

Pontifícia Universidade Católica de Goiás
Programa de Mestrado em Engenharia de Produção e Sistemas

**ALGORITMO GENÉTICO APLICADO AO PLANEJAMENTO DE
REDES DE TELECOMUNICAÇÕES**

Emerson de Souza Campos

2017

ALGORITMO GENÉTICO APLICADO AO PLANEJAMENTO DE REDES DE TELECOMUNICAÇÕES

Emerson de Souza Campos

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Produção e Sistemas da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para obtenção do título de Mestre em Engenharia de Produção e Sistemas.

Orientadora: Prof.^a Maria José Pereira Dantas, Dr.^a.

MARÇO

2017

C198a Campos, Emerson de Souza
Algoritmo genético aplicado ao planejamento de redes
de telecomunicação [manuscrito]/ Emerson de Souza
Campos - 2017.
145 f.; il. 30 cm

Texto em português com resumo em inglês
Dissertação (mestrado) -- Pontifícia Universidade
Católica de Goiânia, Programa de Pós-Graduação Stricto
Sensu em Mestrado em Engenharia de Produção e Sistemas,
Goiânia, 2017
Inclui referências f. 77-80

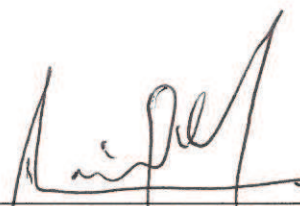
1. Telecomunicações. 2. Interconexão em rede (Telecomunicações).
Algoritmos de computador. 4. Otimização combinatória.
I.Dantas, Maria José Pereira. II. Título.

CDU: 621.39(043)

ALGORITMO GENÉTICO APLICADO AO PLANEJAMENTO DE REDES DE TELECOMUNICAÇÕES

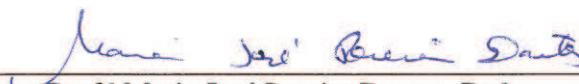
EMERSON DE SOUZA CAMPOS

Esta Dissertação julgada adequada para obtenção do título de Mestre em Engenharia de Produção e Sistemas, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Produção e Sistemas da Pontifícia Universidade Católica de Goiás em 29 de março de 2017.

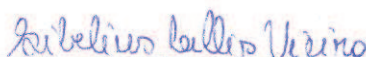


Prof. Ricardo Luiz Machado, Dr.
Coordenador do Programa de Pós-Graduação
em Engenharia de Produção e Sistemas


Banca Examinadora:



Prof.ª Maria José Pereira Dantas, Dr.ª
Orientadora



Prof. Sibelius Lellis Vieira, Dr.
Membro interno



Prof. Rafael Timóteo de Sousa Junior, Dr.
Membro externo

GOIÂNIA – GOIÁS
MARÇO DE 2017

DEDICATÓRIA

Dedico à minha família e a todos os professores que de alguma forma contribuíram para a realização deste trabalho, e em especial, à minha esposa Rita de Cássia que esteve sempre ao meu lado em todos os momentos.

Aos meus avós Ernestina, Durval, Ariosto e Rosa que estão em outra esfera, mas que estiveram ao meu lado em momentos difíceis. Ao meu pai e minha mãe, que sempre me apoiaram incansavelmente.

Aos meus filhos Nicolás e Ieda que tiveram paciência em suportar minha ausência.

Ao meu grande amigo Virgílio que sempre demonstrou companheirismo e uma grande amizade.

Emerson de Souza Campos

AGRADECIMENTOS

A Deus todo poderoso que és minha força e sempre iluminou meus caminhos.

Aos meus queridos pais João e Maria Helena que são os pilares da minha existência e os maiores exemplos da minha vida, responsáveis por meu caráter, minha formação e todo o sucesso que obtive até então.

Às minhas irmãs Amandha e Débora que são mulheres sábias que sempre me receberam de braços abertos em todos os momentos e que de certa forma, sempre me compreenderam e apoiaram-me ao longo de minha vida.

A minha magnífica esposa Rita de Cássia e meus dois filhos Nícolas e Ieda pelo amor, carinho e compreensão em todos os momentos, por estarem sempre presentes e vibrarem com cada uma de minhas vitórias e acreditar plenamente em minhas palavras.

À Maria Antonieta e Jayme que acima de tudo são grandes amigos e que nunca mediram esforços para estar ao meu lado e de minha família, sempre me apoiando com palavras doces e sinceras.

Aos meus amigos Victor Honorato Cunha e Lucas Carvalho Guimarães que não mediram esforços para a realização deste trabalho.

À Élvia e a Regina com as quais em um curto período da minha vida aprendi muito e tenho como exemplo de vida e superação.

Ao Celso com quem pude aprender com cada conversa e sei que levarei estes ensinamentos por toda a minha vida.

À minha grande amiga Delluane que sempre me apoiou com palavras de fé e de otimismo.

À EMBRATEL pela atenção e disponibilidade em especial ao Gleisson Nascimento.

Ao meu grande amigo e professor Dr. Marcos Antônio de Sousa que sempre contribuiu para o meu crescimento e conquistas.

A querida e estimada professora Dra. Maria José que acima de tudo mostrou vontade e paciência em transformar seu rico conhecimento em sábias palavras e que também foi fundamental para que esta conquista se tornasse inesquecível.

À FAPEG - Fundação de Amparo à Pesquisa do Estado de Goiás pelo apoio financeiro e oportunidades imensas dentro do campo da pesquisa.

À Faculdade de tecnologia SENAI Ítalo Bologna pela compreensão e apoio e em especial à diretora Misclay Marjorie Correia da Silva que foi uma amiga nas horas mais difíceis.

A todos meus amigos e professores da Faculdade de Tecnologia SENAI Ítalo Bologna pelo apoio incondicional.

Emerson de Souza Campos

EPÍGRAFE

“Um raciocínio lógico leva você de A a B. A imaginação leva você a qualquer lugar que você quiser”.

Albert Einstein

RESUMO da Dissertação apresentada ao MEPROS/ PUC Goiás como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia de Produção e Sistemas (M.Sc.)

ALGORITMO GENÉTICO APLICADO AO PLANEJAMENTO DE REDES DE TELECOMUNICAÇÕES

Autor: Emerson de Souza Campos

Março de 2017

Orientadora: Maria José Pereira Dantas, Dr^a.

RESUMO

Os sistemas de telecomunicações estão em constante desenvolvimento e a demanda crescente de usuários e novos serviços possibilitaram o surgimento de novas tecnologias. O planejamento tornou-se indispensável devido à competitividade e a grande quantidade de recursos financeiros envolvidos. Este trabalho visa propor e avaliar um algoritmo genético de otimização para o planejamento de redes de telecomunicações. Por se tratar de um problema combinatorial o objetivo é avaliar as vantagens e desvantagens do modelo com base no algoritmo genético. Os grafos que representam as redes foram codificados em matrizes de incidência e os operadores genéticos de cruzamento e mutação foram projetados para atuarem sobre matrizes. O software MATLAB[®] foi utilizado como ferramenta computacional para implementação dos algoritmos. O modelo proposto minimiza o custo, considerando as restrições de demanda e capacidade técnica. Os resultados encontrados são comparados com os resultados publicados na biblioteca de instâncias de rede SNDlib. A avaliação da primeira versão do algoritmo foi feita com base em uma instância PDH (*Plesiochronous Digital Hierarchy*), de pequeno porte. O ganho obtido no custo da rede, em relação à solução apresentada na biblioteca usando programação linear com abordagem arco-caminho, é de 15,15%. Na segunda etapa aplicou-se o algoritmo para otimização de uma rede SDH (*Synchronous Digital Hierarchy*), de maior porte. Identificou-se a necessidade de hibridizar o algoritmo inicial com um algoritmo de pós-otimização. Os resultados encontrados são promissores porque se aproximam de soluções similares em um tempo de execução substancialmente menor que o tempo de referência da SNDlib. Novas pesquisas devem ser feitas para que o algoritmo proposto possa dar boas respostas para redes de grande porte em função de ser esta a realidade desta área de pesquisa.

PALAVRAS-CHAVE: Redes de Telecomunicações; Otimização combinatorial, Algoritmos Evolutivos, Biblioteca SNDlib.

ABSTRACT of Dissertation presented to the MEPROS/ PUC Goiás as part of the requirements required for obtaining a Master's degree in Engineering of Production and Systems (M.Sc.)

GENETIC ALGORITHM APPLIED TO THE PLANNING OF TELECOMMUNICATIONS NETWORKS

Author: Emerson de Souza Campos

March, 2017.

Master's advisor: Maria José Pereira Dantas, Dr^a.

ABSTRACT

Telecommunication systems are in constant development and the increasing demand of users and new services have enabled the emergence of new technologies. Planning has become indispensable due to the competitiveness and the large amount of financial resources involved. This work aims to propose and evaluate a genetic optimization algorithm for the planning of telecommunications networks. Because it is a combinatorial problem, the objective is to evaluate the advantages and disadvantages of the model based on the genetic algorithm. The graphs representing the networks were encoded in incidence matrices and the genetic operators of crossing and mutation were designed to act on matrices. MATLAB® software was used as a computational tool to implement the algorithms. The proposed model minimizes cost, considering the constraints of demand and technical capacity. The results found are compared to the published results in the SNDlib network instance library. The evaluation of the first version of the algorithm was based on a small PDH (Plesiochronous Digital Hierarchy) instance. The gain obtained in the cost of this network, compared to the solution presented in the library using linear programming with an arc-path approach, is 15.15%. In the second step, the algorithm for the optimization of a larger SDH (Synchronous Digital Hierarchy) network was applied. In this case, the need to hybridize the initial algorithm with a post-optimization algorithm was identified. The results obtained for the larger network were close to that of the SNDlib network library, although they were not better. The results found are promising because they approach similar solutions at a substantially shorter execution time than the SNDlib reference time. New research must be done so that the proposed algorithm can give good answers to large networks due to this being the reality of this area of research.

