

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE – UERN

FACULDADE DE CIÊNCIAS EXATAS E NATURAIS – FANAT

DEPARTAMENTO DE INFORMÁTICA – DI

ALAN DOUGLAS CARVALHO REBOUÇAS

**Desenvolvimento de uma Metaheurística BRKGA com Infecção Viral para o  
Problema de Atribuição de Localidades a Anéis em Redes SONET/SDH**

MOSSORÓ - RN

2018

Alan Douglas Carvalho Rebouças

**Densenvolvimento de uma Metaheurística BRKGA com Infecção Viral para o  
Problema de Atribuição de Localidades a Anéis em Redes SONET/SDH**

Monografia apresentada à Universidade do Estado do Rio Grande do Norte como um dos pré-requisitos para obtenção do grau de bacharel em Ciência da Computação, sob orientação do Prof. Dr. Dario José Aloise.

MOSSORÓ - RN

2018

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

C331d Carvalho Rebouças, Alan Douglas  
Desenvolvimento de uma Metaheurística BRKGA com Infecção Viral para o Problema de Atribuição de Localidades a Anéis em Redes SONET/SDH. / Alan Douglas Carvalho Rebouças. - Mossoró, 2018.  
57p.

Orientador(a): Prof. Dr. Dario José Aloise.  
Monografia (Graduação em Ciência de Computação).  
Universidade do Estado do Rio Grande do Norte.

1. Ciência de Computação. 2. PALAS. 3. Otimização combinatória. 4. Redes SONET. 5. BRKGA. I. Aloise, Dario José. II. Universidade do Estado do Rio Grande do Norte.  
III. Título.

**Alan Douglas Carvalho Rebouças**

**DESENVOLVIMENTO DE UMA METAHEURÍSTICA BRKGA COM INFECÇÃO VIRAL PARA O PROBLEMA DE ATRIBUIÇÃO DE LOCALIDADES A ANÉIS EM REDES SONET/SDH**

Monografia apresentada como pré-requisito para a obtenção do título de Bacharel em Ciência da Computação da Universidade do Estado do Rio Grande do Norte – UERN, submetida à aprovação da banca examinadora composta pelos seguintes membros:

Aprovada em: 15/06/2018

**Banca Examinadora**



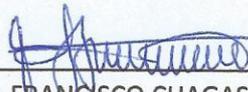
---

Prof. Dr. DARIO JOSÉ ALOISE  
Universidade do Estado do Rio Grande do Norte - UERN



---

Prof. Dr. CARLOS HEITOR PEREIRA LIBERALINO  
Universidade do Estado do Rio Grande do Norte - UERN



---

Prof. Dr. FRANCISCO CHAGAS DE LIMA JÚNIOR  
Universidade do Estado do Rio Grande do Norte - UERN

*À minha família, especialmente aos meus pais, Algecira e José Maurílio, e ao meu irmão Alex.*

## **AGRADECIMENTOS**

Agradeço primeiramente ao meu Senhor Jesus Cristo, por ter me ajudado e me dado forças para a chegar até aqui, sem Ele nada seria possível.

À minha família, em especial aos meus pais, José Maurílio e Algecira, ao meu irmão Alex, por todo incentivo, cuidado, por serem meu alicerce e por me apoiarem sempre, não medindo esforços para me ajudar nos meus objetivos e não me deixar desistir nunca nos momentos de dificuldade.

À minha namorada, Juliete, pelo carinho, amor e apoio. Por sempre estar presente em todos os momentos e estar disposta a enfrentar as dificuldades comigo.

Ao professor Dario, pelas orientações, pela paciência e por ter dedicado parte do seu tempo me ajudando.

A Jeferson Queiroga e Jonathan Lopes pela ajuda na implementação dos algoritmos.

Aos professores do DI/UERN, pela contribuição para o meu desenvolvimento como profissional.

A todos os amigos que fiz durante a graduação, pelos agradáveis momentos que compartilhamos.

A todos que fizeram parte e me ajudaram direta ou indiretamente na minha formação até aqui, obrigado.

*Porque dEle e por Ele, e para Ele, são  
todas as coisas; glória, pois, a Ele  
eternamente. Amém.*

*Romanos 11:36*

## RESUMO

Este trabalho apresenta uma proposta de solução para um problema de otimização combinatória presente no planejamento físico de redes de telecomunicações, conhecido na literatura como: Problema de Atribuição de Localidades a Anéis SONET (do inglês, *Synchronous Optical Network*), denotado pela sigla PALAS. Esse problema consiste em particionar um conjunto de localidades (clientes) em anéis disjuntos de modo a satisfazer uma série de restrições ao menor custo possível. Cada anel deverá possuir a mesma capacidade e cada localidade será atribuída a um único anel. O tráfego entre localidades pertencentes a anéis diferentes ficará a cargo de um anel especial denominado anel federal. O problema surge devido ao alto custo dos equipamentos utilizados na implementação da rede e tem como objetivo minimizar o custo de instalação projetando uma rede com um número mínimo de anéis atendendo às restrições de capacidades destes. Por se tratar de um problema NP-Difícil, algoritmos de soluções exatas se mostram ineficientes, por não conseguirem uma resolução em um tempo hábil. Diante disso, foi necessário buscar métodos aproximativos (metaheurística) para encontrar boas soluções (eventualmente ótimas) em tempo computacional viável. Nesse trabalho foi proposta a estratégia metaheurística BRKGA (*Biased Random-Key Genetic Algorithm - Algoritmo Genéticos com Chaves Aleatórias Viciadas*), acrescida de um procedimento de Infecção Viral, como tentativa otimizar a busca. Os resultados computacionais são apresentados mostrando a viabilidade de utilização da técnica desenvolvida.

**Palavras-chave:** PALAS, otimização combinatória, redes SONET, metaheurística BRKGA, infecção viral, NP-Difícil.

## ABSTRACT

This work presents a proposal for a solution to a combinatorial optimization problem present in the physical planning of telecommunications networks, known in the literature as: SONET Ring Assignment Problem, denoted by the acronym SRAP. This problem consists of partitioning a set of localities (clients) into disjoint rings in order to satisfy a series of constraints at the lowest possible cost. Each ring must have the same capacity and each location will be assigned to a single ring. Traffic between locations belonging to different rings will be handled by a special ring called the federal ring. The problem arises due to the high cost of the equipment used in the implementation of the network and aims to minimize the cost of installation by designing a network with a minimum number of rings meeting the capacity constraints of these. Because it is an NP-Hard problem, algorithms of exact solutions are inefficient because they cannot solve in a timely manner. Therefore, it was necessary to search for approximate (metaheuristic) methods to find good (eventually optimal) solutions in viable computational time. In this work the metaheuristic strategy BRKGA (Biased Random-Key Genetic Algorithm), plus a Viral Infection procedure, was proposed as an attempt to optimize the search. The computational results are presented showing the feasibility of using the developed technique.

**Keywords:** SRAP, combinatorial optimization, SONET networks, BRKGA, viral infection, NP-Hard.

## LISTA DE SIGLAS

ADM	Multiplexador <i>Add/Drop</i>
BRKGA	<i>Biased Random-Keys Genetic Algorithm</i>
DCS	<i>Digital Cross Connect System</i>
ITU	<i>International Telecommunication Union</i>
IV	Infecção Viral
OC	<i>Optical Carrier</i>
PALAS	O problema de atribuição de localidades a anéis SONET
PDH	<i>Plesiochronous Digital Hierarchy</i>
RKGA	<i>Random-Keys Genetic Algorithm</i>
SDH	<i>Synchronous Digital Hierarchy</i>
SONET	<i>Synchronous Optical Network</i>
STM	<i>Synchronous Transport Module</i>
STS	<i>Synchronous Transport Signal</i>

## LISTA DE FIGURAS

Figura 1 - Sobrevivência de uma rede SONET com topologia em anel .....	22
Figura 2 - Topologia em anel de uma rede SONET .....	23
Figura 3 - Configuração inicial da rede .....	25
Figura 4 - Exemplo de rede com topologia em anel formada a partir da configuração inicial .....	25
Figura 5 - Etapas do algoritmo genético .....	30
Figura 6 - Processo evolutivo do BRKGA .....	33
Figura 7 - Estrutura do BRKGA .....	35
Figura 8 - Exemplo de um vírus com representação do tipo inteiro .....	36
Figura 9 - Representação do cromossomo .....	38
Figura 10 - Representação do cromossomo decodificado .....	39
Figura 11 - Exemplo de solução para o PALAS .....	40
Figura 12 - Processo de transcrição .....	42
Figura 13 - Processo de Simbiose .....	43
Figura 14 - Gráfico comparativo do resultados .....	46

## LISTA DE TABELAS

Tabela 1 - Hierarquia SONET/SDH .....	21
Tabela 2 - Exemplo de matriz de demandas para o problema PALAS .....	41
Tabela 3 - Resultados para as instâncias C1 .....	46
Tabela 4 - Resultados para as instâncias C2 .....	49
Tabela 5 - Resultados para as instâncias C3 .....	53
Tabela 6 - Comparação dos resultados obtidos .....	53

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>17</b>
<b>2 PROBLEMA DE ATRIBUIÇÃO DE LOCALIDADES A ANÉIS SONET .....</b>	<b>19</b>
2.1 O PADRÃO SONET .....	19
2.2 O SINAL SONET .....	20
2.3 TOPOLOGIA DE UMA REDE SONET .....	21
2.4 DESCRIÇÃO DO PROBLEMA .....	23
2.5 DESCRIÇÃO FORMAL DO PALAS .....	25
2.5.1 FORMULAÇÕES MATEMÁTICAS .....	26
<b>3 MÉTODOS DE PESQUISA .....</b>	<b>27</b>
3.1 ALGORITMOS EVOLUTIVOS .....	28
3.2 ALGORITMOS GENÉTICOS .....	29
3.3 ALGORITMO GENÉTICO COM CHAVES ALEATÓRIAS .....	32
3.4 ALGORITMO GENÉTICO COM CHAVES ALEATÓRIAS VICIADAS .....	33
<b>4 INFECÇÃO VIRAL .....</b>	<b>35</b>
4.1 O VÍRUS .....	36
4.2 TAXA DE INFECTIVIDADE .....	37
4.3 A INFECÇÃO .....	37
<b>5 BRKGA COM INFECÇÃO VIRAL APLICADO AO PALAS .....</b>	<b>37</b>
5.1 REPRESENTAÇÃO DO PROBLEMA .....	38
5.2 FUNÇÃO APTIDÃO .....	39
5.3 DECODIFICADOR .....	39
5.4 INFECÇÃO VIRAL APLICADA AO BRKGA .....	41
5.4.1 SIMBIOSE .....	43
<b>6 RESULTADOS COMPUTACIONAIS .....</b>	<b>44</b>
<b>7 CONCLUSÃO .....</b>	<b>55</b>
REFERÊNCIAS	

## 1 INTRODUÇÃO

Atualmente, o serviço de telecomunicação tem um papel imprescindível na sociedade. Ao passo que novas tecnologias são introduzidas ao mercado, cresce o número de serviços e produtos, tais como, transferências de arquivos de dados, voz, imagens, transações bancárias, reservas de passagens aéreas, entre outros, provocando o aumento da demanda de usuários atraídos pela praticidade e inovação. Toda sociedade acompanha e se beneficia com a evolução das redes de telecomunicações e com suas inovações, contudo, este crescimento exige uma maior atenção na infraestrutura de rede, para manter o mesmo nível de qualidade de serviço para todos os clientes.

Apesar dos benefícios trazidos pelo desenvolvimento das redes de telecomunicações, surgem agora novos problemas referentes a infraestrutura da rede em virtude dessa expansão, ou seja, é necessário um aumento da capacidade da rede à medida que a demanda pelos serviços oferecidos cresce. A consequência disto, é que os problemas de planejamento de redes de telecomunicações, mesmo com a alta qualidade dos equipamentos desenvolvidos, têm se tornado cada vez maiores e complexos (OMIDYAR; ALDRIDGE, 1993) e (WANSEM; WU; CARDWELL, 1994).

Muitos desses problemas são resolvidos utilizando técnicas de otimização combinatória. Um dos aspectos importantes que reforçam o uso da otimização combinatória na resolução de problemas relacionados ao planejamento de redes de telecomunicações é o fato de que podemos representar esses problemas por meio de grafos, onde se deseja encontrar um subconjunto de vértices e (ou) arestas que satisfaça a determinados requisitos e que otimize uma função objetivo.

O problema de atribuição de localidades a anéis SONET ou PALAS, objeto de estudo deste trabalho, faz parte da etapa de planejamento físico de uma rede de *backbone*. O mesmo é um problema de otimização combinatória, e o propósito é definir e estabelecer as ligações entre as localidades da rede, a um baixo custo, respeitando um conjunto de restrições. Dentre essas restrições, podemos citar como fundamental, a capacidade da rede de se manter ativa, ou seja, a questão de sobrevivência da rede em caso de falhas. A topologia em anel em conjunto com a

tecnologia SONET (*Synchronous Optical Network*) são bastante eficazes para cumprir esse requisito.

O problema descrito acima é NP-Difícil (GOLDSCHMIDT; LAUGIER; OLINICK, 2003), ou seja, não existem algoritmos exatos que possam resolvê-lo em tempo computacional viável. Nesse caso, faz-se necessário o uso de métodos heurísticos para a solução do mesmo, que embora não garantam uma solução ótima, oferecem soluções que cheguem próximas do ótimo ou até mesmo no ótimo, em tempo computacional viável. Diversos métodos já foram aplicados neste problema, e continuam a surgir novos métodos pela integração de métodos heurísticos, conhecidos também como metaheurísticas simples ou híbridas.

O objetivo deste trabalho é explicar o PALAS e apresentar uma metaheurística híbrida como proposta de solução para o problema, descrevendo características da implementação e fazer uma análise e comparação dos dados obtidos. A metaheurística desenvolvida neste trabalho é composta de um algoritmo BRKGA acrescido de uma infecção viral. Cada fase dos algoritmos propostos será estudada detalhadamente durante este texto.

