Soft Computing)*. [S.l.]: Physica-Verlag HD, 1996.

- Girão, A. C. S. *Busca Heurística Através de Algoritmo Genético e Memético com Construção de Vocábulos para o Problema de Atribuição de Localidades a Anéis SONET*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2008.
- Goldberg, D. E. *Genetic algorithms in search, optimization & machine learning*. [S.l.]: Addison-Wesley, 1989.
- Goldschmidt, O. et al. The SONET edge-partition problem. *Networks*, v. 41, p. 3–23, 2003.
- Goldschmidt, O.; Laugier, A.; Olinick, E. V. *SONET/SDH ring assignment with capacity constraints*. [S.l.], 2003.
- Gonçalves, J. F.; Resende, M. G. C. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, v. 17, p. 487–525, 2011.
- Gonçalves, J. F.; Resende, M. G. C.; Toso, R. F. *Biased and unbiased random key genetic algorithms: An experimental analysis*. [S.l.], 2012.
- Holland, J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. [S.l.]: A Bradford Book, 1992.
- Lee, Y et al. A branch-and-cut algorithm for solving an intraring synchronous optical network design problem. *Networks*, v. 35, p. 223–232, 2000.
- Levitin, A. *Introduction to the Design & Analysis of Algorithms*. 3^a. ed. [S.l.]: Addison-Wesley, 2011.
- Macambira, E. M.; Maculan, N.; de Souza, C. C. A column generation approach for SONET ring assignment. *Networks*, v. 47, p. 157–171, 2006.
- Mladenović, N.; Hansen, P. Variable neighborhood search. *Computers & Operations Research*, v. 24, p. 1097–1100, 1997.
- Oliveira, W. *Algoritmo Evolutivo Paralelo para o Problema de Atribuição de Localidades A Anéis em Redes SONET/SDH*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2010.
- Pelleau, M.; Van Hentenryck, P.; Truchet, C. SONET network design problems. In: *Sixth International Workshop on Local Search Techniques in Constraint Satisfaction*. [S.l.: s.n.], 2009. p. 81–95.
- Perros, H. G. *Connection-oriented Networks SONET/SDH, ATM, MPLS and OPTICAL NETWORKS*. 1. ed. [S.l.]: Willey, 2005.
- Pioro, M.; Medhi, D. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. [S.l.]: Morgan Kaufmann, 2004.
- Prins, C.; Lacomme, P.; Prodhon, C. Order-first split-second methods for vehicle routing problems: A review. *Transportation Research Part C: Emerging Technologies*, v. 40, p. 179–200, 2014.
- Resende, M. G. C. Biased random-key genetic algorithms with applications in telecommunications. *TOP*, v. 20, p. 130–153, 2012.

Silva, A.C.G. et al. Genetic algorithm and memetic algorithm with vocabulary building for the SONET ring assignment problem. In: *EU/MEeting 2009 - European Chapter on Metaheuristics*. [S.l.: s.n.], 2009.

Skiena, S. S.; Revilla, M. A. *Programming Challenges: The Programming Contest Training Manual*. [S.l.]: Springer, 2003.

Soares, W. *Heurísticas usando construção de vocabulário aplicadas ao problema de atribuição de localidades a anéis em Redes SONET/SDH*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2008.

Soriano, P. et al. Design and Dimensioning of Survivable SDH/SONET Networks. In: *Telecommunications Network Planning*. [S.l.: s.n.], 1998.

Vasseur, J. P.; Pickavet, M.; Demeester, P. *Network Recovery Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. 1. ed. [S.l.]: Morgan Kaufmann, 2004.

Wall, M. G. *C++ library of genetic algorithm components*. <http://lancet.mit.edu/ga/>, 2000.

Whitley, D. A genetic algorithm tutorial. *Statistics and Computing*, v. 4, p. 65–85, 1994.