KEYWORDS: Telecommunications networks; Combinatorial Optimization; Evolutionary Algorithms, Library SNDlib.

SUMÁRIO

CAPÍTULO I	14
1. INTRODUÇÃO	14
1.1 CONTEXTUALIZAÇÃO E PROBLEMÁTICA	14
1.2 SOLUÇÕES PARA O PLANEJAMENTO DE REDES	15
1.3 OBJETIVOS	17
1.3.1 Objetivos específicos	17
1.4 JUSTIFICATIVA	17
1.5 ORGANIZAÇÃO DO TEXTO	19
CAPÍTULO II	20
2. FUNDAMENTAÇÃO TEÓRICA	20
2.1. REDES DE TELECOMUNICAÇÕES	20
2.2. PRINCIPAIS TECNOLOGIAS UTILIZADAS EM REDES DE TELECOMUNICAÇÕES	23
2.2.1. Tecnologia PDH	24
2.2.2. Tecnologia SDH / SONET	26
2.2.3. Comparação entre as tecnologias SDH E PDH	27
2.3. OTIMIZAÇÃO LINEAR	28
2.3.1. Otimização combinatória e grafos	29
2.3.2. Formulação nó arco e arco caminho	31
2.3.3. Algoritmos genéticos	33
2.3.4. Problemas intratáveis	35
2.3.5. Codificação	35
2.3.6. Aptidão e Seleção	36
2.3.7. Operador de <i>Crossover</i>	36
2.3.8. Operador de Mutação	38
2.3.9. Elitismo	39
2.4. A BIBLIOTECA DE PROJETOS DE REDES FIXAS DE TELECOMUNICAÇÕES - SNDLIB	39
CAPÍTULO III	42
3. METODOLOGIA	42
3.1. CLASSIFICAÇÃO DA PESQUISA	42
3.2. PROCEDIMENTOS METODOLÓGICOS	43
3.3. SELEÇÃO DE INSTÂNCIA SNDLIB	44
3.4. NETWORK DESIGN GENETIC ALGORITHM (NDGA)	57
3.4.1. Codificação, inactibilidade e população inicial	58
3.4.2. Aptidão e Seleção	60
3.4.3. <i>Crossover</i>	60

3.4.4.	Mutação	61
3.4.5.	Elitismo.....	62
3.5.	ALGORITMO DE PÓS-OTIMIZAÇÃO.....	62
CAPÍTULO IV		65
4.	RESULTADOS E DISCUSSÕES.....	65
4.1.	NDGA APLICADO À REDE PDH.....	65
4.2.	NDGA E ALGORITMO DE PÓS-OTIMIZAÇÃO APLICADO À REDE TA1	70
CAPÍTULO V		75
CONCLUSÃO		75
4.3.	DESDOBRAMENTOS DO TRABALHO	76
4.4.	TRABALHOS FUTUROS.....	76
REFERÊNCIAS:		77
APÊNDICE - A.....		81
APÊNDICE - B.....		94
APÊNDICE C – Código-fonte elaborado no software MATLAB para simulação da instância PDH.....		107
APÊNDICE D – Código-fonte do algoritmo de pós-otimização elaborado no software matlab para simulação da instância ta1 (SDH).....		129
APÊNDICE E.....		134
ANEXO A - Carta de apresentação para visita técnica à EMBRATEL.....		145

ÍNDICE DE ILUSTRAÇÕES

Figura 1 – Contextualização das redes de telecomunicações.....	21
Figura 2 – Exemplo de Rede WAN	22
Figura 3 – Redes ópticas baseadas em malhas.....	22
Figura 4 – Divisão simplificada das Redes de Telecomunicações.....	24
Figura 5 – Multiplexador 4x1 – Conversão Paralelo Série	26
Figura 6 – Exemplo de Grafo não-direcionado	30
Figura 7 – Lista de incidência para o grafo da Figura 6.....	30
Figura 8 – Matriz de adjacência	31
Figura 9 – Grafo não direcionado (arco-caminho).....	32
Figura 10 – Grafo não direcionado	32
Figura 11 – Caminhos possíveis para o grafo direcionado de 04 nós	33
Figura 12 – Representação gráfica dos algoritmos genéticos	34
Figura 13 – Exemplo de pontos de corte.....	37
Figura 14 – Exemplo de <i>crossover</i>	37
Figura 15 – Mutação baseada na troca de posição de genes (<i>i para j</i>).....	38
Figura 16 – Mutação baseada na troca de cromossomos	38
Figura 17 - Processo de Planejamento da biblioteca SNDlib.....	40
Figura 18 – Página da biblioteca de redes SNDlib.....	41
Figura 19 – Página da biblioteca de redes SNDlib.....	41
Figura 20 – Processo de modelagem.....	42
Figura 21 – Fluxograma simplificado da metodologia aplicada	43
Figura 22 – Rede Candidata para a instância PDH – Topologia.....	46
Figura 23 – Topologia de rede ta1 (SDH).....	49
Figura 24 – Fluxograma algoritmo proposto - NDGA.....	58
Figura 25 – Codificação das soluções	59
Figura 26 – Exemplo de população inicial.....	60
Figura 27 – <i>Crossover</i> de <i>n</i> pontos para o NDGA.....	61
Figura 28 – Mutação com a ocorrência de 03 trocas.....	61
Figura 29 – Exemplo de rede de telecomunicações	62
Figura 30 – Fluxograma após a otimização.....	64
Figura 31 – Convergência do algoritmo proposto NDGA	68
Figura 32 – Rede PDH otimizada	69
Figura 33 – Convergência do algoritmo proposto NDGA	72
Figura 34 – Rede ta1 (SDH) otimizada.....	74

ÍNDICE DE TABELAS

Tabela 1 – Níveis de hierarquia PDH, canais e taxas.....	25
Tabela 2 – Diferenças principais entre o Sistema T (<i>T-Carrier</i>) e o Sistema E.....	26
Tabela 3 – Taxa de Multiplexação da SONET e SDH.....	27
Tabela 4 – Principais diferenças entre as tecnologias SDH e PDH	27
Tabela 5 – Tempo de execução associados a funções	35
Tabela 6 – Resultados da rede atlanta--D-B-M-N-C-A-N-N	40
Tabela 7 – Instância da Biblioteca de redes SNDlib (U-U-E-N-S-A-N-N).	44
Tabela 8 – Valores para cada demanda prevista da rede PDH.....	46
Tabela 9 – Dados da rede PDH	47
Tabela 10 – Valores de demandas previstas da rede SDH (D1 a D102).....	50
Tabela 11 – Valores de demandas previstas da rede SDH (D103 a D196).....	51
Tabela 12 – Valores de demandas previstas da rede SDH (D197 a D309).....	52
Tabela 13 – Valores de demandas previstas da rede SDH (D10 a D396).....	53
Tabela 14 – Dados da rede SDH para os módulos de 1 a 4.....	54
Tabela 15 – Dados da rede SDH para os módulos de 5 a 8.....	55
Tabela 16 – Dados da rede SDH para os módulos de 9 a 11.....	56
Tabela 17 – Configuração dos Experimentos.....	65
Tabela 18 – Resultados das simulações e da biblioteca SNDlib.....	67
Tabela 19 – Resultados do experimento 04.....	68
Tabela 20 – Características principais do melhor resultado apresentado pelo algoritmo	69
Tabela 21 – Configuração dos Experimentos.....	70
Tabela 22 – Resultados das simulações e da biblioteca SNDlib para 5 caminhos.....	70
Tabela 23 – Resultados das simulações e da biblioteca SNDlib para 10 caminhos.....	71
Tabela 24 – Resultados do experimento 04.....	72
Tabela 25 – Características principais do melhor resultado apresentado pelo algoritmo	72
Tabela 26 – Caminhos utilizados para escoamento de demandas.....	74
Tabela 27 – Links utilizados para escoamento das demandas para a rede ta1 (SDH)	134

LISTA DE SÍMBOLOS E ABREVIATURAS

ADSL	<i>Asymmetric Digital Subscriber Line</i>
ANSI	American National Standards Institute
ATM	<i>Asynchronous Transfer Mode</i>
DWDM	<i>Dense Wavelength Division Multiplexing</i>
GPON	<i>Gigabit passive optical network</i>
GSM	<i>Global System for Mobile</i>
ITU-T	Internacional Telecommunications Union
IP	<i>Internet Protocol</i>
LTE	<i>Long-Term Evolution</i>
METRO ETHERNET	<i>Ethernet aplicada em áreas metropolitanas</i>
NDGA	<i>Network Design Genetic Algorithm</i>
NG-SDH	<i>Next Generation – SDH (Synchronous Digital Hierarchy)</i>
NP-Completo	Não Polinomial – Completos
OTN	<i>Optical Transport Network</i>
OXC	<i>Optical Cross-Connect</i>
PCM	<i>Pulse Code Modulation</i>
PDH	<i>Plesiochronous Digital Hierarchy</i>
SDH	<i>Synchronous Digital Hierarchy</i>
SNDlib	<i>Survivable fixed telecommunication Network Design</i>
SONET	<i>Synchronous Optical Digital Network</i>
STM	<i>Synchronous Transport Module</i>
ta1	Instância de rede da biblioteca SNDlib
TDM	<i>Time Division Multiplexing</i>
UMTS	<i>Universal Mobile Telecommunication System</i>
WDM	<i>Wavelength-Division Multiplex</i>
xDSL	<i>Digital Subscriber Line (x: significa protocolo genérico)</i>

CAPÍTULO I

1. INTRODUÇÃO

A Internet e sua evolução acelerada tornou-se uma referência para a comunicação que de forma clássica proporcionava serviços tais como telefonia, teledifusão, rádio e serviços de envio de *e-mails*. A demanda crescente de usuários e o surgimento de novos serviços agregaram novas tecnologias e serviços multimídia tais como, jogos on-line, compartilhamento de arquivos, mensagens instantâneas de dados e vídeo e outros (BEN-AMEUR e KERIVIN, 2005).