O trabalho está organizado da seguinte forma: no capítulo 2 é feita uma apresentação de forma detalhada do problema de atribuição de localidade a redes SONET, mostrando as particularidades do padrão SONET, sinal de transporte e uma explicação de como funciona a topologia de uma rede SONET. É feita também a descrição formal do problema e mostradas as formulações matemáticas utilizadas. No capítulo 3 são apresentados os fundamentos, características e algoritmos em que a metaheurística é baseada. No capítulo 4 é feita uma abordagem detalhando todos os aspectos da infecção viral. No capítulo 5 é explicada a metaheurística desenvolvida como proposta de solução para o problema. Os experimentos e resultados computacionais são descritos no capítulo 6. Por fim, a conclusão é apresentada no capítulo 7.

## 2 PROBLEMA DE ATRIBUIÇÃO DE LOCALIDADES A ANÉIS SONET

O planejamento de uma rede SONET com topologia em anel é algo bastante complexo (OMIDYAR; ALDRIDGE, 1993) e (WANSEM; WU; CARDWELL, 1994). Este planejamento é dividido em duas partes: Projeto físico e lógico. O projeto físico busca determinar os subconjuntos de localidades que darão origem aos anéis. O projeto lógico trata do estabelecimento das conexões entre as localidades. Neste capítulo, abordaremos os detalhes e as características do padrão SONET e a questão que envolve a determinação dos subconjuntos de localidades que compõem os anéis de uma rede SONET, também conhecida como problema de atribuição de localidades a anéis SONET. Veremos posteriormente que o PALAS é um problema de particionamento de grafos e tratado com técnicas de otimização combinatória.

Este capítulo foi organizado da seguinte maneira: na seção 2.1 e 2.2 temos uma abordagem detalhada sobre o padrão SONET e o seu sinal base de transporte, na seção 2.3 consta uma breve explicação de como funciona a topologia da rede SONET em anel, na seção 2.4 é apresentado o problema de atribuição de localidades a anéis SONET e por fim nas seções 2.5 e 2.5.1 a descrição formal do problema e as formulações matemáticas utilizadas.

### 2.1 O PADRÃO SONET

O SONET (*Synchronous Optical Network*) é um padrão para redes de telecomunicações óticas de alta velocidade e alta capacidade utilizado para multiplexação TDM (Time Division Multiplexing) com altas taxas de bits, tendo a fibra ótica como meio físico preferencial de transmissão. Criado em 1985 pelo o ECSA (*Exchange Carriers Standards Association*), dos Estados Unidos, com o objetivo de utilizar os recursos proporcionados pelo advento da fibra ótica.

Em 1988, o ITU-T (*International Telecommunication Union – Telecommunication Standardization Sector*) propôs o SONET de padrão mundial, chamado SDH (*Synchronous Digital Hierarchy*). Os padrões diferem somente na nomenclatura dos sinais utilizados.

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 padrão SONET traz vantagens significativas em relação aos antigos sistemas de telecomunicação, 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, e maior flexibilidade na configuração e disponibilidade de banda.

Estima-se que o modelo completo do SONET proporcione a infraestrutura de transporte adequada para as telecomunicações até o ano 2020, aproximadamente (Perros, H. G., 2005).

## 2.2 O SINAL SONET

O sinal SONET permite a transmissão de diferentes capacidades através de uma hierarquia ótica, flexível e síncrona, utilizando-se de uma multiplexação chamada *Byte-Interleaving*, que simplifica a multiplexação e oferece gerenciamento de rede ponto-a-ponto.

STS-1 (do inglês, *Synchronous Transport Signal level 1*) representa a taxa básica do sinal SONET. A partir dessa, todas as outras taxas são obtidas simplesmente multiplicando seu valor básico. Assim, o STS-3 é obtido multiplicando-se o STS-1 por 3. A partir do STS-3, todos os outros níveis hierárquicos pulam em fatores de 4.

Embora o SONET e o SDH tenham a estrutura do quadro básico diferente, o SONET inicia com um canal chamado STS-1, de 51,84 Mbit/s, o sistema SDH começa com um canal designado de STM-1 (do inglês, *Synchronous Transport Module level 1*), de 155,52 Mbit/s. Os padrões permitem interoperabilidade entre si, já que convergiram para taxa de bits comuns e tamanho de quadros iguais, ou seja, para cada canal no SONET, existe um equivalente no SDH. Vemos essa comparação das hierarquias e suas equivalências na tabela 1.

Tabela 1 - Hierarquia SONET/SDH

Sinal SONET	Taxa de bits (Mb/s)	Equivalente SDH
STS-1, OC-1	51,84	STM-0
STS-3, OC-3	155,52	STM-1
STS-12, OC-12	622,08	STM-4
STS-48, OC-48	2488,32	STM-16
STS-192, OC-192	9953,28	STM-64
STS-768, OC-768	39813,12	STM-256

Fonte: (Pioro, M.; Medhi, D, 2004)

### 2.3 TOPOLOGIA SONET/SDH

Existem diversas topologias de rede disponíveis para serem utilizadas com o padrão SONET/SDH, a escolha depende da necessidade visualizada pelo projetista. As topologias base são: ponto a ponto, ponto multiponto, barramento e anel. Sendo a mais comumente utilizada, a topologia em anel. Neste trabalho faremos uso da topologia em anel com a tecnologia SONET/SDH que envolve uma estrutura hierárquica em que múltiplos anéis disjuntos são conectados através de um anel especial denominado Anel Federal (GOLDSCHMIDT; LAUGIER; OLINICK, 2003).

A topologia em anel consiste em um conjunto de cliente conectados através de um ou mais anéis. Cada cliente presente em um anel local envia, recebe e transmite sinais por meio de um dispositivo *Add-Drop Multiplexer* (ADM) (W.J. Goralski, 2002). Vários ADMs podem ser organizados em uma configuração anelar, tanto para tráfego unidirecional quanto para bidirecional.

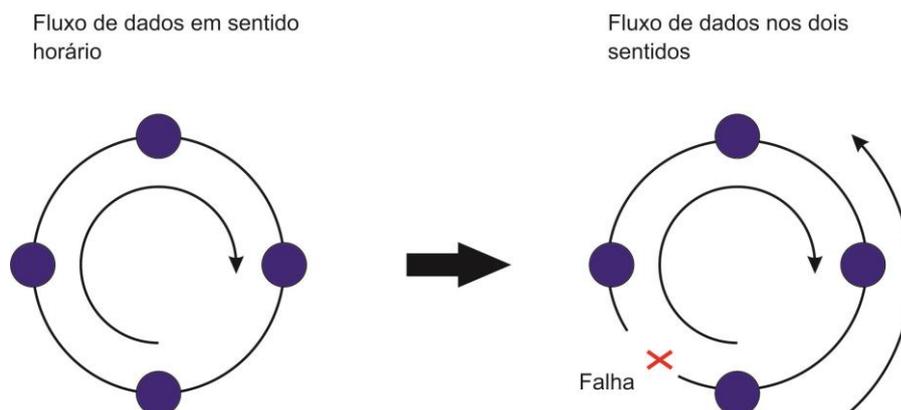
Os anéis existentes na rede podem ou não se comunicar entre si, caso isso ocorra, essa comunicação será realizada por um dispositivo chamado *Digital Cross Connect System* (DCS), onde um anel especial denominado anel federal é formado por estes dispositivos, e faz a ligação de todos os anéis locais.

A capacidade de tráfego de um anel local corresponde à largura de banda dos ADMs presentes no anel. Essa capacidade deve comportar a soma das demandas de todas as localidades pertencentes a esse mesmo anel. No anel federal a capacidade é limitada pela largura de banda dos DCSs utilizadas e deve comportar a soma das demandas dos anéis da rede.

Como apresentado anteriormente, uma restrição fundamental é a capacidade da rede de se manter ativa, ou seja, a questão de sobrevivência da rede em caso de falhas (LAGUNA, 1994). Esta característica de sobrevivência é tratada nas redes de *backbone* adotando-se a topologia em anel.

A Figura 1 exemplifica a ocorrência de falha num anel SONET onde ocorre 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).

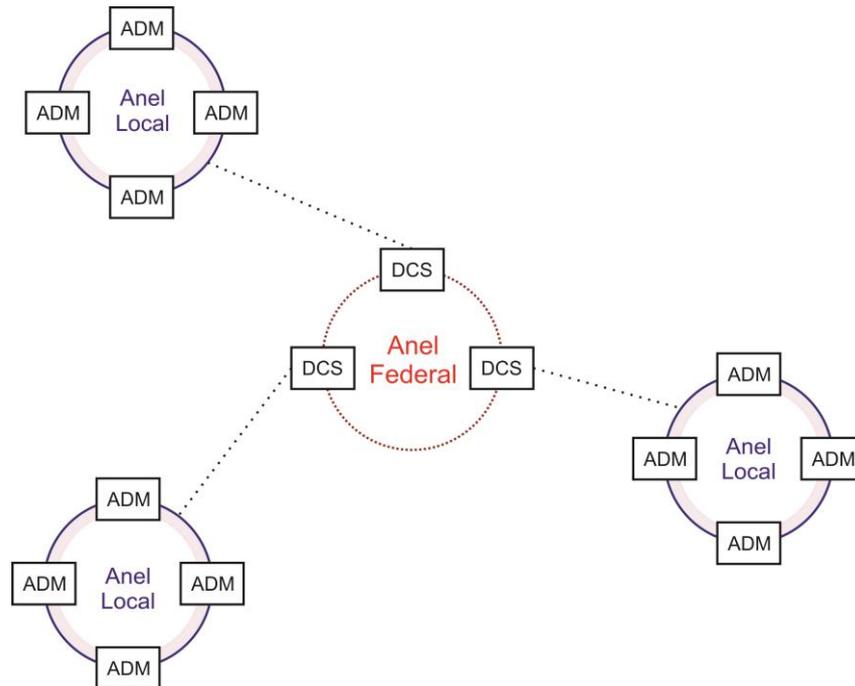
Figura 1 - Sobrevivência de uma rede SONET com topologia em anel



Fonte: Autoria própria

Na Figura 2 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 DCSs que compõem o anel federal.

Figura 2 - Topologia em anel de uma rede SONET



Fonte: Autoria própria

## 2.4 DESCRIÇÃO DO PROBLEMA

O problema PALAS foi inicialmente proposto por Goldschmidt (2001) e pertence à classe NP-Difícil (GOLDSCHMIDT; LAUGIER; OLINICK, 2003), ou seja, algoritmos exatos eficientes (de complexidade polinomial) não são conhecidos, podendo inclusive, nem existir. Surge na fase de planejamento físico de uma rede de *backbone*. No problema em questão, cada localidade cliente deve ser atribuída a um único anel SONET e os anéis locais são conectados por um anel especial, chamado anel federal.

Como vimos na seção 2.3 uma rede SONET possui um conjunto de  $n$  localidades, as quais, são agrupadas em anéis que enviam, recebem e transmitem sinais através de um equipamento instalado em cada localidade chamado ADM. Quando temos localidades situadas em anéis diferentes que precisam se comunicar, utilizamos um sistema de conexão denominado DCS.

O PALAS acontece devido ao custo dos equipamentos envolvidos na construção da rede, na qual, está diretamente relacionado ao custo dos equipamentos DCSs, tendo em vista que o custo dos ADMs diminuiu bastante nos

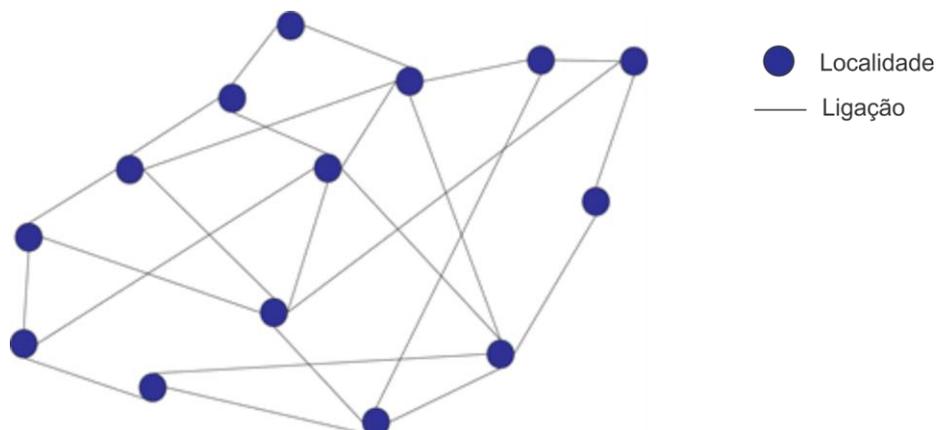
últimos anos e o custo dos cabos de fibra ótica possuem pequeno peso nesse total. Sendo assim, uma rede SONET com topologia em anel pode sair bastante caro, pois quanto mais anéis mais DCSs serão necessários para fazer essa ligação.

Dessa forma, uma solução ótima para o PALAS consiste em encontrar uma atribuição para as localidades que minimize o número de anéis locais e este, por sua vez, corresponde a quantidade de DCSs a serem instalados. Devido a sua grande complexidade, diferentes conjuntos de restrições e considerações podem ser propostas para a resolução do PALAS, lembrando que o objetivo principal é minimizar o custo total da rede. Então, semelhantemente aos trabalhos encontrados na literatura, consideraremos as seguintes restrições:

- Cada localidade tem que ser atribuída a um único anel;
- A capacidade máxima de cada anel, inclusive a do anel federal, é limitada por um valor comum, representado por  $B$ ;
- A ligação entre cada anel representado por um DCS ocorre por meio de um anel federal.