Com o crescimento das demandas foi necessário o aprimoramento das tecnologias utilizadas, tais como, por exemplo o surgimento da tecnologia PDH (*Plesiochronous Digital Hierarchy*) que possibilitou a transmissão de sinais de altas velocidades com taxas cada vez mais elevadas, com menores níveis de erro.

Devido à diversidade de aplicações e crescimento de demanda está se tornando cada vez mais difícil prever padrões para o tráfego e capturar as variações, gerando riscos e incertezas de mercado. Logo, um planejamento adequado é necessário para a escolha correta das tecnologias a serem utilizadas, de forma a atender as demandas existentes com a melhor relação custo-benefício.

O planejamento de redes consiste em determinar problemas complexos e de difíceis soluções onde sistemas computacionais verificam, testam e analisam inúmeras possibilidades de conexões em relação ao número de nós e a quantidade de caminhos para valores de demanda. Uma das características desses problemas consiste em definir quais os melhores caminhos para escoar as demandas e assim atribuir capacidades aos equipamentos com o objetivo de minimizar custos (BEN-AMEUR e KERIVIN, 2005).

1.1 CONTEXTUALIZAÇÃO E PROBLEMÁTICA

Os problemas de otimização de redes de telecomunicações estão presentes na literatura científica e as mais variadas soluções dependem diretamente de estudos que implementem modelos matemáticos e computacionais e que se constituem de etapas como a concepção ou formulação, implementação e pôr fim a análise dos resultados.

Em uma rede de telecomunicações existem demandas que possuem uma origem e um destino e são conectados por meio físico ou remoto representados através de *links*.

Desta forma, o problema consiste em definir os meios de conexão e suas capacidades para atender todos os usuários com qualidade e menor custo.

Para otimizar as redes de telecomunicações, a literatura tem apresentado uma abordagem determinística que corresponde aos métodos tradicionais. Como exemplo pode-se citar o método simplex, *branch and bound*, dentre outros (BOAVENTURA NETTO, 2011). Tais métodos garantem que a solução encontrada seja ótima de acordo com a formulação matemática aplicada.

O tempo de busca por soluções de problemas está associado à sua complexidade. Problemas intratáveis, como o planejamento de redes tornam as técnicas tradicionais inviáveis. (LINDEN, 2012).

Diante da limitação do tempo de solução e busca por valores ótimos, surgem as técnicas heurísticas que realizam uma busca aleatória direcionada, mas que não garantem soluções ótimas, apenas bons resultados. Para solucionar diversos problemas em sistemas de telecomunicações, novas propostas e novas técnicas são apresentadas na literatura. Atualmente estudos no campo de projetos hierárquicos de sobrevivência de rede são aplicados.

Uma tendência atual, são os algoritmos evolucionários que combinam duas ou mais funções objetivo e estabelece o ordenamento dos indivíduos, do mais apto ao menos apto (ROCHA, *et al.*, 2016). Na área de inteligência computacional, os Algoritmos Genéticos são métodos de otimização global inspirados nos processos genéticos de organismos biológicos (COUËTOUX, *et al.*, 2010).

As soluções para aplicação de métodos heurísticos são controladas por critérios probabilísticos, que nos últimos anos, tiveram grandes avanços devido aos recursos computacionais, uma vez que necessitam de um número elevado de avaliações da função objetivo.

1.2 SOLUÇÕES PARA O PLANEJAMENTO DE REDES

Banov (2005) apresentou um modelo multiperíodo capaz de permitir um conjunto de equipamentos instalados. A expansão da rede de acesso objetivava o atendimento de todas as demandas solicitadas maximizando a receita. Para a resolução de problemas utiliza um método heurístico baseado em Algoritmos Genéticos.

Hillier e Lieberman (2009), descrevem em seu livro a evolução das redes de telecomunicações e apresentam metodologias e técnicas capazes de resolver problemas

de otimização com melhor eficiência de modo a permitir soluções de problemas intratáveis de décadas atrás.

CantoDosSantos e Barreto (2011) propuseram estratégias para o planejamento de redes através da simulação de falhas nos enlaces (*links*) de interligação entre estações e avaliavam o desempenho do algoritmo através da recomposição das redes.

Tripathy, Dash e Tripathy (2012), abordaram a confiabilidade dos componentes dos sistemas ao projetar redes de telecomunicações em que problemas de otimização topológica é definido como a existência de pelo menos um caminho confiável entre um par de nós que satisfazem o custo predefinido da rede.

Estudos apresentados por DeSouza *et al.*, (2013) utilizaram como suporte no processo decisório um Modelo de Programação Linear Inteira Mista – PLIM para o dimensionamento da rede de interconexão entre nós e *links*, em que a demanda crescente de usuários por serviços de telecomunicações influenciava diretamente a capacidade de atendimento na telefonia móvel celular. Uma das principais estratégias de mercado adotadas pelas operadoras tem sido alugar os enlaces de transmissão.

Oliveira (2015) em sua dissertação de mestrado otimizou redes de telecomunicações através de modelos matemáticos baseados em Programação Linear Inteira Mista – PLIM e Lógica *Fuzzy* para auxiliar na tomada de decisão e planejamento estratégico onde as demandas eram imprecisas.

Baumgartner, Reddy e Bauschert (2015), diferentemente dos modelos clássicos para a interconexão de redes móveis celulares, desenvolveram um novo modelo de otimização matemática para encapsulamentos de redes de núcleo móveis virtuais objetivando os limites de latência. A plataforma virtual atribui equipamentos tais como *gateways* de núcleo único.

Rodríguez-Martín, Salazar-González e Yaman (2016) desenvolveram um algoritmo de ramificação e corte e o seu comportamento foi testado em instâncias da biblioteca de redes SNDlib.

Outros projetos de diversas áreas envolvendo a teoria de redes em geral, podem ser utilizados como base para o planejamento de redes de telecomunicações. Como exemplo, o dimensionamento de redes de gasodutos de transmissão de hidrogênio (ANDRÉ, AURAY, *et al.*, 2013) e roteamento de canais em circuitos VLSI (LIENIG e THULASIRAMAN, 1993).

Considerando as limitações dos métodos determinísticos é necessário a utilização de técnicas heurísticas. O planejamento de redes é de alta complexidade, logo com o uso

dos algoritmos genéticos pode-se chegar a boas soluções com menor tempo e esforço computacional.

Por todos estes motivos a pesquisa propõe o desenvolvimento de um algoritmo genético capaz de otimizar custos, visando o planejamento de redes de telecomunicações de forma a atender todas as demandas a ela solicitadas.

1.3 OBJETIVOS

O objetivo geral deste trabalho é de propor e avaliar um algoritmo genético para planejamento de redes de telecomunicações com o objetivo de minimizar custos, obedecendo às restrições impostas devido às características técnicas da rede.

1.3.1 Objetivos específicos

- Propor um algoritmo genético (AG) mono-objetivo para otimizar o custo da rede, com codificação bidimensional e operadores genéticos para atuarem sobre matrizes;
- Avaliar o algoritmo proposto a partir de instâncias da biblioteca SNDlib que utiliza programação linear e abordagem arco-caminho;
- Validar o método proposto para instâncias de rede de maior porte disponíveis na biblioteca SNDlib.

1.4 JUSTIFICATIVA

O crescimento das demandas está relacionado diretamente com o custo da rede. Como mencionado, os métodos com abordagem determinística apresentam sérias limitações de tempo à medida que aumenta a complexidade dos problemas.

Desta forma, o planejamento de redes de telecomunicações insere-se numa gama de problemas complexos a serem solucionados por sistemas computacionais. Isso se explica pela relação entre o número de nós de uma rede com a complexidade computacional do problema, pois na medida em que se aumenta a quantidade de nós também há um aumento da quantidade de variáveis. Essa relação entre número de nós, quantidade de *links* e valores de demanda caracteriza o problema como combinatorial.

Ben-Ameur e Kerivin (2005) introduziram um modelo flexível onde o tráfego em uma rede de telecomunicações pertence a um grafo onde o tráfego de demandas é escoado entre os nós e que podem percorrer muitos caminhos distintos, pertencentes a recursos da rede.

Segundo Ben-Ameur e Kerivin (2005), estabilidade e o roteamento da rede de telecomunicações, através de cada caminho definido com as demandas solicitadas devem ser escoadas para garantir o custo mínimo. Mostraram que os estudos teóricos e computacionais com detalhes algorítmicos podem ser mais economicamente viáveis do que um modelo determinístico clássico.

As técnicas de otimização determinísticas, que encontram a solução ótima, apresentam algumas limitações quanto às características de um problema de redes. Isso ocorre devido às funções a serem otimizadas, que não são contínuas, são não convexas, apresentam multimodalidade (vários pontos ótimos); existe a necessidade de se trabalhar com valores discretos para as variáveis, entre outras (CANTODOSSANTOS e BARRETO, 2011).

A heurística pode ser definida como uma técnica computacional que objetiva encontrar soluções aceitáveis com esforços computacionais considerados razoáveis e que apresenta condições para que as soluções se tornem viáveis. A meta-heurística é originada da heurística e que pode ser definida como uma arquitetura de regras, que através de um tema comum, pode guiar heurísticas específicas melhorando o espaço de busca e encontrando soluções (GOLDBARG, GOLDBARG e LUNA, 2016).

As Meta-heurísticas têm origem na Inteligência Artificial nas décadas de 60 e 70, mas somente na década de 90 que atingiu popularidade e maturidade. As principais características são: fácil implementação matemática pois guiam o processo de busca; capacidade de escapar de mínimos locais; produzem rapidamente boas soluções; não garantem soluções ótimas; são métodos probabilísticos e usualmente não-determinísticos (LUKE, 2015).

A introdução dos algoritmos genéticos foi primeiramente realizada por John Holland em 1975 (HOLLAND, 1975), com o objetivo de ser uma ferramenta para a resolução de problemas de otimização de grande complexidade, como por exemplo, o problema do caixeiro viajante. (COUËTOUX, GOURVÈS, *et al.*, 2010).

Cada possível solução de um problema é codificada em uma estrutura chamada de "cromossomo", que é composta por uma estrutura de dados que descreve os genes. Estes cromossomos representam indivíduos, que são evoluídos ao longo de várias gerações, analogamente aos seres vivos, de acordo com os princípios de seleção natural e sobrevivência dos mais aptos, descritos pela primeira vez por Charles Darwin em seu livro "A Origem das Espécies" (GOLDBARG, GOLDBARG e LUNA, 2016).

1.5 ORGANIZAÇÃO DO TEXTO

O Capítulo 2 apresenta a fundamentação teórica e descrição dos principais conceitos e estudos referentes ao planejamento de redes de telecomunicações e as principais tecnologias aplicadas.

O Capítulo 3 apresenta a metodologia e a descrição da biblioteca de instâncias de redes de telecomunicações SNDlib.

O Capítulo 4 compreende a apresentação e discussão dos resultados das simulações em uma rede PDH de baixa complexidade e para uma rede SDH com alta complexidade.

E por fim, o capítulo 5 apresenta as conclusões e resultados obtidos das simulações do algoritmo proposto para a otimização das redes de telecomunicações para instâncias PDH e SDH, além de indicar trabalhos futuros.

CAPÍTULO II

2. FUNDAMENTAÇÃO TEÓRICA

Esta etapa da dissertação é destinada para a descrição geral do sistema de telecomunicações com destaque no planejamento estratégico e a tomada de decisões que é o tema principal deste trabalho. São apresentadas nesta fundamentação teórica, tecnologias atuais em redes de telecomunicações.

2.1. REDES DE TELECOMUNICAÇÕES

O Regulamento Geral de Interconexão foi aprovado através da Resolução n.º 40, de 23 de julho de 1998 e no art. 3º, item VII definiu as redes de telecomunicações:

“Redes de Telecomunicações é o conjunto operacional contínuo de circuitos e equipamentos, incluindo funções de transmissão, comutação, multiplexação ou quaisquer outras indispensáveis à operação de serviço de telecomunicações.”

As redes de telecomunicações podem ser divididas em três grandes categorias (GUIOTTI, 2007):

Redes de Acesso: são plataformas de atendimento responsáveis por permitir a conexão de clientes (assinantes) ao seu provedor de serviços imediatos, neste caso as operadoras de telecomunicações. Possui baixa capacidade de multiplexação e disponibiliza meios físicos dedicados a clientes ou grupos de pequeno porte.

Plataformas ou Redes de Comutação: Possuem a função de disponibilizar e compartilhar recursos aos clientes que estão conectados através das redes de acesso. São responsáveis por otimizar recursos e possui características em estabelecer conexões em menor tempo.

Redes de Transporte ou redes de transmissão: São responsáveis pela transferência de informação de um ponto a outro da rede. Os sistemas de transmissão são compostos por meios físicos tais como: fibras ópticas e meios não físicos: o espaço livre.

Os meios de transmissão de Informação podem ser feitos através de cabo de pares (par trançado de cobre), cabo coaxial e o cabo de fibras ópticas através de conexões terrestres e submarinas. Os meios de transmissão através do espaço livre podem ser realizados através de rádio, micro-ondas, satélite, FSO (*Free-Space Optics*).

A Figura 1 apresenta a contextualização das redes de telecomunicações segundo Guiotti, 2007.

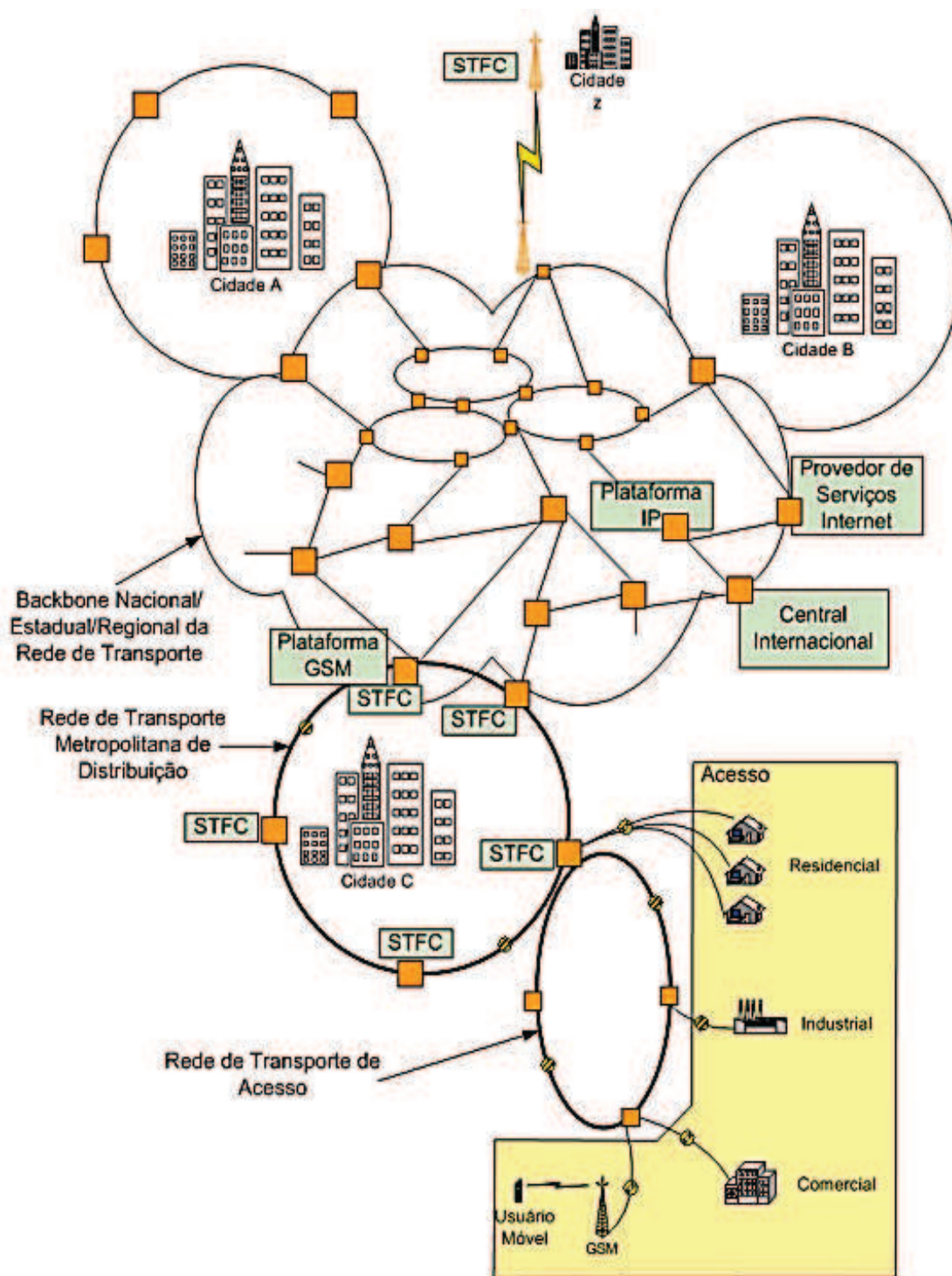


Figura 1 – Contextualização das redes de telecomunicações.
Fonte: (GUIOTTI, 2007).

Dentre os principais tipos de redes de telecomunicações pode-se citar as redes remotas WAN – *Wide Area Network* que abrangem grandes áreas geográficas tais como: grandes áreas metropolitanas, países e continentes. A Figura 2 apresenta uma rede WAN conectando países e continentes. A ligação entre a América do Sul e Europa é feita por

um único cabo submarino: o Atlantis-2 cobrindo a extensão de 12.000 km e capacidade de 40 Gbps.