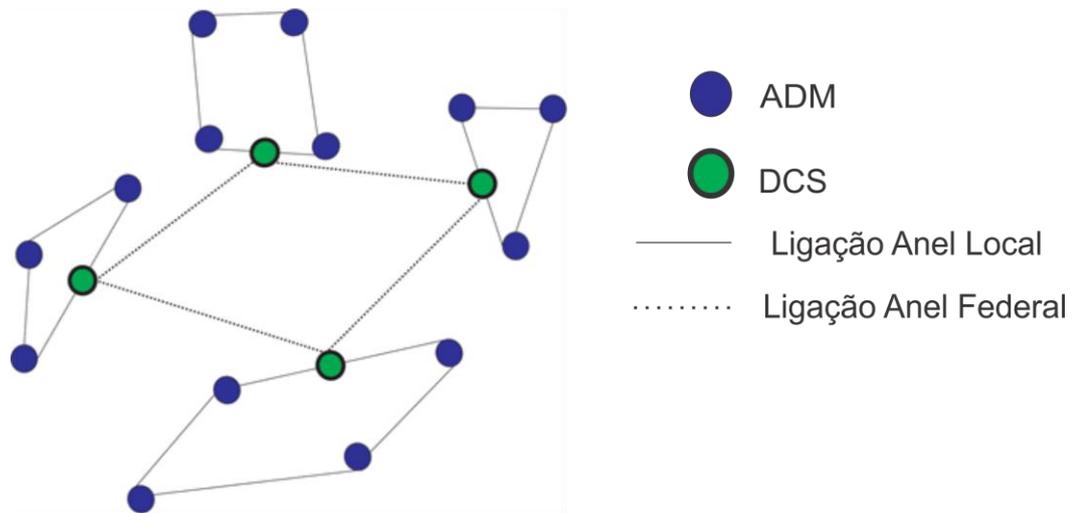
Como exemplo vejamos a figura 3 que apresenta o estado inicial de uma rede de telecomunicações. Essa rede seguindo a topologia em anel é descrita na figura 4, onde as localidades são divididas em agrupamentos (ou anéis) após, sendo estes agrupamentos conectados pelo anel federal, que está representado pela linha pontilhada.

Figura 3 - Configuração inicial da rede



Fonte: Autoria própria

Figura 4 - Exemplo de rede com topologia em anel formada a partir da configuração inicial



Fonte: Autoria própria

## 2.5 DESCRIÇÃO FORMAL DO PALAS

Como serão determinados agrupamentos (ou anéis) de localidades, podemos descrever o PALAS como um problema de particionamento dos vértices de um grafo (Mehrotra et al., 1998), onde as localidades representam os vértices e os pesos nas arestas a demanda de tráfego das localidades. Assim, o problema pode ser formulado como um problema de decisão da seguinte forma (GAREY e JOHNSON, 1979):

Seja  $G = (V, E)$  um grafo não orientado e um inteiro positivo  $B$ . Em cada aresta  $(u, v)$  pertencente a  $E$  está associado um inteiro não negativo  $d_{uv}$ . Uma solução viável do problema, corresponde a uma partição do conjunto  $V$  de vértices em  $k$  subconjuntos disjuntos  $V_1, V_2, \dots, V_k$ , tal que:

$$\sum_{u,v \in V_i, u < v} d_{uv} + \sum_{u \in V_i, v \notin V_i} d_{uv} \leq B \quad \text{onde } i = 1, 2, \dots, K \quad (1)$$

$$\sum_{i=1}^{k-1} \sum_{j=i+1}^k \sum_{u \in V_i} \sum_{v \in V_j} d_{uv} \leq B \quad (2)$$

A restrição definida pela equação 1 estabelece o limite de capacidade  $B$  aos anéis locais da rede. Por sua vez, a equação 2 impõe o limite  $B$ , de tráfego máximo, ao anel federal. O inteiro  $d_{uv}$  corresponde à demanda de tráfego entre as localidades  $u$  e  $v$  em ambos os sentidos. Se não houver demanda entre  $u$  e  $v$  então o valor de  $d_{uv}$  será zero.

### 2.5.1 FORMULAÇÕES MATEMÁTICAS

As fórmulas a seguir foram introduzidas por (BASTOS, 2005), e serão utilizadas nos algoritmos propostos.

Considerando que  $S$  seja uma solução do problema com  $K$  anéis, os valores associados aos tráfegos em um anel de índice  $a$  são:

$$demI_a = \sum_{u,v \in a, u < v} d_{uv} \quad (3)$$

$$demE_a = \sum_{u \in a, v \notin a} d_{uv} \quad (4)$$

$$demT_a = demI_a + demE_a \quad (5)$$

As equações 3, 4 e 5 representam, respectivamente, as demandas Interna, externa e total no anel  $a$ . O anel  $a$  é viável quando  $demT_a$  for menor ou igual a  $B$ . A demanda total no anel federal da solução  $S$  é dada por:

$$demAF = \frac{\sum_{i=1}^k demE_i}{2} \quad (6)$$

O anel federal será viável se  $demAF$  for menor ou igual a  $B$ . Se todos os anéis (inclusive o anel federal) forem viáveis então  $S$  é uma solução viável.

### 3 MÉTODOS DE PESQUISA

Assim como os vários problemas da otimização combinatória o PALAS é difícil de ser resolvido nos levando a procurar soluções aproximativas através de métodos heurísticos, que mesmo que não sejam exatas nos dariam uma solução próxima do ótimo, já que o tempo para se conseguir uma resposta para a classe desse tipo de problema (NP-difícil) é inviável (Goldschmidt, O.; Laugier, A.; Olinick, E. V., 2003).

A heurística é um conjunto de regras e métodos que conduzem à descoberta, à invenção e à resolução de problemas. Métodos heurísticos, de maneira geral, são algoritmos polinomiais que buscam por boas soluções (próximas da ótima). Geralmente utiliza-se uma heurística para encontrar soluções plausíveis para problemas NP-Difíceis.

Os principais aspectos de uma heurística são a rapidez e uma estratégia aplicada a um projeto específico. Espera-se que uma heurística consiga produzir uma solução satisfatória em uma quantidade de tempo razoável mesmo quando a entrada do problema aumenta exponencialmente. Já as estratégias se referem a possíveis maneiras de como podemos gerar uma solução. Duas são as estratégias comumente utilizadas: estratégia gulosa e busca local (Macambira, 2003).

As metaheurísticas, por sua vez, são um conjunto de conceitos que podem ser usados para definir métodos heurísticos que podem ser aplicados para um amplo conjunto de diferentes problemas. Em outras palavras, uma metaheurística pode ser vista como um framework algorítmico genérico que pode ser aplicado a diferentes problemas de otimização com relativamente poucas modificações para adaptá-lo a um problema específico (Metaheuristics Network Website, 2006). Uma qualidade nas metaheurísticas em relação a heurísticas tradicionais é a capacidade das mesmas se livrarem de ótimos locais, os quais podem consistir em soluções prematuras para a resolução do problema.

As metaheurísticas mais utilizadas, atualmente, na literatura são: Algoritmos Genéticos (Goldberg, 1989), Simulated Annealing (Beasley et al., 1992), GRASP (Greedy Randomized Adaptive Search Procedure) (Feo et al., 1995), VNS (Variable Neighbourhood Search) (Mladenovic et al., 1997), ACO (Ant Colony Optimization) (Bonabeau et al., 1999) e Busca Tabu (BT) (Glover, 1997 e Hansen, 1986) e mais

recentemente o Algoritmo genético com chaves aleatórias viciadas (BRKGA) (veja RESENDE, 2013) que é o tema de estudo deste trabalho.

Neste capítulo faremos uma apresentação detalhando as técnicas utilizadas para a solução do problema em estudo. Mais precisamente, uma abordagem sobre algoritmos evolutivos, algoritmos genéticos, assim como as metaheurísticas RKGA, BRKGA.

### 3.1 ALGORITMOS EVOLUTIVOS

Essa classe de algoritmos tem como base o princípio de seleção natural, proposta por Dawin em 1858. Este princípio traz 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;

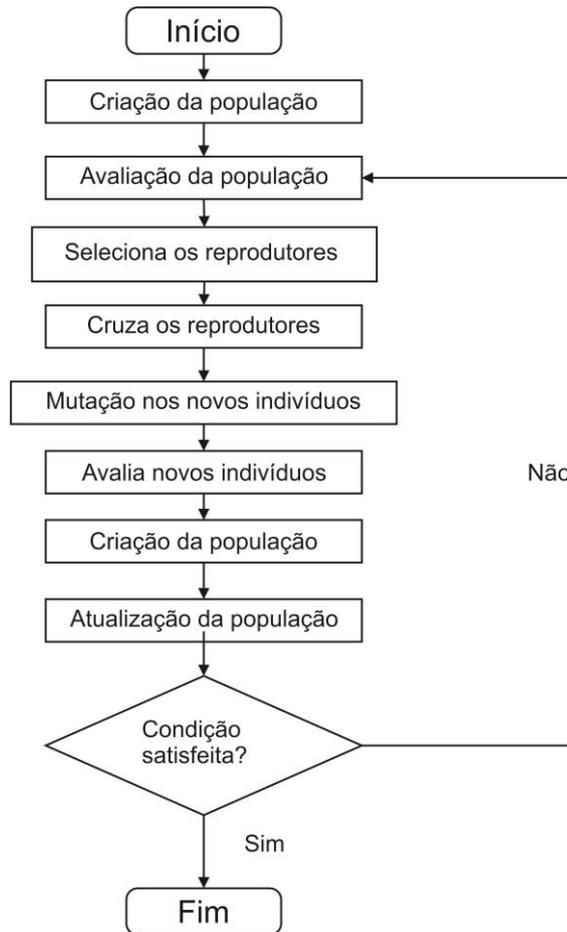
- No decorrer do processo evolutivo, as características mais necessárias para adaptações tendem a serem propagadas nas espécies, lhes dando um maior grau de adaptação.
- O propagar das características pelas espécies possibilita a ocorrência de um fenômeno conhecido como mutação. Tal fenômeno pode ser 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.

### 3.2 ALGORITMOS GENÉTICOS

Os algoritmos genéticos têm sua base na teoria de seleção natural de Darwin para resolver diversos problemas. Devido a sua flexibilidade, capacidade de alteração de adaptação tornou-se uma técnica fortemente estudada e utilizada em várias áreas do campo da otimização (OLIVEIRA, 2015).

Segundo (Holland, J. H., 1992) os algoritmos genéticos seguem etapas baseadas no processo evolutivo para resolver problemas, como podemos ver na Figura 5. O algoritmo cria uma população que representa uma possível solução para o problema em questão, onde cada indivíduo dessa população será submetido ao processo evolutivo.

Figura 5 - Etapas do algoritmo genético



Fonte: Adaptado de Holland (1992)

Os principais conceitos dos algoritmos genéticos são:

- Indivíduos

Unidade fundamental de um algoritmo genético conhecidos como cromossomos e seus valores como genes. O valor de um indivíduo está no seu fenótipo, onde cada cromossomo tem um grau de adaptação, que se refere a quão boa é a resposta para o problema, calculado por uma função objetivo. Além do grau de aptidão ou fitness que diz respeito ao nível de adaptação em relação aos demais indivíduos (GOLDBERG, 1989).

- População

Conjunto de indivíduos que em um processo evolutivo de populações passam características de adaptações e de melhorias para as próximas

gerações conseguidas por meio do cruzamento e de mutações. Representando a quantidade de vezes que uma população passou por todos os processos, como de seleção, reprodução, mutação e atualização (DAVIS, L. D., 1991).

- Inicialização

A população inicial é gerada de forma aleatória (GOLDBERG, D. E., 1989).

- Avaliação

Todos os indivíduos da população passarão por uma avaliação para que seja verificado a aptidão desse indivíduo, verificando a qualidade dessa solução para o problema tratado.

- Seleção

Nessa etapa a adaptação dos indivíduos é avaliada julgando se suas características serão propagadas por gerações. Esse processo pode ser dirigido: não aleatório e determinístico; ou cumulativo: processo onde os benefícios são passados de uma geração para outra, obtendo indivíduos mais preparados e possíveis melhores soluções.

- Reprodução

Após a etapa de seleção, na reprodução os indivíduos seguem uma probabilidade pré-estabelecida pelo cruzamento, combinando partes dos genes dos pais para gerações de filhos. A escolha de indivíduos para o cruzamento pode se dar de forma aleatória, como também por métodos celetistas (GEYER-SCHULZ, 1996).

- Mutação

A mutação ocorre nos indivíduos resultantes do cruzamento e com uma probabilidade então pré-definida de fazer algum tipo de alteração em sua estrutura. Tem como importância a diversidade que pode gerar, se bem utilizado (GOLDBERG, 1989).

- Atualização

Inserção dos indivíduos originários do cruzamento e mutação na população, tomando o lugar daqueles que lhes deram origem (DAVIS, 1991).

### 3.3 ALGORITMOS GENÉTICOS COM CHAVES ALEATÓRIAS

O Algoritmo genético com chaves aleatórias (RKGA – Random Key Genetic Algorithm) foi inicialmente proposto por (Bean, J. C., 1994), e teve como base os algoritmos genéticos tradicionais. Sua população de cromossomos é formada por vetores preenchidos com valores reais no intervalo  $[0, 1]$ . Cada elemento contido no vetor é denominado chave e é gerado aleatoriamente na população inicial. As soluções são geradas por um procedimento denominado decodificador, que mapeia um vetor de chaves aleatórias numa solução do problema de otimização e calcula o custo desta solução.

Neste algoritmo, um novo cromossomo é gerado através do cruzamento entre dois outros cromossomos, denominados pais. O cromossomo gerado tem maior probabilidade de herdar genes do cromossomo pai mais apto. A população inicial é gerada aleatoriamente com  $p$  indivíduos e  $n$  chaves aleatórias. Na produção de uma nova geração a população é dividida em três grupos distintos: Os Conjuntos Elite, não-elite e mutante. O Conjunto elite possui os indivíduos com melhor aptidão, os indivíduos menos aptos são mantidos no conjunto não-elite, e os mutantes são gerados aleatoriamente. O tamanho dos conjuntos e da população são ajustados na implementação.

Como está exemplificado na figura 6, o conjunto elite é copiado para as gerações seguintes, enquanto o conjunto Mutantes tem seus cromossomos substituídos por novos a cada geração, gerados de maneira aleatória. Os

cromossomos do conjunto não-elite também são substituídos por cromossomos (filhos), gerados pelo cruzamento de dois indivíduos selecionados aleatoriamente.

### 3.4 ALGORITMOS GENÉTICOS COM CHAVES ALEATÓRIAS VICIADAS (BRKGA)

O algoritmo genético de chaves aleatórias viciadas (*Biased Random-Key Genetic Algorithm* - BRKGA) é uma metaheurística evolutiva para problemas de otimização discreta e global baseada no algoritmo de chaves aleatórias de Bean (1994), cuja a solução é representada por um vetor de  $n$  chaves aleatórias. O BRKGA é constituído de dois elementos característicos: chave aleatória, um número real gerado aleatoriamente no intervalo contínuo  $[0,1]$ ; e o decodificador, componente que mapeia um vetor de chaves aleatórias em uma solução para o problema de otimização, podendo assim calcular o seu custo simplesmente ordenando o vetor de chaves, gerando assim uma permutação que corresponde aos índices dos elementos ordenados (RESENDE, 2013).

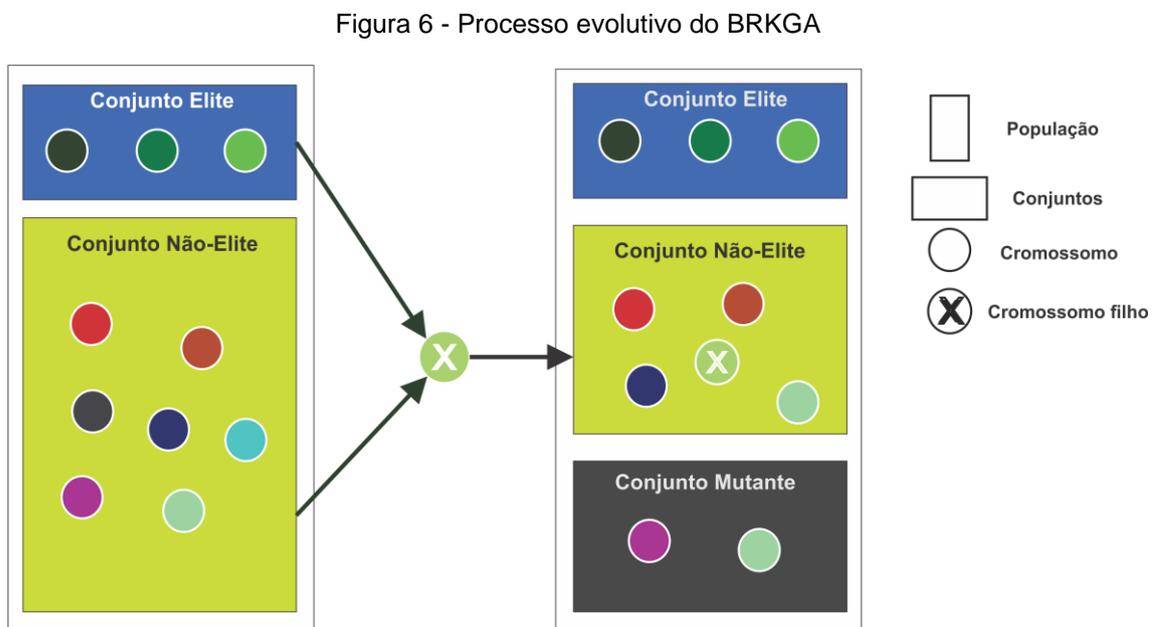
O BRKGA assim como o algoritmo de Bean, também faz uso do princípio Darwinista, onde os indivíduos mais fortes de uma população possuem mais chances de se reproduzir e passar seu material genético para as próximas gerações. Seguindo os passos do RKGA, o BRKGA tem sua população inicial gerada aleatoriamente com  $p$  vetores de  $n$  chaves aleatórias. A partir daí o algoritmo começa a evolução e a produção de novas gerações. Seguindo os passos:

i) Na  $k$ -ésima geração, os  $p$  vetores da população são particionados em um conjunto pequeno de  $p_e < p/2$  vetores que correspondem às melhores soluções, conjunto esse é chamado de *elite* e o outro conjunto com o restante da população chamado de *não-elite*.

ii) Todos os vetores *elite* são copiados, sem mudança, para a população da  $k+1$ -ésima geração, caracterizando o elitismo do princípio darwinista.

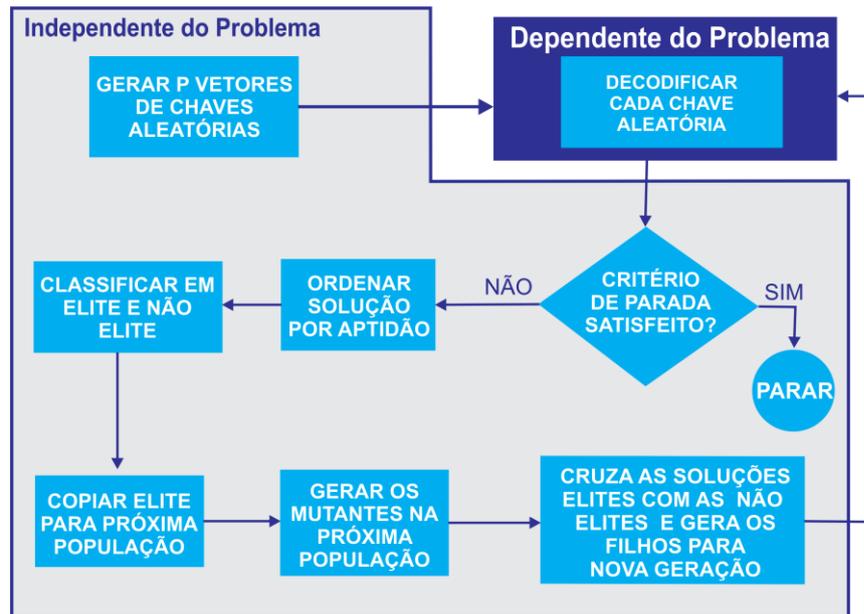
iii) Em seguida,  $p_m$  vetores de chaves aleatórias são introduzidos na população da  $k+1$ -ésima geração. Esses vetores são chamados de *mutantes* e tem a função de evitar que a população convirja para um ótimo local não global.

iv) Para completar os  $p$  elementos da população da  $k+1$ -ésima geração,  $p - p_e - p_m$  vetores são gerados combinando pares de soluções da população da  $k$ -ésima geração com a combinação uniforme parametrizada de Spears e DeJong (1991), onde uma solução é *elite* e a outra não. Podemos ver todo o processo do algoritmo na figura 6.



Fonte: Adaptado de Resende (2013).

Figura 7 - Estrutura do BRKGA



Fonte: Adaptado de Resende (2013).

A diferença entre ambos os algoritmos se dá no processo de seleção dos pais para o cruzamento e como este cruzamento é implementado. O BRKGA vai além no que se refere ao darwinismo. Tanto o BRKGA quanto o RKGA selecionam aleatoriamente os pais e com reposição. Dessa maneira, um pai pode ter mais de um filho por geração. Entretanto, o BRKGA distingue-se ao sempre selecionar um pai do conjunto *elite* e outro do conjunto *não-elite* (ou, da população inteira), enquanto o RKGA sempre seleciona os pais da população inteira. Ambos os algoritmos utilizam a combinação de pais com o cruzamento uniforme de Spears e DeJong (1991) para gerar o filho. Assim, um filho gerado no BRKGA tem maior probabilidade de herdar chaves do pai elite, no RKGA isso não acontece necessariamente. Essa diferença quase sempre faz com que o BRKGA seja superior ao RKGA (Resende, 2013).

#### 4 INFECÇÃO VIRAL

Segundo Nakahara e Sagawa (1986) a transferência de material genético de um organismo para outro, na teoria da evolução, pode ocorrer através de vírus

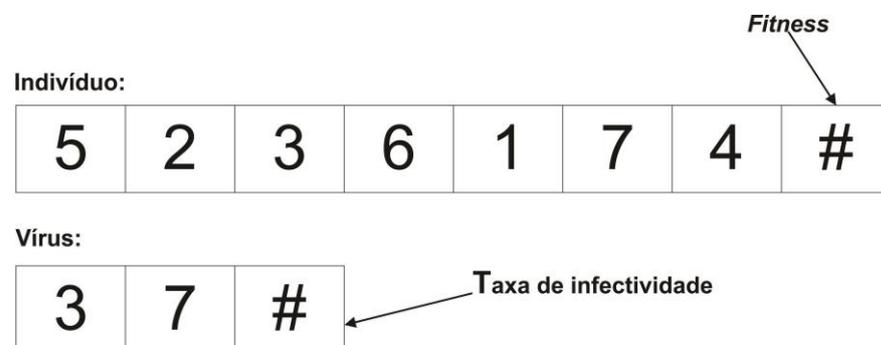
(KANOH *et al.*, 1996), isto é, um vírus transportando material genético, pode infectar um organismo e, através desta infecção, reescrever o gene do organismo infectado.

A ideia da Infecção Viral vem sendo utilizada no Algoritmo Genético em substituição ao operador de mutação, como em: Kanoh & Nakamura (2000), Kanoh *et al.* (1996), Kanoh, Kondo & Sugimoto (2002), Guedes, Leite & Aloise (2005), entre outros. E segundo Kanoh, Kondo & Sugimoto (2002), o uso da Infecção viral no Algoritmo Genético melhora a taxa de busca do algoritmo.

#### 4.1 O VIRUS

O vírus é uma solução parcial que ao infectar um indivíduo reescreve o material genético deste, aumentando ou diminuindo o seu valor de *fitness*. Considerando-se o problema do Caixeiro Viajante como exemplo, onde o indivíduo representa uma rota completa, um vírus seria um trecho (pedaço) de uma rota.

Figura 8 - Exemplo de um vírus com representação do tipo inteiro



Semelhante como ocorre na genética, nos algoritmos evolutivos o vírus necessita de um organismo para interagir e só atuará na evolução de uma geração se houver um indivíduo onde possa hospedar-se, pois o vírus não é uma solução completa, e sim, um trecho de uma solução de um determinado problema. Assim como o indivíduo possui um valor de aptidão, o vírus também traz consigo uma taxa de infectividade, que varia de acordo com sua qualidade.

## 4.2 TAXA DE INFECTIVIDADE

A taxa de infectividade diz respeito a qualidade de um vírus, ou o quanto este vírus melhora determinada solução. Todos os vírus são iniciados com a taxa de infectividade igual, independente da iteração em que se encontra o algoritmo. Esta taxa pode aumentar até um valor máximo ou reduzir a um valor mínimo, se o vírus melhorar a solução a taxa será incrementada, caso contrário, decrementada. Os valores iniciais, máximo e mínimo são pré-estabelecidos.

Segundo Noronha, Aloise & Silva (2001), o sucesso de uma heurística depende de sua capacidade de diversificação de busca quando nenhuma melhoria adicional parece possível e de intensificação da busca em regiões promissoras. A diversificação e intensificação são fatores relevantes na busca de ótimos globais em diversos problemas, no caso da infecção viral estão diretamente ligadas à taxa de infectividade.

## 4.3 A INFECÇÃO

O ato de infectar um indivíduo é chamado de transcrição (Algoritmo 2). Este processo consiste em o vírus modificar o indivíduo infectado de forma que esse contenha um trecho idêntico ao representado pelo mesmo e fazer os ajustes necessários para manter a viabilidade do agente receptor (GUEDES, LEITE E ALOISE, 2005).

## 5 BRKGA COM INFECÇÃO VIRAL APLICADO AO PALAS

Apresentaremos neste capítulo detalhes da implementação da metaheurística BRKGA e o processo de hibridização com a infecção viral, que foram os algoritmos propostos para resolução do problema de atribuição de localidades a anéis SONET neste trabalho.

Seguindo os passos citados anteriormente e fazendo uso de seus componentes (chave aleatória e decodificador) conseguimos êxito na implementação e obtenção de resultados.

## 5.1 REPRESENTAÇÃO DO PROBLEMA

Os elementos do problema, como localidades, suas respectivas ligações e demandas foram representados através de estruturas de dados que armazenam todas as informações relevantes, construindo assim os componentes característicos da metaheurística desenvolvida, como, por exemplo, indivíduos, cromossomos, genes, populações. Utilizamos listas de vetores para representar os cromossomos dos indivíduos, cada lista contém as localidades que formarão uma solução do problema. Cada localidade é representada por uma chave aleatória no intervalo de 0 e 1.

Figura 9 - Representação do cromossomo

Cromossomo								
<b>Chaves</b>	0,23455	0,16473	0,56798	0,47628	0,22343	0,69821	0,15876	0,43128
<b>Localidades</b>	1	2	3	4	5	6	7	8

Fonte: Autoria própria

A reprodução dos indivíduos é feita utilizando a combinação de pais com o cruzamento uniforme de Spears e DeJong (1991) para gerar o filho onde a probabilidade de herdar os genes do pai elite é sempre maior. Buscando evoluir as soluções geradas para a população seguinte foi copiado o grupo *elite* da população anterior. Acrescentando a essa população que está sendo evoluída filhos gerados combinando cromossomos do grupo *elite* e do *não-elite* escolhidos aleatoriamente.

## 5.2 FUNÇÃO DE APTIDÃO

Uma solução é viável se todos os anéis locais e o anel federal forem viáveis, isto é, a demanda total de todos os anéis for menos que ou igual a B. A função de aptidão precisa diferenciar soluções viáveis de inviáveis, pois podemos ter uma viável com a mesma quantidade de anéis de uma inviável. No nosso caso atribuímos uma penalidade as soluções inviáveis acrescentando a demanda desta à demanda do anel federal.

## 5.3 DECODIFICADOR

Como dito anteriormente, o BRKGA necessita de um decodificador que vai mapear as chaves aleatórias e transformá-las em uma solução real do problema. No nosso caso, temos um problema de particionamento de conjuntos em subconjuntos disjuntos. Dessa forma, o decodificador terá que ordenar o vetor chaves aleatórias e teremos outro componente que irá particionar a solução gerada dando origem aos anéis que compõem a mesma.