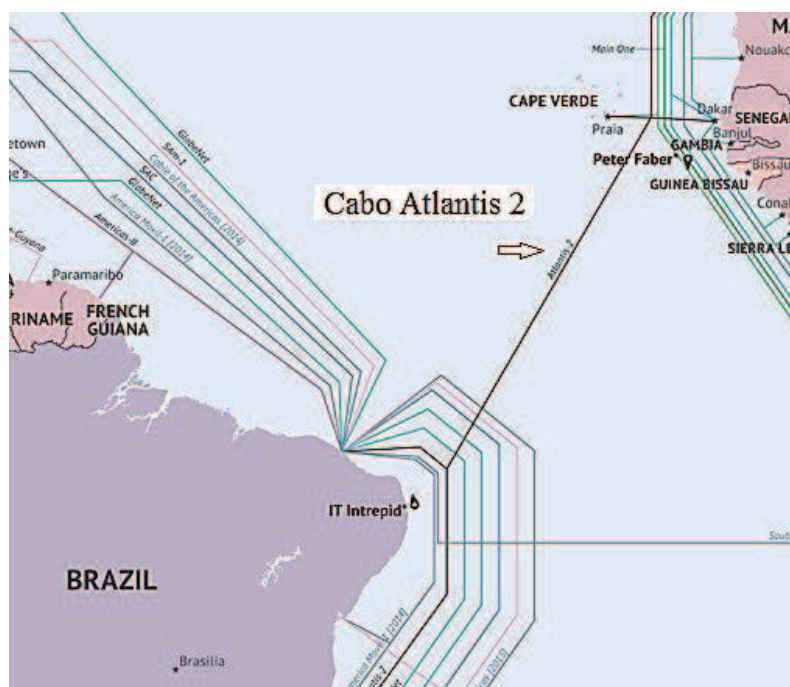


Figura 2 – Exemplo de Rede WAN

Fonte: <http://submarine-cable-map-2014.telegeography.com>

A Figura 3 apresenta a conexão de redes ópticas baseadas em malhas entre países europeus.



Figura 3 – Redes ópticas baseadas em malhas.

Fonte: (VASSEUR, PICKAVET e DEMEESTER, 2004).

2.2. PRINCIPAIS TECNOLOGIAS UTILIZADAS EM REDES DE TELECOMUNICAÇÕES

A seguir são apresentadas as principais tecnologias envolvidas no planejamento estratégico e tomada de decisões e estão presentes nas redes de transmissão das operadoras de telecomunicações.

As redes de telecomunicações, no início da década de 80 tornaram-se cada vez mais complexas devido às técnicas digitais que proporcionaram maiores demandas e maiores números de usuários para as grandes empresas de telecomunicações. Desta forma, as operadoras não poderiam atender a aumentos de grandes magnitudes com os padrões de transmissão existentes (PASTOR, 2007).

As hierarquias digitais obtidas com a multiplexação TDM (*Time Division Multiplexing*) são padronizadas por organizações da área de telecomunicações como o ITU-T (International Telecommunication Union – Telecommunication Standardization Sector) ou ANSI (American National Standards Institute). A padronização foi direcionada para uma velocidade de 64 kbps que é destinada para o transporte de comunicação de voz. A técnica TDM foi estruturada nas redes PSTN (*Public Switched Telephone Network*), com o objetivo de transportar voz utilizando a tecnologia PDH (PASTOR, 2007).

Esta era foi marcada pela técnica digital de multiplexação plesiócrons conhecida atualmente por PDH (*Plesiochronous Digital Hierarchy*) que posteriormente o método de multiplexação passaria a ser síncrono.

Surgiram, então, o SDH (*Synchronous Digital Hierarchy*) e o SONET (*Synchronous Optical Digital Network*) como uma nova tecnologia com maior confiabilidade, flexibilidade de comutação de taxas primárias, com interfaceamento padronizado, com maiores controles de gerenciamento e atendimento à novos serviços (BABATUNDE e MBAROUK, 2014).

A Figura 4 apresenta uma divisão simplificada das redes de telecomunicações atuais das principais tecnologias empregadas atualmente. Pode-se observar que as tecnologias PDH e SDH (*Synchronous Digital Hierarchy*) fazem parte da rede de transporte e serão descritas nos itens subsequentes.

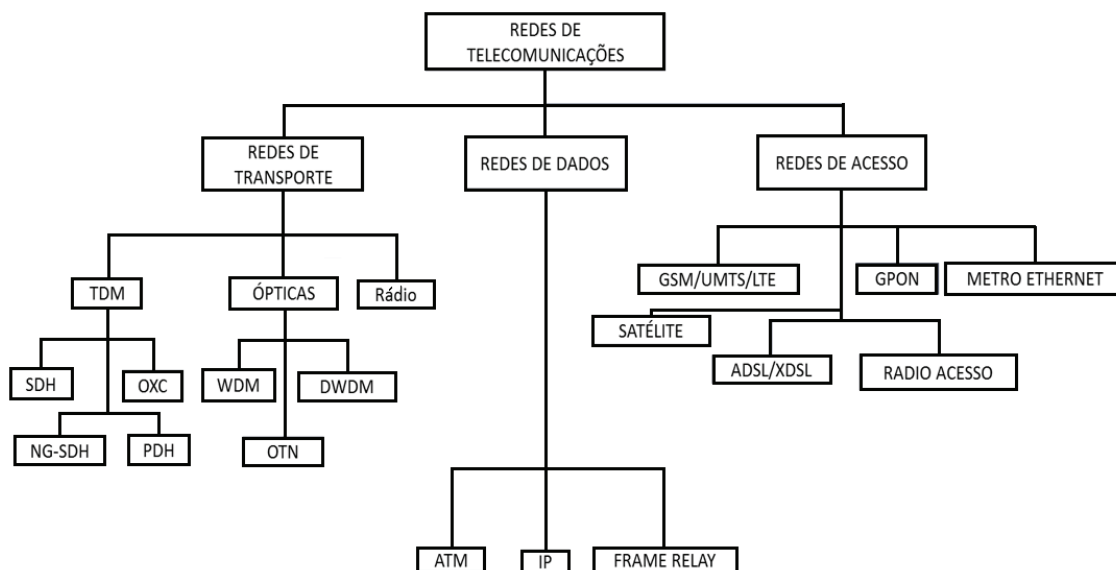


Figura 4 – Divisão simplificada das Redes de Telecomunicações

Para as redes de transporte, a tecnologia TDM permite transmitir simultaneamente vários sinais pois realiza a multiplexação determinística ou síncrona. Na Figura 4, os sistemas de transporte englobam os sistemas SDH, OXC (*Optical Cross-Connect*), NG-SDH (*Next Generation – Synchronous Digital Hierarchy*) e PDH.

Para as redes de transporte ópticas associa-se técnicas e protocolos tais como: DWDM (*Dense Wavelength Division Multiplexing*), WDM (*Wavelength-Division Multiplex*) e a OTN (*Optical Transport Network*), que está em evidência pois representa a evolução das redes de transporte.

As redes de dados de altas velocidades envolvem as tecnologias ATM (*Asynchronous Transfer Mode*), IP (*Internet Protocol*) e Frame Relay que é uma tecnologia de comutação de quadros.

As redes de acesso compreendem as tecnologias:

GSM (*Global System for Mobile*), UMTS (*Universal Mobile Telecommunication System*), LTE (*Long-Term Evolution*), GPON (*Gigabit passive optical network*), Metro Ethernet, Satélite e a tecnologia de transmissão de dados através de linhas telefônicas ADSL (*Asymmetric Digital Subscriber Line*), xDSL (*Digital Subscriber Line (x: significa protocolo genérico)*) e Rádio Acesso.

2.2.1. Tecnologia PDH

A tecnologia PDH surgiu com o objetivo de atender serviços de voz e que a técnica de multiplexação por divisão no tempo (TDM) era empregado para canais de comunicação. A multiplexação consiste em combinar dois ou mais canais em apenas um

meio de transmissão. Como exemplo, a conversão paralelo-série de sinais em um processo de compartilhamento permitindo múltiplos usuários. A transmissão de vários canais foi possível devido à PCM (*Pulse Modulation Codificated*). Este processo de digitalização permitiu enviar oito mil amostras por segundo com resolução de oito bits, resultando em canais de 64kbps em um total de 32 canais, sendo que 02 canais eram destinados para a sinalização.

A Tabela 1 mostra os valores do padrão PDH na Europa, América do Norte e Japão onde a primeira coluna apresenta a localidade, a segunda os níveis de hierarquias, e as demais a quantidade de canal com a respectiva taxa e nome aplicável. Vale ressaltar que no Brasil o padrão adotado é o mesmo que o europeu da qual é recomendado pela ITU G.702 (ITU 1988, 1993).

Tabela 1 – Níveis de hierarquia PDH, canais e taxas

Local	Hierarquia Digital (Nível)	Canais	Taxa de bits (kbps)	Nome
América do Norte	Canal de Dados/Voz	-	64	DS0
	1	24	1544	DS1
	Intermediário	48	3152	DS1C
	2	96	6312	DS2
	3	672	44736	DS3
	4	4032	274176	DS4
Europa	5	5760	400352	DS5
	Canal de Dados/Voz	-	64	E0
	1	32	2048	E1
	2	128	8448	E2
	3	512	34368	E3
	4	4032	139264	E4
Japão	5	5760	565148	E5
	Canal de Dados/Voz	-	64	-
	1	24	1544	-
	2	96	6312	-
	3	120	7786	-
	4	480	32064	-
	5	1440	97728	-
		8192	565148	-

Fonte: Adaptado de G.702 ITU-T
(ITU-T 1988, 1993)

Os maiores problemas apresentados pela tecnologia PDH é por possuir arquitetura assíncrona, uma não padronização das taxas de bits e posições rígidas dos tributários, tornando-o ineficiente e inflexível dificultando a identificação individual dos canais (BABATUNDE e MBAROUK, 2014).