O algoritmo gerador de anéis recebe um cromossomo decodificado (figura 10) e passa a verificar os pontos de corte para formar o anel com base nas demandas e na capacidade de tráfego dos anéis.

Figura 10 - Representação do cromossomo decodificado

Cromossomo - Decodificado								
<b>Chaves</b>	0,15876	0,16473	0,22343	0,23455	0,43128	0,47628	0,56798	0,69821
<b>Localidades</b>	7	2	5	1	8	4	3	6

Fonte: Autoria Própria.

O gerador de anéis funciona de acordo com o seguinte algoritmo (Algoritmo 1):

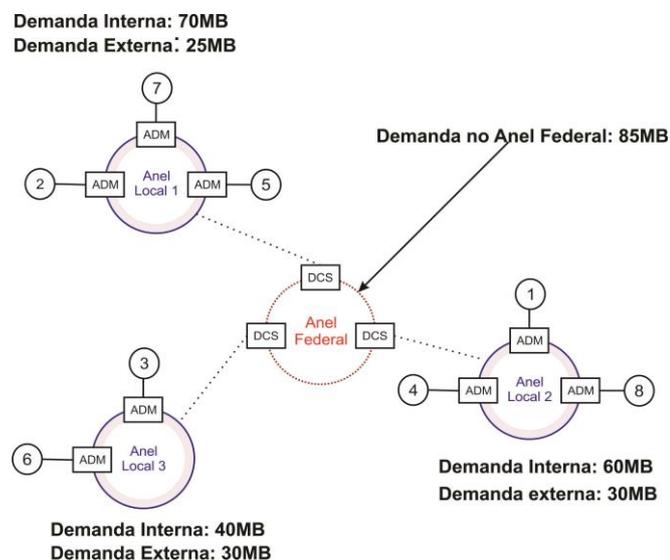
Algoritmo 1 - Procedimento para gerar anéis

1	<b>Procedimento para gerar anéis</b>
2	<b>Para</b> cada localidade faça
3	Inserir localidade no Anel
4	Demanda Total := Demanda Interna + Demanda Externa
5	Verificar Demanda Total
6	<b>Se</b> Demanda Total não é viável
7	Criar Novo Anel
8	Inserir Localidades no novo Anel
9	<b>FimSe</b>
10	<b>FimPara</b>

Fonte: Autoria própria

Após passar pelo gerador de anéis (Algoritmo 1), temos uma solução real do problema. A figura 11 exemplifica uma solução viável  $S$  com as localidades na configuração do cromossomo decodificado (figura 10) seguindo a matriz de demandas da tabela 2. A capacidade máxima do ADM e DCS adotada neste exemplo foi de 100MB, ou seja, o tráfego de dados no anel local ou federal não poderá exceder 100MB. A solução encontrada está distribuída entre três anéis locais.

Figura 11 - Exemplo de solução para o PALAS



Fonte: Autoria própria.

Tabela 2 - Exemplo de matriz de demandas para o problema PALAS

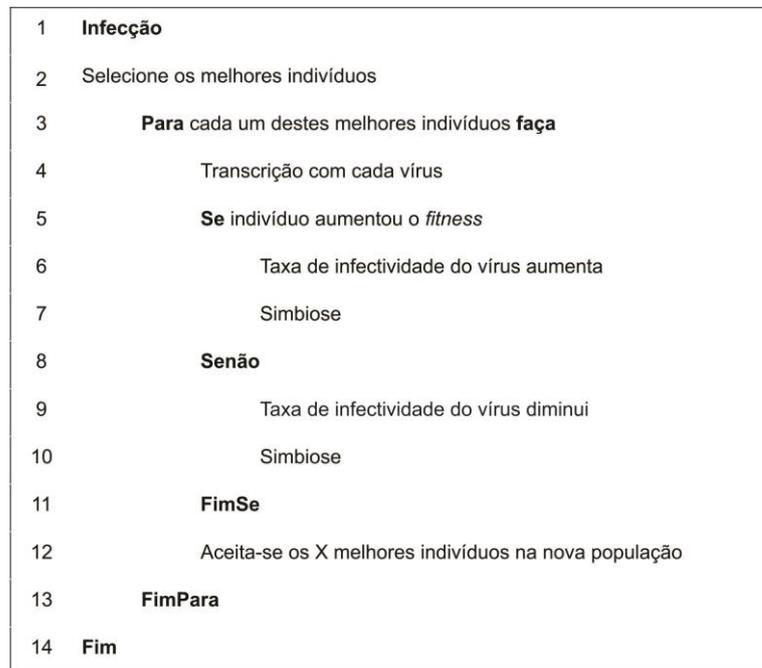
	1	2	3	4	5	6	7	8
1	0	0	0	25	0	10	0	20
2	0	0	0	0	10	0	30	15
3	0	20	0	0	0	40	0	0
4	0	0	0	0	0	0	0	15
5	0	0	0	10	0	0	30	0
6	0	0	0	0	10	0	0	0
7	0	0	0	0	0	0	0	0
8	0	20	30	0	0	0	0	0

Fonte: Autoria própria.

#### 5.4 INFECÇÃO VIRAL APLICADA AO BRKGA

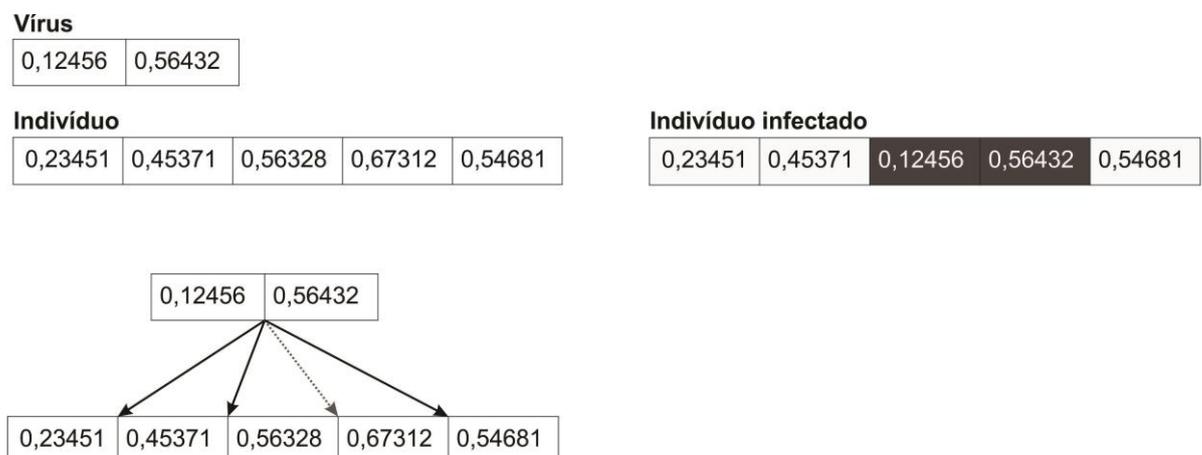
Neste trabalho, a infecção viral foi implementada utilizando uma população de vírus gerada aleatoriamente. Na infecção, metade da população-elite da geração atual, será infectada por todos os vírus da população de vírus, em seguida, serão selecionados os indivíduos que apresentaram melhor aptidão após a infecção, e estes farão parte da nova geração de indivíduos. O processo de infecção ocorre segundo o Algoritmo 2.

## Algoritmo 2 – Infecção Viral



Para realizar a transcrição, o vírus é testado entre cada par de genes do indivíduo escolhendo-se como posição para adicioná-lo aquela que o indivíduo tiver a maior aptidão. O vírus foi gerado em formato de chave aleatória. Como o BRKGA trabalha com chaves aleatórias, não é necessário se preocupar com genes repetidos, pois a infecção é feita antes da decodificação nas chaves aleatórias.

Figura 12 - Processo de transcrição



Fonte: Autoria própria

#### 5.4.1 SIMBIOSE

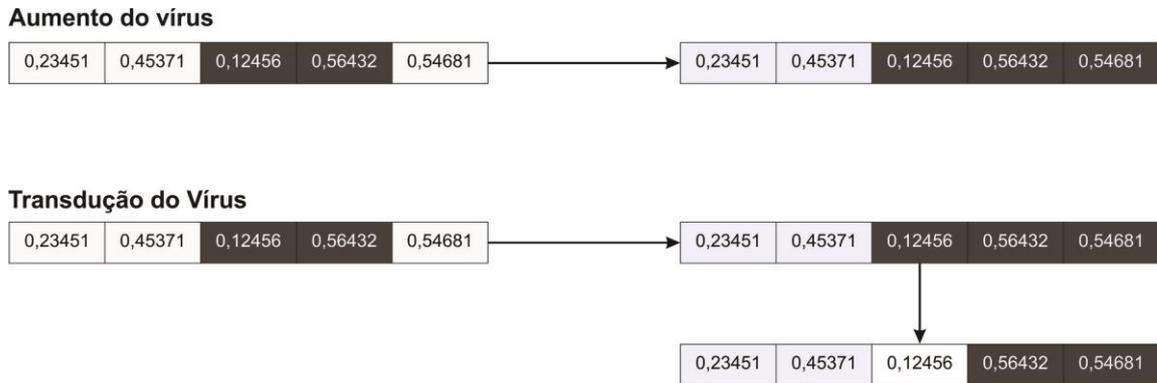
Na Biologia, a simbiose pode ser definida como uma associação a longo prazo entre dois organismos de espécies diferentes seja essa relação benéfica para ambos os indivíduos envolvidos ou não (Sapp, J., 1994).

Analogamente a este conceito, utilizou-se o processo de simbiose devido a associação entre os indivíduos infectados da população fixa e o vírus gerar um benefício mútuo, produzindo indivíduos com melhor aptidão e possíveis alterações no material genético do vírus.

Na geração inicial da população de vírus, eles terão sempre um tamanho mínimo pré-estabelecido, se durante a evolução das gerações os vírus que mantiverem bons resultados com as infecções poderão chegar até um tamanho máximo.

No ato de infectar, se um determinado vírus atingir sua taxa de infectividade máxima, ele incorpora um dos genes do indivíduo que está ao seu lado (Figura 13) transformando-se em um novo vírus, conseqüentemente, terá uma taxa de infectividade igual à inicial. Quando um vírus infectar um indivíduo e não ocorrer melhora deste, sua taxa de infectividade diminui. Esta taxa chegando a um valor mínimo, e o vírus não estando no seu tamanho mínimo, este abandonará um gene de uma de suas extremidades (aquela de maior custo), podendo retornar a um vírus existente anteriormente. O vírus passa a ter sua taxa de infectividade igual ao valor máximo ao final deste processo.

Figura 13 - Processo de Simbiose



Fonte: Autoria Própria

Pode ocorrer, ainda no processo de Simbiose, o caso de o vírus estar no seu tamanho mínimo e possuir taxa de infectividade mínima. Neste caso o vírus terá que ser eliminado e um novo vírus será gerado para substituí-lo. Neste trabalho, optou-se por gerar o novo vírus de forma aleatória.

## 6 RESULTADOS COMPUTACIONAIS

A implementação da metaheurística como proposta de resolução para o SRAP foi desenvolvida na linguagem JAVA e os testes computacionais foram executados em um PC com processador AMD FX 9370 de 4.4Ghz e 8GB de RAM no Sistema Operacional Windows 10 Pro.

Uma série de testes foram desenvolvidos para investigar o desempenho dos algoritmos propostos sobre as instâncias. O algoritmo foi executado 10 vezes sobre cada instância do problema, o critério de parada aplicado foi encontrar uma solução viável com o número de anéis igual ao da solução ótima de cada instância, ou, caso não fosse encontrada, o algoritmo era parado após 300 iterações.

O BRKGA foi configurado com os seguintes parâmetros:

- População  $P = 150$ ,
- População-elite  $Pe = 20$ ,
- Fator de herança de um pai-elite  $Pr = 0,7$ .

A taxa de infecção utilizada foi de 50%, ou seja, a cada geração, a metade da população-elite é infectada.

Os parâmetros utilizados para o vírus foram:

- Tamanho inicial do vírus: valor arredondado de 10% do tamanho do indivíduo;
- Taxa de infectividade inicial do vírus: 5;
- Taxa de infectividade mínima do vírus: 2;
- Taxa de infectividade máxima do vírus: 8.

Foram utilizados 3 conjuntos de instancias C1, C2, C3. As instancias do grupo C1 são chamadas de GLO, foram propostas por (Goldschmidt, O. et al., 2003). O conjunto de instancias GLO é dividido em dois grupos: 80 instancias geométricas e 80 instancias 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.

Pode-se notar de acordo com a tabela 3 e a figura 14 que o BRKGA com a infecção viral se saiu melhor em todas as classes de instâncias. A partir dos

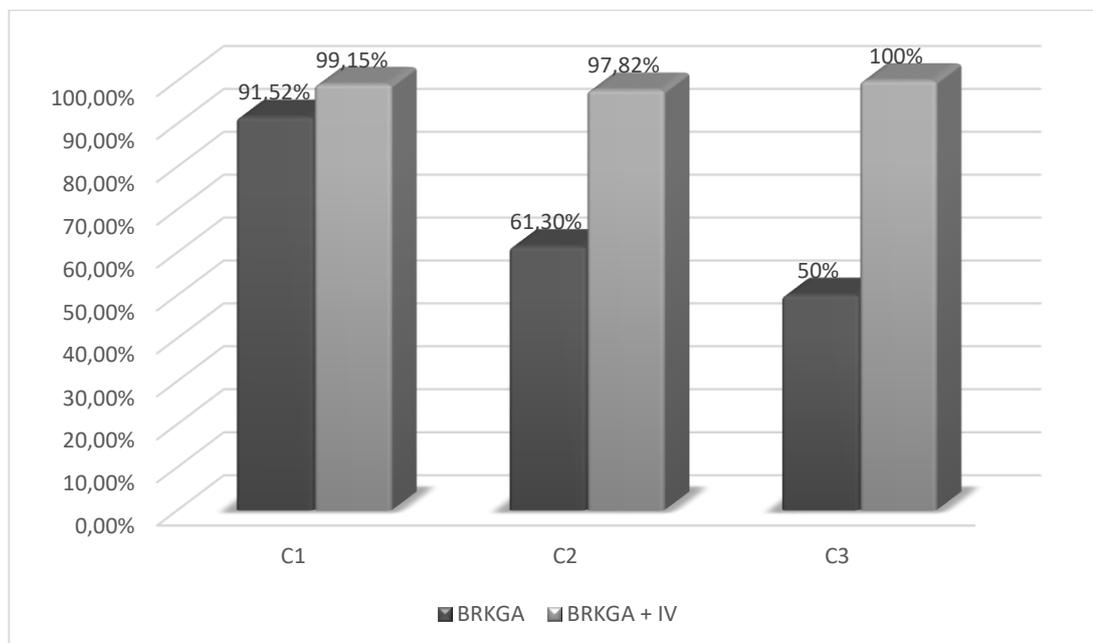
resultados, observa-se que o BRKGA não funcionou muito bem para os conjuntos de instâncias da classe C2 e C3, conseguiu encontrar apenas 61,30% e 50% das soluções ótimas, respectivamente. A estratégia de infecção viral se mostrou eficaz para a solução deste tipo de problema encontrando quase 100% das soluções ótimas das instâncias testadas. As tabelas 4, 5 e 6, em anexo, demonstram os resultados obtidos com as implementações dos algoritmos BRKGA Clássico e BRKGA com Infecção Viral para as classes C1, C2 e C3. A tabela 3 apresenta uma comparação da performance entre os mesmos.

Tabela 3 - Comparação dos resultados obtidos

Conjunto de instâncias	Instâncias	Instâncias Viáveis	BRKGA	BRKGA + IV
C1	160	118	91,52%	99,15%
C2	230	230	61,30%	97,82%
C3	40	2	50%	100%

Fonte: Autoria própria

Figura 14 - Gráfico comparativo dos resultados



Fonte: Autoria própria

## 7 CONCLUSÃO

Neste trabalho, apresentou-se uma metaheurística BRKGA acrescido da técnica de infecção viral como proposta de solução para o Problema de Atribuição de Localidades a Anéis SONET.

Extensivos experimentos computacionais foram realizados sobre as classes de instancias C1, C2 e C3. Os três algoritmos foram executados 10 vezes sobre cada uma dessas instancias utilizando os mesmos parâmetros. Os resultados computacionais foram coletados e analisados observando-se a qualidade das soluções, a média de tempo e quantidade de ótimos encontrados durante as 10 execuções de cada uma das instâncias.

Verificou-se que o BRKGA com a infecção viral obteve as melhores soluções em relação ao BRKGA simples, já que conseguiu mesclar a diversificação com a aleatoriedade do BRKGA e o método de intensificação da infecção viral, características ideais para que metaheurísticas encontrem soluções ótimas em qualquer problema.

## REFERÊNCIAS

- ARINGHIERI, R.; DELL'AMICO, M. Solution of the sonet ring assignment problem with capacity constraints. Technical Report 12, DISMI - University of Modena and Reggio Emilia, 2001.
- BASTOS, L. de O. Soluções Heurísticas para o Problema de Atribuição de Localidades a Anéis em Redes SONET. [S.l.]: Dissertação (Mestrado em Computação) – Programa de Pós-graduação em Computação, UFF, 2005.
- COOK, W. et al. *Combinatorial Optimization*. New York: John Wiley & Sons, 1998.
- 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.
- GAREY, M. R., JOHNSON, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: W.H Freeman, 1979.
- 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.; LAUGIR, 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.

GUEDES, A.C.B.; ALOISE, D.J.; FONTES F.F.C. & OLIVEIRA, W. Um Algoritmo com Infecção Viral para Resolução do Problema do Caixeiro Viajante Assimétrico. Encontro Regional de Pesquisa Operacional, artigo aceito em 2007 para publicação, aguarda impressão.

GUEDES, A. C. B.; LEITE, J. N. F.; ALOISE, D. J. *Um algoritmo genético com infecção viral para o problema do caixeiro viajante*. *PublCa 1*, p.16 – 24, 2005.

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.

KANO, H. et al. *Solving Constraint Satisfaction Problems by a Genetic Adopting Viral Infection*. IEEE, 1996.

KANO, H.; NAKAMURA, N. *Route Guidance with Unspecified Staging Posts Using Genetic Algorithm for Car Navigation Systems*. IEEE Intelligent Transportation Systems, 2000.

KANO, H., KONDO, M.; SUGIMOTO, M. *Solving Timetabling Problems Using Genetic Algorithms Based on Minimizing Conflict Heuristics*. Barcelona: CIMNE, 2002.

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.

MACAMBIRA, E. M.; MACULAN, N.; DE SOUZA, C. C. *A column generation approach for SONET ring assignment. Networks*, v. 47, p. 157–171, 2006.

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.

OLIVEIRA, T. H. F. *Abordagem híbrida para o problema de atribuição de localidades a anéis em redes SONET/SDH*. Dissertação de Mestrado – UERN/UFERSA, 2015.

OMIDYAR, C. G.; ALDRIDGE, A. *Introduction to sdh/sonet*. *Communications Magazine*, p. 30–33, 1993.

PERROS, H. G. *Connection-oriented Networks SONET/SDH, ATM, MPLS and OPTICAL NETWORKS*. 1. ed. [S.I.]:Willey, 2005.

PIORO, M.; MEDHI, D. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. [S.I.]: Morgan Kaufmann, 2004.

RESENDE, M. G. C. *Biased random-key genetic algorithms with applications in telecommunications. TOP*, v. 20, p. 130–153, 2012.

RESENDE, M. G. C.; *Introdução aos algoritmos genéticos de chaves aleatórias viciadas*, Anais do Simpósio Brasileiro de Pesquisa Operacional - SBPO, (2013).

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.I.: s.n.], 1998.

## APÊNDICE A – RESULTADOS PARA AS CLASSES DE INSTÂNCIAS

Tabela 4 - Resultados para as instâncias C1

Instâncias		BRKGA		BRKGA+IV	
Nome	Solução Ótima	T. Médio	Solução Encontrada	T. Médio	Solução Encontrada
GL_15_1	3	(0:0:1)	3	(0:0:11)	3
GL_15_2	3	(0:0:15)	3	(0:0:5)	3
GL_15_3	2	(0:0:10)	2	(0:0:32)	2
GL_15_6	2	(0:0:4)	2	(0:0:9)	2
GL_15_7	2	(0:0:3)	2	(0:0:4)	2
GL_15_8	3	(0:0:51)	3	(0:0:65)	3
GL_15_9	3	(0:0:1)	3	(0:0:5)	3
GL_15_10	2	(0:0:115)	2	(0:0:4)	2
GL_25_1	3	(0:0:35)	3	(0:0:161)	3
GL_25_2	2	(0:0:437)	2	(0:0:188)	2
GL_25_3	2	(0:0:732)	2	(0:0:281)	2
GL_25_4	3	(0:0:150)	3	(0:0:182)	3
GL_25_7	4	(0:0:544)	4	(0:0:208)	4
GL_25_8	3	(0:0:101)	3	(0:0:183)	3
GL_25_9	3	(0:0:87)	3	(0:0:184)	3
GL_30_1	3	(0:0:256)	3	(0:0:348)	3
GL_30_2	4	(0:0:245)	4	(0:0:373)	4
GL_30_3	4	(0:0:322)	4	(0:0:355)	4
GL_30_4	3	(0:0:438)	3	(0:0:391)	3
GL_30_5	4	(0:0:320)	4	(0:0:367)	4
GL_30_9	4	(0:0:251)	4	(0:0:375)	4
GL_30_10	3	(0:0:38)	3	(0:0:177)	3
GL_50_1	5	(0:3:766)	5	(0:3:971)	5
GL_50_4	4	(0:4:720)	4	(0:38:470)	4
GL_50_5	5	(0:2:130)	5	(0:3:666)	5
GL_50_6	5	(0:4:45)	5	(0:7:232)	5
GL_50_7	5	(0:3:213)	5	(0:3:744)	5
GS_15_1	3	(0:0:9)	3	(0:0:91)	3
GS_15_4	3	(0:0:4)	3	(0:0:50)	3
GS_15_7	3	(0:0:3)	3	(0:0:60)	3
GS_15_9	3	(0:0:2)	3	(0:0:5)	3
GS_25_1	4	(0:0:3)	4	(0:0:209)	4
GS_25_3	3	(0:0:21)	3	(0:0:254)	3
GS_25_7	3	(0:0:16)	3	(0:0:215)	3
GS_25_8	4	(0:0:357)	4	(0:0:385)	4
GS_30_1	4	(0:0:343)	4	(0:0:521)	4

Instâncias		BRKGA		BRKGA+IV	
Nome	Solução Ótima	T. Médio	Solução Encontrada	T. Médio	Solução Encontrada
GS_30_2	4	(0:0:284)	4	(0:0:381)	4
GS_30_3	4	(0:0:219)	4	(0:0:362)	4
GS_30_4	4	(0:0:271)	4	(0:0:405)	4
<b>GS_30_5</b>	<b>4</b>	<b>(0:0:875)</b>	<b>7</b>	<b>(0:0:379)</b>	<b>4</b>
GS_30_6	4	(0:0:298)	4	(0:0:361)	4
GS_30_7	4	(0:0:378)	4	(0:0:368)	4
GS_30_8	4	(0:0:225)	4	(0:0:361)	4
GS_30_9	4	(0:0:434)	4	(0:0:354)	4
GS_50_1	5	(0:0:47)	5	(0:4:639)	5
GS_50_4	6	(0:0:86)	6	(0:3:25)	6
GS_50_7	5	(0:0:51)	5	(0:5:3)	5
GS_50_8	5	(0:0:157)	5	(0:6:684)	5
RS_15_1	3	(0:0:23)	3	(0:0:11)	3
RS_15_3	2	(0:0:8)	2	(0:0:5)	2
RS_15_4	3	(0:0:24)	3	(0:0:78)	3
RS_15_6	3	(0:0:7)	3	(0:0:4)	3
RS_15_8	3	(0:0:4)	3	(0:0:3)	3
RS_15_9	3	(0:0:5)	3	(0:0:3)	3
RS_15_10	3	(0:0:41)	3	(0:0:49)	3
<b>RS_25_1</b>	<b>4</b>	<b>(0:1:101)</b>	-	<b>(0:3:308)</b>	<b>4</b>
RS_25_2	3	(0:0:193)	3	(0:0:179)	3
<b>RS_25_4</b>	<b>4</b>	<b>(0:1:56)</b>	-	<b>(0:0:295)</b>	<b>4</b>
RS_25_5	4	(0:0:136)	4	(0:0:184)	4
RS_25_6	4	(0:0:706)	4	(0:0:227)	4
RS_25_7	4	(0:0:327)	4	(0:0:200)	4
RS_25_8	3	(0:0:28)	3	(0:0:174)	3
RS_25_9	4	(0:0:305)	4	(0:0:206)	4
RS_25_10	4	(0:0:10)	4	(0:0:205)	4
RS_30_1	3	(0:0:17)	3	(0:0:346)	3
RS_30_3	4	(0:0:519)	4	(0:0:451)	4
RS_30_4	4	(0:1:345)	4	(0:0:711)	4
RS_30_5	4	(0:0:16)	4	(0:0:360)	4
RS_30_6	4	(0:1:271)	4	(0:5:816)	4
RS_30_7	3	(0:0:18)	3	(0:4:853)	3
RS_30_8	4	(0:0:14)	4	(0:0:368)	4
<b>RS_30_10</b>	<b>4</b>	<b>(0:1:622)</b>	-	<b>(0:0:381)</b>	<b>4</b>
<b>RS_50_1</b>	<b>5</b>	<b>(0:0:351)</b>	-	<b>(0:7:472)</b>	<b>5</b>
RS_50_3	4	(0:0:528)	4	(0:44:491)	4
RS_50_6	4	(0:0:62)	4	(0:4:86)	4

Instâncias		BRKGA		BRKGA+IV	
Nome	Solução Ótima	T. Médio	Solução Encontrada	T. Médio	Solução Encontrada
<b>RS_50_7</b>	<b>5</b>	<b>(0:5:571)</b>	-	<b>(0:53:947)</b>	<b>5</b>
RS_50_10	4	(0:0:131)	4	(0:4:107)	4
RL_15_1	3	(0:0:111)	3	(0:0:5)	3
RL_15_3	3	(0:0:141)	3	(0:0:38)	3
RL_15_4	2	(0:0:4)	2	(0:0:11)	2
RL_15_5	3	(0:0:2)	3	(0:0:2)	3
RL_15_6	3	(0:0:1)	3	(0:0:2)	3
RL_15_7	2	(0:0:2)	2	(0:0:4)	2
RL_15_8	2	(0:0:2)	2	(0:0:45)	2
RL_15_9	3	(0:0:2)	3	(0:0:6)	3
RL_25_1	3	(0:0:611)	3	(0:0:221)	3
RL_25_2	3	(0:0:7)	3	(0:0:6)	3
RL_25_3	3	(0:0:9)	3	(0:0:7)	3
RL_25_4	3	(0:0:655)	3	(0:4:724)	3
<b>RL_25_5</b>	<b>4</b>	<b>(0:0:724)</b>	-	<b>(0:13:740)</b>	<b>4</b>
RL_25_6	4	(0:0:281)	4	(0:0:180)	4
RL_25_7	3	(0:0:8)	3	(0:0:9)	3
RL_25_8	4	(0:0:553)	4	(0:0:178)	4
RL_25_9	2	(0:0:10)	2	(0:0:11)	2
RL_25_10	3	(0:0:7)	3	(0:0:7)	3
RL_30_1	3	(0:0:10)	3	(0:0:9)	3
RL_30_2	4	(0:0:586)	4	(0:0:588)	4
RL_30_3	3	(0:0:14)	3	(0:0:15)	3
RL_30_4	3	(0:0:10)	3	(0:0:334)	3
RL_30_6	4	(0:1:91)	4	(0:5:679)	4
RL_30_7	4	(0:0:615)	4	(0:0:336)	4
RL_30_8	4	(0:1:7)	4	(0:0:497)	4
RL_30_9	4	(0:1:117)	4	(0:0:440)	4
RL_30_10	4	(0:1:183)	4	(0:0:333)	4
RL_50_1	4	(0:0:39)	4	(0:3:215)	4
RL_50_2	3	(0:0:52)	3	(0:3:848)	3
<b>RL_50_3</b>	<b>4</b>	<b>(0:3:391)</b>	-	<b>(4:17:156)</b>	<b>5</b>
RL_50_4	4	(0:3:413)	4	(1:43:376)	4
RL_50_5	3	(0:0:41)	3	(0:2:460)	3
RL_50_6	3	(0:0:42)	3	(0:2:519)	3
<b>RL_50_7</b>	<b>5</b>	<b>(0:3:390)</b>	-	<b>(2:45:701)</b>	<b>5</b>
<b>RL_50_8</b>	<b>4</b>	<b>(0:0:33)</b>	-	<b>(0:2:631)</b>	<b>4</b>
RL_50_9	4	(0:0:33)	4	(0:2:397)	4
RL_50_10	4	(0:3:407)	4	(0:17:94)	4