Os Padrões Digitais de Multiplexação PDH e as principais diferenças entre eles são apresentados na Tabela 2, sendo que o padrão adotado no Brasil é o Sistema E (*E-Carrier*), padrão Europeu. No Japão e América do Norte o padrão é o Sistema T (*T-Carrier*) (BABATUNDE e MBAROUK, 2014).

Tabela 2 – Diferenças principais entre o Sistema T (*T-Carrier*) e o Sistema E

Sistema T (<i>T-Carrier</i>)	Sistema E (<i>E-Carrier</i>)
24 canais de 64Kbps no primeiro nível de multiplexação – T1 (1544 Mbps)	32 canais de 64Kbps no primeiro nível de multiplexação – E1 (2048 Mbps)
Cada quadro tem um <i>bit</i> adicional e não tem canal de controle.	2 canais de controle (1 para sincronização e outro para sinalização)
Codificação AMI	Codificação HDB3
Quadro de 193 <i>bits</i>	Quadro de 256 <i>bits</i>

Fonte: Adaptado de G.707 ITU-T (ITU-T, 2003)

Na Tabela 2, pode-se observar que dos 32 canais dos Sistema E, 30 são reservados para voz e 02 canais reservados para o controle. Cada *slot* transporta uma amostra PCM composta por 8 *bits*, assim, cada *frame* transporta 32 canais de 8 *bits*. Um *frame* é igual a 256 *bits* que é o produto de 32 x 8. A Capacidade para o sistema *E-Carrier* é igual a 2048 kbps que é o produto de 256 *bits* x 8000 amostras, enquanto que para o sistema *T-Carrier* é igual a 1544 kbps que é o produto de 193 *bits* x 8000 amostras.

O multiplexador apresentado na Figura 5 realiza a conversão paralelo série de sinais. Como exemplo, os 04 sinais de entrada possuem frequência de 100 Hz e os converte em apenas 01 sinal de saída com uma frequência de 400 Hz.

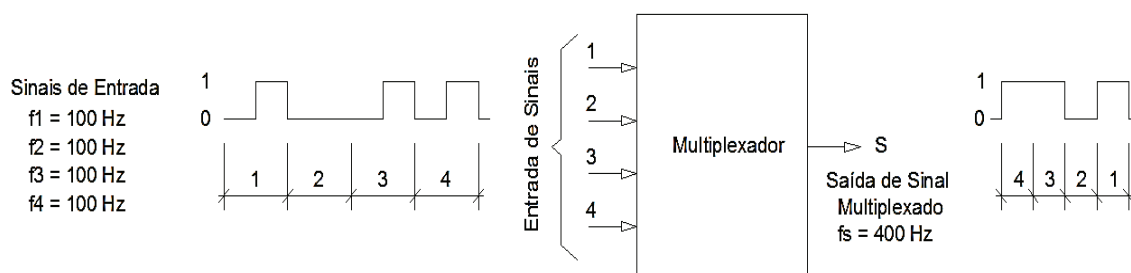


Figura 5 – Multiplexador 4x1 – Conversão Paralelo Série

2.2.2. Tecnologia SDH / SONET

Em 1988 a ITU-T publicou com base nas especificações da rede digital síncrona norte americana a tecnologia SONET caracterizando-se em um novo padrão internacional, designado por SDH. Em 1992, para ser utilizada na América do Norte, o padrão SONET foi concedido pela Bellcore e definida pela ANSI (PALLE, 2013).

A tecnologia SDH realiza a multiplexação por divisão do tempo - TDM podendo ser determinística ou síncrona. Porém, o sistema de transporte de informações opera em velocidade altas e é muito utilizado para acessos à Internet. As velocidades síncronas de operação são apresentadas nas Tabela 3 (TANENBAUM A. S., 2003):

Tabela 3 – Taxa de Multiplexação da SONET e SDH

SONET		SDH	Taxa de dados (Mbps)		
Elétrico	Óptico	Óptico	Gross (Bruto)	¹ SPE	Usual
STS-1	OC-1	—	51,84	50,112	49,536
STS-3	OC-3	STM-1	155,52	150,336	148,608
STS-9	OC-9	STM-3	466,56	451,008	445,824
STS-12	OC-12	STM-4	622,08	601,344	594,432
STS-18	OC-18	STM-6	933,12	902,016	891,648
STS-24	OC-24	STM-8	1.244,16	1.202,688	1.188,864
STS-36	OC-36	STM-12	1.866,24	1.804,032	1.783,296
STS-48	OC-48	STM-16	2.488,32	2.405,376	2.377,728
STS-192	OC-192	STM-64	9953,28	9.621,504	9.510,912

Fonte: Tanenbaum A. S., (2003)

Cada canal opera com um relógio sincronizado com os relógios dos outros canais, e é sincronizado com o equipamento multiplex através de um processo de justificação de *bit* e encapsulamento da informação (*containers*).

O transmissor possui um relógio que regula o *timing* dos *bits* transmitidos, isto é, para cada 1 Mbps (1 *bit* a cada 1 ms), o receptor fará uma amostragem a cada 1 ms baseado em seu próprio relógio. Se os relógios de transmissão e recepção não estiverem precisamente alinhados, ocorrerá um problema.

O SDH pode transportar diferentes tipos de sinais PDH, através do padrão STM-N (*Synchronous Transport Module*), utilizado tanto para sinais elétricos como para sinais ópticos, conforme foi apresentado na Tabela 3. O SDH pode também multiplexar *frames* baseadas em transmissão de pacotes, como sinais de *Ethernet* (JYOTHIRMAI, VALLI e KRISHNA, 2012).

2.2.3. Comparação entre as tecnologias SDH e PDH

Segundo Babatunde e Mbarouk (2014), as redes PDH e SDH possuem características e diferenças que são mostradas na Tabela 4:

Tabela 4 – Principais diferenças entre as tecnologias SDH e PDH

SDH	PDH
Transmissão síncrona de sinais STM-N	Transmissão plesiócrona
Organização em <i>bytes</i>	Entrelaçamento em <i>bits</i>
Uniformidade no comprimento dos quadros (125µs)	Não possui uniformidade
Utiliza ponteiros para indicar o início de cada quadro e processar eventuais justificações	Utiliza palavras de alinhamento
Alta capacidade de gerência e supervisão	Não possui gerência e supervisão
Compatibilidade	Não compacto
Possui um único padrão mundial	Várias padronizações
A transmissão através de <i>Cross-Connections</i> e <i>Add/Drop</i> .	Transmissão ponto-a-ponto.

¹ SPE – (do inglês: Synchronous Payload Envelope - Envelope Síncrono de Carga Útil)

As tecnologias PDH e SDH são utilizadas nos sistemas de telecomunicações para transmissão digital de sinais. Com os avanços tecnológicos a tecnologia PDH foi substituída pela SDH devido a uma série de carências apresentadas principalmente para o gerenciamento das redes de comunicação, vários padrões utilizados e ineficiência para agregar novos sistemas. O sistema SDH apresentou maior confiabilidade e flexibilidade.

2.3. OTIMIZAÇÃO LINEAR

A otimização linear é inerente ao planejamento estratégico e se resume na busca pela melhor solução de um determinado problema. A elaboração e implementação de técnicas e métodos de otimização possibilita melhores alternativas no processo de tomada de decisão.

Um problema de otimização é caracterizado matematicamente por uma função objetivo, que representa o que se deseja maximizar (ou minimizar), as variáveis de decisão, que são os parâmetros que devem ser escolhidos para que se atinja o objetivo e por último as restrições do problema. As escolhas das variáveis de decisão devem obedecer às restrições impostas pelo problema a ser resolvido. Por exemplo o problema de alocação de aulas às salas, possui a restrição de que uma sala não pode atender à duas aulas ao mesmo tempo (CIRINO, SANTOS e DELBEM, 2015).

Uma definição generalista de um problema de otimização linear é descrita nas equações (1), (2) e (3), a seguir: 1.

$$\text{Minimizar } \mathbf{f}(\mathbf{x}) = \mathbf{c}^T \cdot \mathbf{x} \quad (1)$$

$$\text{Sujeito a: } \mathbf{Ax} = \mathbf{b} \quad (2)$$

$$\mathbf{x} \geq \mathbf{0} \quad (3)$$

Onde:

$\mathbf{f}(\mathbf{x})$	Função objetivo;
\mathbf{x}	Variáveis de decisão;
\mathbf{b}	Lado direito das restrições;
\mathbf{c}	Coefficiente da função objetivo;
\mathbf{A}	Matriz A.

A função objetivo e os coeficientes do sistema de equações lineares conforme apresentados nas Equações (1), (2) e (3), são relevantes para a solução do problema e podem ser descritos na seguinte forma:

$x_1, x_2, x_3, \dots, x_n$	variáveis de decisão
$c_1, c_2, c_3, \dots, c_n$	coeficientes da função objetivo
$a_1, a_2, a_3, \dots, a_n$	a_j é a j -ésima coluna da matriz A

O problema de otimização geralmente apresenta um espaço de busca definida onde podem ser encontradas todas as possíveis soluções de um determinado problema (LACERDA e CARVALHO, 1999).