Tabela 5 - Resultados para as instâncias C2

Instâncias		BRKGA		BRKGA + IV	
Nome	Solução Ótima	T. Médio	Solução Encontrada	T. Médio	Solução Encontrada
new.GL_15_3.1	4	(0:0:89)	4	(0:0:60)	4
new.GL_15_3.2	4	(0:0:137)	4	(0:0:101)	4
new.GL_15_3.3	4	(0:0:70)	4	(0:0:55)	4
new.GL_15_3.4	4	(0:0:46)	4	(0:0:28)	4
new.GL_15_3.5	4	(0:0:104)	4	(0:0:147)	4
new.GL_15_3.6	4	(0:0:166)	4	(0:0:44)	4
new.GL_15_3.7	4	(0:0:27)	4	(0:0:33)	4
new.GL_15_3.8	4	(0:0:83)	4	(0:0:68)	4
new.GL_15_3.9	4	(0:0:168)	4	(0:0:52)	4
new.GL_15_3.10	4	(0:0:70)	4	(0:0:26)	4
new.GL_15_6.1	3	(0:0:68)	3	(0:0:233)	3
new.GL_15_6.2	3	(0:0:21)	3	(0:1:61)	3
new.GL_15_6.3	3	(0:0:99)	3	(0:0:54)	3
new.GL_15_6.4	3	(0:0:84)	3	(0:0:545)	3
new.GL_15_6.5	3	(0:0:139)	3	(0:1:154)	3
new.GL_15_6.6	3	(0:0:62)	3	(0:0:144)	3
new.GL_15_6.7	3	(0:0:88)	3	(0:0:345)	3
new.GL_15_6.8	3	(0:0:83)	3	(0:0:206)	3
new.GL_15_6.9	3	(0:0:71)	3	(0:0:73)	3
new.GL_15_6.10	3	(0:0:27)	3	(0:0:25)	3
new.GL_25_9.1	4	(0:0:671)	4	(0:1:439)	4
new.GL_25_9.2	4	(0:0:633)	4	(0:0:832)	4
new.GL_25_9.3	4	(0:0:606)	4	(0:0:724)	4
new.GL_25_9.4	4	(0:0:617)	4	(0:1:25)	4
new.GL_25_9.5	4	(0:0:703)	4	(0:1:479)	4
new.GL_25_9.6	4	(0:0:178)	4	(0:0:130)	4
new.GL_25_9.7	4	(0:0:592)	4	(0:0:251)	4
new.GL_25_9.8	4	(0:0:603)	4	(0:0:280)	4
new.GL_25_9.9	4	(0:0:490)	4	(0:0:210)	4
new.GL_25_9.10	4	(0:0:402)	4	(0:0:169)	4
new.GL_25_10.1	5	(0:0:565)	5	(0:4:257)	5
new.GL_25_10.2	5	(0:0:564)	5	(0:0:664)	5
new.GL_25_10.3	5	(0:0:489)	5	(0:0:372)	5
new.GL_25_10.4	5	(0:0:560)	5	(0:0:151)	5
new.GL_25_10.5	5	(0:0:608)	5	(0:0:422)	5
new.GL_25_10.6	5	(0:0:586)	5	(0:0:191)	5
<b>new.GL_25_10.7</b>	<b>5</b>	<b>(0:0:594)</b>	-	<b>(0:1:338)</b>	<b>5</b>

Instâncias		BRKGA		BRKGA + IV	
Nome	Solução Ótima	T. Médio	Solução Encontrada	T. Médio	Solução Encontrada
<b>new.GL_25_10.8</b>	<b>5</b>	<b>(0:0:657)</b>	-	<b>(0:0:690)</b>	<b>5</b>
new.GL_25_10.9	5	(0:0:107)	5	(0:0:341)	5
new.GL_25_10.10	5	(0:0:100)	5	(0:0:718)	5
new.GL_30_10.1	5	(0:0:865)	5	(0:3:812)	5
new.GL_30_10.2	5	(0:0:833)	5	(0:0:944)	5
new.GL_30_10.3	5	(0:0:746)	5	(0:1:924)	5
<b>new.GL_30_10.4</b>	<b>5</b>	<b>(0:0:946)</b>	-	<b>(0:0:422)</b>	<b>5</b>
<b>new.GL_30_10.5</b>	<b>5</b>	<b>(0:0:968)</b>	-	<b>(0:0:901)</b>	<b>5</b>
new.GL_30_10.6	5	(0:0:855)	5	(0:0:441)	5
new.GL_30_10.7	5	(0:0:852)	5	(0:3:8)	5
new.GL_30_10.8	5	(0:0:783)	5	(0:0:569)	5
new.GL_30_10.9	5	(0:0:834)	5	(0:2:407)	5
<b>new.GL_30_10.10</b>	<b>5</b>	<b>(0:0:841)</b>	-	<b>(0:0:754)</b>	<b>5</b>
<b>new.GL_50_6.1</b>	<b>6</b>	<b>(0:2:640)</b>	-	<b>(1:9:228)</b>	-
<b>new.GL_50_6.2</b>	<b>6</b>	<b>(0:3:955)</b>	-	<b>(1:5:796)</b>	<b>6</b>
<b>new.GL_50_6.3</b>	<b>6</b>	<b>(0:2:918)</b>	-	<b>(1:3:832)</b>	<b>6</b>
<b>new.GL_50_6.4</b>	<b>6</b>	<b>(0:2:984)</b>	-	<b>(0:50:571)</b>	<b>6</b>
new.GL_50_6.5	6	(0:4:14)	6	(0:55:484)	6
<b>new.GL_50_6.6</b>	<b>6</b>	<b>(0:3:692)</b>	-	<b>(0:21:899)</b>	<b>6</b>
<b>new.GL_50_6.7</b>	<b>6</b>	<b>(0:2:512)</b>	-	<b>(0:52:278)</b>	<b>6</b>
<b>new.GL_50_6.8</b>	<b>6</b>	<b>(0:3:350)</b>	-	<b>(0:43:974)</b>	<b>6</b>
<b>new.GL_50_6.9</b>	<b>6</b>	<b>(0:3:351)</b>	-	<b>(0:25:86)</b>	<b>6</b>
new.GL_50_6.10	6	(0:3:126)	6	(0:16:635)	6
new.GL_50_10.1	5	(0:3:864)	6	(1:32:216)	6
<b>new.GL_50_10.2</b>	<b>5</b>	<b>(0:3:768)</b>	-	<b>(1:29:230)</b>	<b>6</b>
<b>new.GL_50_10.3</b>	<b>6</b>	<b>(0:3:768)</b>	-	<b>(1:24:859)</b>	-
new.GL_50_10.4	5	(0:3:780)	5	(1:22:225)	5
new.GL_50_10.5	5	(0:2:479)	5	(1:18:921)	5
<b>new.GL_50_10.6</b>	<b>5</b>	<b>(0:3:821)</b>	-	<b>(1:30:901)</b>	<b>6</b>
<b>new.GL_50_10.7</b>	<b>6</b>	<b>(0:3:970)</b>	-	<b>(1:11:699)</b>	<b>6</b>
new.GL_50_10.8	5	(0:3:854)	6	(1:32:357)	6
new.GL_50_10.9	6	(0:2:482)	6	(0:55:638)	6
<b>new.GL_50_10.10</b>	<b>6</b>	<b>(0:3:778)</b>	-	<b>(1:27:769)</b>	-
new.GH_25_5.1	4	(0:0:867)	4	(0:0:513)	4
new.GH_25_5.2	4	(0:0:505)	4	(0:0:256)	4
new.GH_25_5.3	4	(0:0:727)	4	(0:3:0)	4
new.GH_25_5.4	4	(0:0:719)	4	(0:0:308)	4
new.GH_25_5.5	4	(0:0:765)	4	(0:0:307)	4
new.GH_25_5.6	4	(0:0:681)	4	(0:0:146)	4

Instâncias		BRKGA		BRKGA + IV	
Nome	Solução Ótima	T. Médio	Solução Encontrada	T. Médio	Solução Encontrada
<b>new.GH_25_5.7</b>	<b>4</b>	<b>(0:0:772)</b>	<b>5</b>	<b>(0:1:515)</b>	<b>4</b>
new.GH_25_5.8	4	(0:0:671)	4	(0:0:688)	4
new.GH_25_5.9	4	(0:0:709)	4	(0:3:638)	4
new.GH_25_5.10	4	(0:0:760)	5	(0:1:803)	4
new.GH_30_8.1	4	(0:1:173)	5	(0:4:22)	4
new.GH_30_8.2	4	(0:0:114)	4	(0:0:354)	4
<b>new.GH_30_8.3</b>	<b>4</b>	<b>(0:0:536)</b>	-	<b>(0:2:127)</b>	<b>4</b>
new.GH_30_8.4	4	(0:1:800)	4	(0:8:255)	4
new.GH_30_8.5	4	(0:1:140)	4	(0:3:797)	4
new.GH_30_8.6	4	(0:1:328)	4	(0:0:455)	4
new.GH_30_8.7	4	(0:1:1)	4	(0:0:597)	4
<b>new.GH_30_8.8</b>	<b>4</b>	<b>(0:1:301)</b>	-	<b>(0:3:908)</b>	<b>4</b>
<b>new.GH_30_8.9</b>	<b>4</b>	<b>(0:1:257)</b>	-	<b>(0:0:926)</b>	<b>4</b>
new.GH_30_8.10	4	(0:0:943)	4	(0:0:680)	4
<b>new.GH_50_2.1</b>	<b>6</b>	<b>(0:2:880)</b>	<b>7</b>	<b>(0:42:604)</b>	<b>6</b>
<b>new.GH_50_2.2</b>	<b>6</b>	<b>(0:2:656)</b>	<b>7</b>	<b>(0:33:537)</b>	<b>6</b>
<b>new.GH_50_2.3</b>	<b>6</b>	<b>(0:2:548)</b>	<b>7</b>	<b>(0:6:182)</b>	<b>6</b>
<b>new.GH_50_2.4</b>	<b>6</b>	<b>(0:2:524)</b>	-	<b>(0:6:664)</b>	<b>6</b>
<b>new.GH_50_2.5</b>	<b>6</b>	<b>(0:2:540)</b>	<b>7</b>	<b>(0:20:488)</b>	<b>6</b>
<b>new.GH_50_2.6</b>	<b>6</b>	<b>(0:2:559)</b>	-	<b>(0:5:334)</b>	<b>6</b>
<b>new.GH_50_2.7</b>	<b>6</b>	<b>(0:2:540)</b>	-	<b>(0:29:387)</b>	<b>6</b>
<b>new.GH_50_2.8</b>	<b>6</b>	<b>(0:2:578)</b>	<b>7</b>	<b>(0:13:974)</b>	<b>6</b>
<b>new.GH_50_2.9</b>	<b>6</b>	<b>(0:2:550)</b>	-	<b>(0:6:435)</b>	<b>6</b>
<b>new.GH_50_2.10</b>	<b>6</b>	<b>(0:2:415)</b>	<b>7</b>	<b>(0:26:16)</b>	<b>6</b>
<b>new.GH_50_3.1</b>	<b>5</b>	<b>(0:3:214)</b>	<b>6</b>	<b>(0:44:123)</b>	<b>5</b>
<b>new.GH_50_3.2</b>	<b>5</b>	<b>(0:3:596)</b>	-	<b>(0:51:336)</b>	<b>5</b>
new.GH_50_3.3	5	(0:0:282)	5	(0:15:504)	5
new.GH_50_3.4	5	(0:3:866)	6	(1:28:655)	6
<b>new.GH_50_3.5</b>	<b>5</b>	<b>(0:4:580)</b>	-	<b>(0:48:610)</b>	<b>5</b>
new.GH_50_3.6	5	(0:0:616)	5	(0:44:834)	5
<b>new.GH_50_3.7</b>	<b>5</b>	<b>(0:3:466)</b>	-	<b>(0:42:543)</b>	<b>5</b>
<b>new.GH_50_3.8</b>	<b>5</b>	<b>(0:3:423)</b>	-	<b>(0:7:944)</b>	<b>5</b>
<b>new.GH_50_3.9</b>	<b>5</b>	<b>(0:3:423)</b>	-	<b>(0:8:243)</b>	<b>5</b>
<b>new.GH_50_3.10</b>	<b>6</b>	<b>(0:3:365)</b>	<b>7</b>	<b>(0:38:465)</b>	<b>6</b>
<b>new.GH_50_8.1</b>	<b>6</b>	<b>(0:4:714)</b>	<b>7</b>	<b>(0:42:970)</b>	<b>6</b>
<b>new.GH_50_8.2</b>	<b>6</b>	<b>(0:4:52)</b>	-	<b>(0:30:112)</b>	<b>6</b>
<b>new.GH_50_8.3</b>	<b>6</b>	<b>(0:4:36)</b>	-	<b>(0:7:366)</b>	<b>6</b>
<b>new.GH_50_8.4</b>	<b>6</b>	<b>(0:4:718)</b>	-	<b>(0:7:662)</b>	<b>6</b>
<b>new.GH_50_8.5</b>	<b>6</b>	<b>(0:3:845)</b>	-	<b>(0:53:299)</b>	<b>6</b>