Para os mais diversos problemas de otimização, é necessário que o critério de avaliação de um conjunto de soluções possibilite indicar que uma solução é “melhor” que a outra. Porém em muitos problemas reais (como os problemas combinatoriais) as variáveis de decisão são inteiras, ou seja, estão contidas em um espaço não contínuo.

2.3.1. Otimização combinatória e grafos

A otimização combinatória pertence à área da ciência da computação e matemática aplicada. Nos últimos anos, vem se tornando mais eficiente devido a uma crescente tecnologia computacional e a evolução de algoritmos permitindo novas aplicações práticas e reais de programação inteira com diversas variáveis de decisão e restrições (GOLDBARG, GOLDBARG e LUNA, 2016).

Problemas de fluxo em redes, roteamento e distribuição, são exemplos de otimização combinatória, onde a formulação matemática do problema é feita através da teoria de grafos, como exemplo clássico temos o problema do caixeiro viajante (BOAVENTURA NETTO, 2011).

O problema do caixeiro viajante é um problema de otimização NP-difícil que tenta determinar a menor rota a percorrer de uma série de cidades, onde cada cidade deve ser visitada uma única vez e, por fim, retornar ao ponto de origem.

Esses problemas são tratados como redes, que possuem nós, arcos e informações intrínsecas a cada modelagem feita. Muitos desses problemas possuem uma restrição específica e de grande relevância para a solução do problema: as variáveis de decisão podem ser assumir apenas valores inteiros; são chamados de problemas de programação linear inteira mista (MELO, 2013).

Um grafo G é definido por um conjunto de nós V , e arestas E , ou seja, $G = (V, E)$. A Figura 6 mostra um grafo com arcos não direcionados, sendo o conjunto de vértices $V = \{A, B, C, D, E, F\}$ e o conjunto de arcos $E = \{E1, E2, E3, E4, E5, E6, E7, E8, E9, E10\}$.

Para que seja definida uma formulação matemática é necessário que o grafo seja representado, de forma que, nenhuma das suas informações sejam perdidas e que permita alterações.

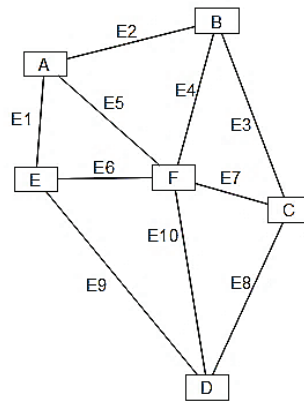


Figura 6 – Exemplo de Grafo não-direcionado

Fonte: Adaptado de (GOLDBARG e GOLDBARG, 2012)

Essa representação pode ser feita de diversas formas: lista ou matriz de adjacência e incidência, listas lineares ou encadeadas, ou ainda por vetorização. A Figura 7 corresponde à lista de incidência do grafo da Figura 6.

A	E1	E2	E5		
B	E2	E3	E4		
C	E3	E7	E8		
D	E8	E9	E10		
E	E1	E6	E9		
F	E4	E5	E6	E7	E10

Figura 7 – Lista de incidência para o grafo da Figura 6

Na Figura 6, para o vértice A existem três arcos conectados E1, E2 e E5. Como o grafo é não-direcionado, não existe a representação do sentido dos arcos. A representação do grafo por lista de incidência é simples e carrega apenas as informações necessárias, essa característica auxilia no processo de implementação de algum método de otimização, pois tem baixo custo computacional. A desvantagem é que a quantidade de arcos incidentes não é constante para todos os nós, o que dificulta a etapa de implementação.

Outra forma de representar um grafo é através de uma matriz de incidência que possui ordem $V \times V$, conforme observado na Figura 8. A matriz é preenchida com 0's e 1's de acordo com a seguinte regra:

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

	A	B	C	D	E	F
A	0	1	0	0	1	1
B	1	0	1	0	0	1
C	0	1	0	1	0	1
D	0	0	1	0	1	1
E	1	0	0	1	0	1
F	1	1	1	1	1	0

Figura 8 – Matriz de adjacência

Uma matriz de incidência tem um esforço computacional maior quando comparada à lista de incidência, porém independentemente do grafo a ser representado a sua ordem é constante, facilitando a implementação computacional.

2.3.2. Formulação Nó Arco e Arco Caminho

Muitos problemas envolvendo grafos estão relacionados com a atribuição de capacidade aos arcos e a distribuição do fluxo, como, por exemplo, problemas de otimização redes (água, gás, dados). Uma determinada demanda precisa sair de um nó (fonte) e chegar até outro nó (destino) contido no conjunto V de vértices. A forma que essa demanda é tratada no processo de modelagem podendo apresentar duas formulações distintas: a formulação nó arco e arco caminho.

A formulação nó arco utiliza o conceito de conservação de fluxo. Dessa forma o fluxo de demanda que chega a um determinado nó subtraído do fluxo que sai do nó deve ser igual a demanda do nó correspondente, onde a equação de balanço está sendo aplicada. Essa formulação pode ser aplicada apenas a grafos direcionados, pois na equação de conservação do fluxo leva-se em consideração se o arco está saindo ou chegando ao nó (PIÓRO e MEDHI, 2004).

Já a formulação arco caminho trabalha com a criação de uma lista de caminhos para que uma demanda seja escoada conforme apresentada na Figura 9.

Nos problemas de roteamento várias demandas precisam ser escoadas ao mesmo tempo, assim é escolhido um caminho para que cada demanda saia do seu nó de origem e chegue até o seu nó de destino.

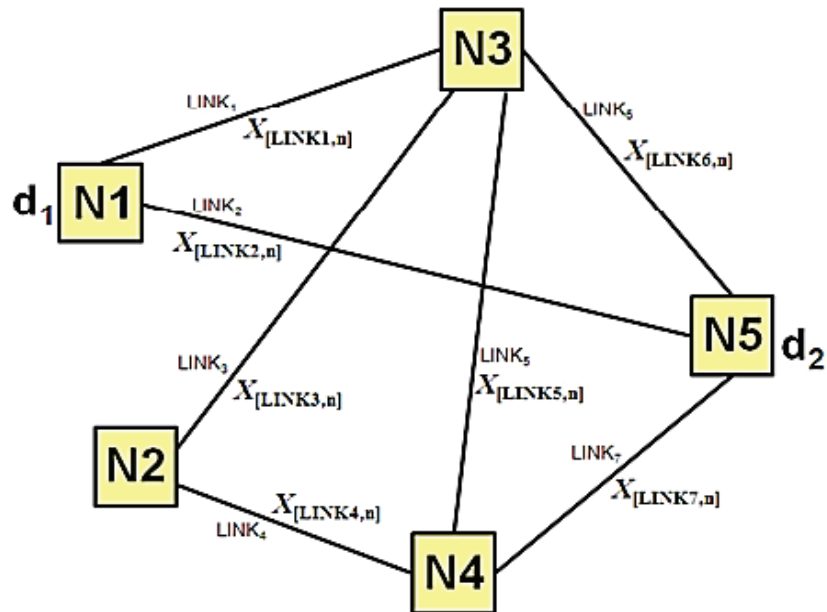


Figura 9 – Grafo não direcionado (arco-caminho)
Fonte: (OLIVEIRA, 2015)

Na rede da Figura 10 podem ser escoadas D demandas ($D \in \mathbb{N}$) com origem e destino dentro da rede. Os caminhos que podem ser utilizados para o escoamento dessas demandas devem estar no conjunto \mathbf{P} que contém todos os caminhos possíveis para essa rede (Figura 11).

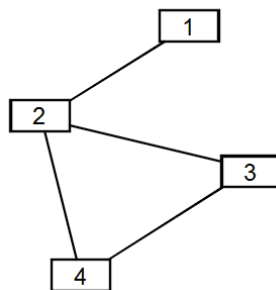


Figura 10 – Grafo não direcionado

A restrição relativa à escolha dos caminhos pode ser feita de duas formas diferentes: uma para quando os arcos possuem capacidade definidas, e outra para quando a capacidade precisa ser alocada no arco (PIÓRO e MEDHI, 2004).

Um sistema de distribuição que foi desenvolvido sem um planejamento adequado, o planejador precisa decidir qual a melhor forma de utilizar os recursos. Assim quando, na escolha dos caminhos, os mesmos devem ser escolhidos de forma a obedecer à capacidade máxima de cada arco.