Instâncias		BRKGA		BRKGA + IV	
Nome	Solução Ótima	T. Médio	Solução Encontrada	T. Médio	Solução Encontrada
<b>new.GH_50_8.6</b>	<b>6</b>	<b>(0:4:434)</b>	<b>7</b>	<b>(0:26:377)</b>	<b>6</b>
new.GH_50_8.7	6	(0:3:786)	6	(0:54:498)	6
new.GH_50_8.8	6	(0:3:332)	6	(0:46:481)	6
<b>new.GH_50_8.9</b>	<b>6</b>	<b>(0:3:243)</b>	-	<b>(1:6:957)</b>	<b>6</b>
<b>new.GH_50_8.10</b>	<b>6</b>	<b>(0:3:321)</b>	<b>7</b>	<b>(0:49:565)</b>	<b>6</b>
<b>new.GH_50_9.1</b>	<b>5</b>	<b>(0:3:440)</b>	<b>6</b>	<b>(0:6:866)</b>	<b>5</b>
<b>new.GH_50_9.2</b>	<b>5</b>	<b>(0:3:703)</b>	-	<b>(0:20:901)</b>	<b>5</b>
<b>new.GH_50_9.3</b>	<b>5</b>	<b>(0:3:549)</b>	<b>6</b>	<b>(0:11:971)</b>	<b>5</b>
new.GH_50_9.4	5	(0:3:543)	5	(0:47:851)	5
<b>new.GH_50_9.5</b>	<b>5</b>	<b>(0:3:541)</b>	-	<b>(0:52:273)</b>	<b>5</b>
<b>new.GH_50_9.6</b>	<b>5</b>	<b>(0:3:678)</b>	-	<b>(0:50:825)</b>	<b>5</b>
<b>new.GH_50_9.7</b>	<b>5</b>	<b>(0:3:576)</b>	<b>6</b>	<b>(1:3:127)</b>	<b>5</b>
new.GH_50_9.8	5	(0:0:926)	5	(0:56:113)	5
<b>new.GH_50_9.9</b>	<b>5</b>	<b>(0:3:813)</b>	-	<b>(0:15:13)</b>	<b>5</b>
new.GH_50_9.10	5	(0:0:274)	5	(0:15:140)	5
<b>new.GH_50_10.1</b>	<b>5</b>	<b>(0:3:93)</b>	<b>6</b>	<b>(0:20:239)</b>	<b>5</b>
new.GH_50_10.2	5	(0:3:161)	5	(1:16:905)	5
new.GH_50_10.3	5	(0:3:92)	5	(0:21:962)	5
<b>new.GH_50_10.4</b>	<b>5</b>	<b>(0:3:37)</b>	-	<b>(0:29:673)</b>	<b>5</b>
<b>new.GH_50_10.5</b>	<b>5</b>	<b>(0:3:220)</b>	<b>6</b>	<b>(0:42:607)</b>	<b>5</b>
<b>new.GH_50_10.6</b>	<b>5</b>	<b>(0:3:43)</b>	-	<b>(0:46:616)</b>	<b>5</b>
<b>new.GH_50_10.7</b>	<b>5</b>	<b>(0:3:123)</b>	<b>6</b>	<b>(1:16:608)</b>	<b>5</b>
new.GH_50_10.8	5	(0:3:52)	5	(1:10:573)	5
new.GH_50_10.9	5	(0:3:259)	5	(0:48:126)	5
new.GH_50_10.10	5	(0:3:205)	5	(0:27:222)	5
new.RL_15_2.1	3	(0:0:68)	3	(0:0:78)	3
new.RL_15_2.2	4	(0:0:272)	4	(0:0:268)	4
new.RL_15_2.3	3	(0:0:135)	3	(0:0:85)	3
new.RL_15_2.4	3	(0:0:52)	3	(0:0:07)	3
new.RL_15_2.5	3	(0:0:241)	3	(0:0:88)	3
new.RL_15_2.6	3	(0:0:58)	3	(0:0:101)	3
new.RL_15_2.7	4	(0:0:81)	4	(0:0:75)	4
new.RL_15_2.8	3	(0:0:44)	3	(0:0:52)	3
new.RL_15_2.9	4	(0:0:07)	4	(0:0:275)	4
new.RL_15_2.10	3	(0:0:67)	3	(0:0:126)	3
new.RL_15_5.1	3	(0:0:56)	3	(0:0:51)	3
new.RL_15_5.2	3	(0:0:68)	3	(0:0:93)	3
new.RL_15_5.3	3	(0:0:54)	3	(0:0:96)	3
new.RL_15_5.4	3	(0:0:19)	3	(0:0:64)	3

Instâncias		BRKGA		BRKGA + IV	
Nome	Solução Ótima	T. Médio	Solução Encontrada	T. Médio	Solução Encontrada
new.RL_15_5.5	3	(0:0:57)	3	(0:0:78)	3
new.RL_15_5.6	3	(0:0:139)	3	(0:0:89)	3
new.RL_15_5.7	3	(0:0:49)	3	(0:0:78)	3
new.RL_15_5.8	3	(0:0:55)	3	(0:0:67)	3
new.RL_15_5.9	3	(0:0:236)	3	(0:0:69)	3
new.RL_15_5.10	3	(0:0:45)	3	(0:0:71)	3
new.RL_25_3.1	4	(0:0:100)	4	(0:0:391)	4
new.RL_25_3.2	4	(0:0:106)	4	(0:0:348)	4
new.RL_25_3.3	4	(0:0:109)	4	(0:0:470)	4
new.RL_25_3.4	4	(0:0:95)	4	(0:0:249)	4
new.RL_25_3.5	4	(0:0:203)	4	(0:0:278)	4
new.RL_25_3.6	4	(0:0:213)	4	(0:0:391)	4
new.RL_25_3.7	4	(0:0:105)	4	(0:0:808)	4
new.RL_25_3.8	4	(0:0:105)	4	(0:0:337)	4
new.RL_25_3.9	4	(0:0:97)	4	(0:0:245)	4
new.RL_25_3.10	4	(0:0:432)	4	(0:0:345)	4
<b>new.RL_30_2.1</b>	<b>4</b>	<b>(0:0:521)</b>	-	<b>(0:0:719)</b>	<b>4</b>
new.RL_30_2.2	4	(0:0:363)	4	(0:1:154)	4
<b>new.RL_30_2.3</b>	<b>4</b>	<b>(0:0:540)</b>	-	<b>(0:2:614)</b>	<b>4</b>
new.RL_30_2.4	4	(0:0:469)	4	(0:1:049)	4
new.RL_30_2.5	4	(0:0:332)	4	(0:0:686)	4
new.RL_30_2.6	4	(0:0:222)	4	(0:0:660)	4
<b>new.RL_30_2.7</b>	<b>4</b>	<b>(0:0:982)</b>	<b>5</b>	<b>(0:8:944)</b>	<b>4</b>
<b>new.RL_30_2.8</b>	<b>4</b>	<b>(0:1:3)</b>	-	<b>(0:5:938)</b>	<b>4</b>
new.RL_30_2.9	4	(0:0:944)	4	(0:2:310)	4
new.RL_30_2.10	4	(0:0:923)	4	(0:3:707)	4
new.RL_30_9.1	4	(0:1:23)	4	(0:3:63)	4
<b>new.RL_30_9.2</b>	<b>4</b>	<b>(0:1:12)</b>	<b>5</b>	<b>(0:4:115)</b>	<b>4</b>
<b>new.RL_30_9.3</b>	<b>4</b>	<b>(0:1:18)</b>	-	<b>(0:4:102)</b>	<b>4</b>
new.RL_30_9.4	4	(0:1:17)	4	(0:8:223)	4
new.RL_30_9.5	4	(0:1:70)	4	(0:4:772)	4
new.RL_30_9.6	4	(0:1:102)	4	(0:8:760)	4
<b>new.RL_30_9.7</b>	<b>4</b>	<b>(0:1:5)</b>	<b>5</b>	<b>(0:4:790)</b>	<b>4</b>
<b>new.RL_30_9.8</b>	<b>4</b>	<b>(0:1:10)</b>	-	<b>(0:5:124)</b>	<b>4</b>
<b>new.RL_30_9.9</b>	<b>4</b>	<b>(0:1:25)</b>	<b>5</b>	<b>(0:5:795)</b>	<b>4</b>
<b>new.RL_30_9.10</b>	<b>4</b>	<b>(0:1:30)</b>	-	<b>(0:11:169)</b>	<b>4</b>
<b>new.RL_50_2.1</b>	<b>5</b>	<b>(0:2:929)</b>	-	<b>(1:23:692)</b>	<b>5</b>
<b>new.RL_50_2.2</b>	<b>6</b>	<b>(0:2:965)</b>	<b>7</b>	<b>(0:51:179)</b>	<b>6</b>
new.RL_50_2.3	5	(0:2:866)	5	(1:34:253)	5

Instâncias		BRKGA		BRKGA + IV	
Nome	Solução Ótima	T. Médio	Solução Encontrada	T. Médio	Solução Encontrada
<b>new.RL_50_2.4</b>	<b>5</b>	<b>(0:2:891)</b>	<b>6</b>	<b>(1:51:308)</b>	<b>5</b>
<b>new.RL_50_2.5</b>	<b>5</b>	<b>(0:2:946)</b>	-	<b>(1:30:762)</b>	<b>5</b>
new.RL_50_2.6	5	(0:2:992)	5	(1:24:960)	5
<b>new.RL_50_2.7</b>	<b>5</b>	<b>(0:3:527)</b>	-	<b>(1:15:960)</b>	<b>5</b>
<b>new.RL_50_2.8</b>	<b>5</b>	<b>(0:3:147)</b>	-	<b>(1:17:982)</b>	<b>5</b>
<b>new.RL_50_2.9</b>	<b>5</b>	<b>(0:3:50)</b>	-	<b>(1:13:323)</b>	<b>5</b>
<b>new.RL_50_2.10</b>	<b>5</b>	<b>(0:3:547)</b>	<b>6</b>	<b>(1:28:828)</b>	<b>5</b>
<b>new.RL_50_8.1</b>	<b>5</b>	<b>(0:3:70)</b>	-	<b>(1:14:360)</b>	<b>5</b>
<b>new.RL_50_8.2</b>	<b>6</b>	<b>(0:3:111)</b>	<b>7</b>	<b>(1:8:926)</b>	<b>6</b>
<b>new.RL_50_8.3</b>	<b>5</b>	<b>(0:3:180)</b>	-	<b>(1:18:947)</b>	<b>5</b>
<b>new.RL_50_8.4</b>	<b>5</b>	<b>(0:3:316)</b>	-	<b>(1:30:412)</b>	<b>5</b>
<b>new.RL_50_8.5</b>	<b>6</b>	<b>(0:3:637)</b>	-	<b>(1:25:826)</b>	<b>5</b>
<b>new.RL_50_8.6</b>	<b>5</b>	<b>(0:3:687)</b>	-	<b>(1:30:146)</b>	<b>5</b>
<b>new.RL_50_8.7</b>	<b>5</b>	<b>(0:3:694)</b>	-	<b>(1:40:940)</b>	<b>5</b>
<b>new.RL_50_8.8</b>	<b>5</b>	<b>(0:3:692)</b>	-	<b>(1:30:708)</b>	<b>5</b>
<b>new.RL_50_8.9</b>	<b>5</b>	<b>(0:3:682)</b>	-	<b>(1:32:738)</b>	<b>5</b>
<b>new.RL_50_8.10</b>	<b>5</b>	<b>(0:3:657)</b>	-	<b>(1:29:708)</b>	<b>5</b>
new.RH_15_10.1	3	(0:0:388)	3	(0:0:65)	3
new.RH_15_10.2	3	(0:0:54)	3	(0:0:5)	3
new.RH_15_10.3	3	(0:0:119)	3	(0:0:81)	3
new.RH_15_10.4	3	(0:0:72)	3	(0:0:148)	3
new.RH_15_10.5	3	(0:0:131)	3	(0:0:83)	3
new.RH_15_10.6	3	(0:0:65)	3	(0:0:87)	3
new.RH_15_10.7	3	(0:0:58)	3	(0:0:69)	3
new.RH_15_10.8	3	(0:0:47)	3	(0:0:77)	3
new.RH_15_10.9	3	(0:0:6)	3	(0:0:106)	3
new.RH_15_10.10	3	(0:0:103)	3	(0:0:65)	3
new.RH_30_5.1	4	(0:0:137)	4	(0:0:459)	4
new.RH_30_5.2	4	(0:0:129)	4	(0:0:422)	4
new.RH_30_5.3	4	(0:0:156)	4	(0:0:943)	4
new.RH_30_5.4	4	(0:0:29)	4	(0:0:914)	4
<b>new.RH_30_5.5</b>	<b>4</b>	<b>(0:0:533)</b>	-	<b>(0:0:922)</b>	<b>4</b>
new.RH_30_5.6	4	(0:0:54)	4	(0:1:797)	4
new.RH_30_5.7	4	(0:0:477)	4	(0:1:528)	4
<b>new.RH_30_5.8</b>	<b>4</b>	<b>(0:0:493)</b>	-	<b>(0:1:893)</b>	<b>4</b>
<b>new.RH_30_5.9</b>	<b>4</b>	<b>(0:0:471)</b>	-	<b>(0:1:259)</b>	<b>4</b>
new.RH_30_5.10	4	(0:0:498)	4	(0:1:856)	4

Tabela 6 - Resultados para as instâncias C3

Instâncias		BRKGA			BRKGA+IV		
Nome	*	T. Médio	(*)	C(%)	T. Médio	(*)	C(%)
<b>LSHK_25_35_02</b>	<b>6</b>	<b>(0:0:529)</b>	<b>-</b>	<b>0</b>	<b>(0:9:44)</b>	<b>6</b>	<b>20</b>
LSHK_25_35_08	6	(0:0:451)	6	40	(0:0:218)	6	100