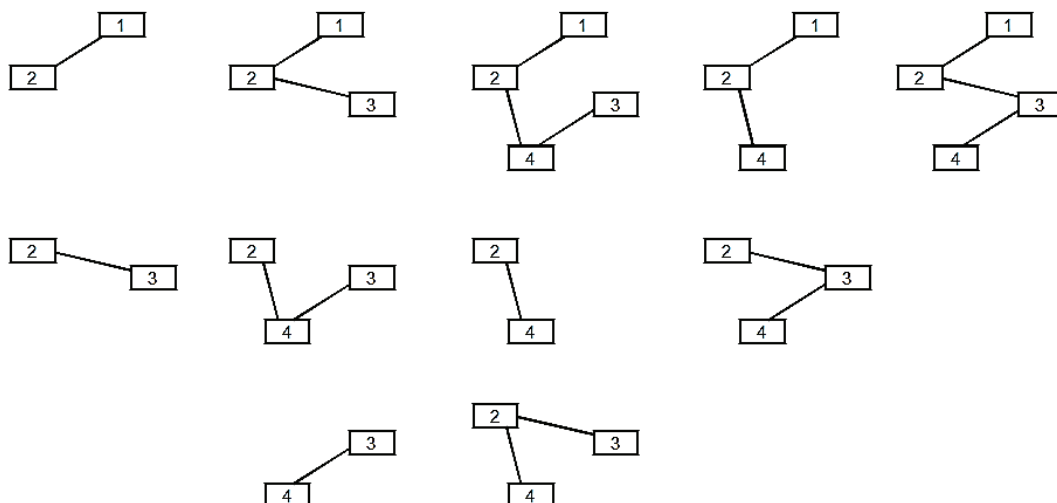


Figura 11 – Caminhos possíveis para o grafo direcionado de 04 nós

Outra abordagem que representa um problema ainda mais complexo é quando as capacidades dos arcos são variáveis a serem definidas, isso acontece em sistemas a serem implantados, como por exemplo, uma rede de telecomunicação que pode ser descrita como um grafo. Dessa forma a limitação passa a ser técnica, pois a soma de demandas sendo escoadas por um arco deve ser menor que a capacidade máxima disponível para ser alocada em um arco da rede. A vantagem de se utilizar a formulação arco caminho está no fato de se poder trabalhar com grafos direcionados, não direcionados ou mistos.

2.3.3. Algoritmos Genéticos.

Os Algoritmos Genéticos (AGs) foram introduzidos por John Holland através de estudos baseados em processos evolutivos e apresentou de forma que pudesse estudar a adaptação e a evolução relacionando ao mundo real (HOLLAND, 1975).

Os Algoritmos Genéticos são métodos de otimização que se originaram dos algoritmos evolucionários. Pode-se dizer que são técnicas heurísticas de otimização e que seguem o princípio da seleção natural apresentada pelo inglês Charles Darwin em seu livro, *A Origem das Espécies* (LINDEN, 2012).

Os AGs foram popularizados por David Goldberg e tornaram-se uma excelente ferramenta muito explorada e aplicada por cientistas da área de computação para a resolução de problemas (LINDEN, 2012).

A partir de um conjunto de soluções, os indivíduos são escolhidos aleatoriamente formando a população inicial.

Os indivíduos mais aptos são selecionados para a próxima geração e recombinados para formar uma nova população. Através do cruzamento, gerando filhos, que ainda podem sofrer um processo de mutação.

A nova população então é utilizada como entrada para a próxima iteração do algoritmo e a cada geração, a adaptação de cada solução é avaliada através da função *fitness*.

Os operadores genéticos tais como seleção, cruzamento e mutação são responsáveis por aprimorar a solução no decorrer das gerações do algoritmo (TANOMARU, 1995).

Os algoritmos genéticos possuem características principais aplicáveis em problemas com muitas variáveis e com um conjunto de soluções indefinidas e são aplicados às diversas áreas de conhecimentos e aplicação (MELO, 2005).

A Figura 12 mostra uma representação gráfica dos algoritmos genéticos e pode-se observar que os operadores genéticos recombina os pais escolhidos na etapa de seleção.

A avaliação aplicada a cada um dos indivíduos da geração e o módulo da população (nova população) é definido a partir da geração existente e dos filhos gerados. Os sobreviventes (filhos) e suas características são transferidas, isto é, copiadas sobre seus pais. Os ‘indivíduos desconsiderados’ referem-se a toda antiga geração de pais que é descartada e o critério de parada é em relação ao número de gerações ou pela qualidade das gerações (LINDEN, 2012).

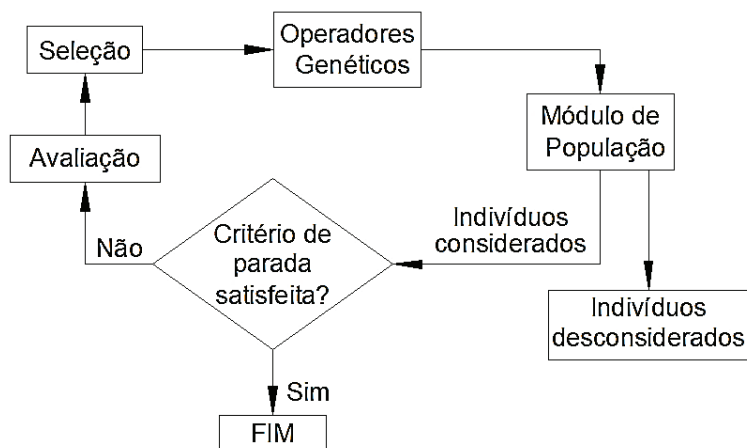


Figura 12 – Representação gráfica dos algoritmos genéticos
Fonte: Adaptado (LINDEN, 2012)

Em 2014, Elsayed, Sarke e Essam mostraram que o sucesso de qualquer algoritmo genético depende de como é a elaboração e especificação dos seus operadores de pesquisa, bem como da sua implementação adequada. Propuseram um operador de

diversidade para ser utilizado no lugar do operador de mutação e também manter um arquivo de boas soluções. Além disso, apresentaram um novo operador de *crossover* de três pais com GA que gera três novas populações. (ELSAYED, SARKE e ESSAM, 2014).

2.3.4. Problemas intratáveis

Problemas da ordem fatorial são considerados problemas intratáveis devido ao tempo de resolução. O tempo de execução se torna cada vez mais intratável com problemas de ordem exponencial, conforme apresentado na Tabela 5.

Tabela 5 – Tempo de execução associados a funções

n	n^2	n^3	2^n	$n!$
10	10^2	10^3	$\approx 10^3$	$\approx 10^6$
100	10^4	10^6	$\approx 10^{30}$	$\approx 10^{158}$
1000	10^6	10^9	$\approx 10^{300}$	$\approx 10^{1500}$
10000	10^8	10^{12}	$> 10^{3000}$	$\gg 10^{10000}$

Fonte: (LINDEN, 2012)

Onde: n = número de operações

Observa-se que na Tabela 5, problemas intratáveis são aqueles que possuem um tempo inaceitável para encontrar uma solução pois o seu limite superior de complexidade é exponencial ou fatorial. Como exemplo: 2^n ou $n!$

2.3.5. Codificação

Antes da escolha dos operadores é necessário primeiramente definir como as propostas de solução do problema (os indivíduos) serão representadas, ou seja, definir a codificação do problema.

A codificação mais comum é a codificação binária que foi utilizada nos trabalhos iniciais de John Holland. Mesmo não sendo computacionalmente eficiente, possui relevância histórica e tem sido utilizada devido sua simplicidade. A codificação deve ser escolhida de forma que não haja perda de informações no processo de codificação e decodificação.

Um exemplo de codificação pode, por exemplo, mapear pontos no espaço de soluções de cadeias de *bits* (ENGELBRECHT, 2007):

Considere uma função $f(x, b)$ de v , variáveis reais e binárias, de acordo com (5) e (6):

$$x = (x_0, x_1, x_2, x_3, \dots, x_{v-1}) \quad \text{onde } x_i \in \mathfrak{R} \quad (5)$$

$$b = (b_0, b_1, b_2, b_3, \dots, b_{v-1}) \quad \text{onde } b_i \in \{0,1\} \quad (6)$$

Considerando uma codificação que atribui para x_0 , x_1 , x_2 e x_3 valores reais de 5, 2, 12 e 14 respectivamente. Sendo assim, obtemos um cromossomo de 16 *bits*, com genes de 4 bits, podendo ser escrito da seguinte forma: 0101-0010-1100-1110.

2.3.6. Aptidão e Seleção

Estes operadores são fundamentais para que o algoritmo tenha uma convergência com maiores índices de aceitação e possibilite melhores soluções, pois a avaliação dos indivíduos contribuirá para resultados satisfatórios.

A aptidão ou *fitness* está ligada diretamente com o grau de adaptação de cada indivíduo ao ambiente e está relacionada com a escolha dos mais aptos de forma que ocorra a reprodução passando parte de seu material genético a gerações futuras (TANOMARU, 1995).

A função de avaliação é a que determina a qualidade de um indivíduo e é a única ligação verdadeira do programa com o problema real e que não armazena nenhum tipo de informação da solução que se pretende encontrar. A função de avaliação sempre contempla de forma numérica os objetivos do problema (LINDEN, 2012).

O operador de seleção é considerado um operador Darwiniano, pois surgiu na teoria da evolução e é utilizado para direcionar o processo de otimização para as melhores regiões do espaço de busca (MITCHELL, 1996).

2.3.7. Operador de *crossover*

O *crossover* é o operador pelo qual bons indivíduos repassam suas características para a próxima geração. Em conjunto com o operador de seleção garante que a busca do algoritmo genético não seja totalmente aleatória (PRATA, 2015).

O *crossover* mais encontrado na literatura, também utilizado por Holland é *crossover* de um ponto para codificação binária, onde um indivíduo (cadeia de *bits*) é cortado em um ponto aleatório; e após o corte dois indivíduos têm suas informações cruzadas, produzindo dois novos indivíduos (filhos) que irão compor a próxima geração.

A Figura 13 apresenta um *crossover* de um ponto e evidencia a posição entre dois genes de um determinado cromossomo e para cada indivíduo de n genes compreende $n-1$ pontos de corte (LINDEN, 2012).

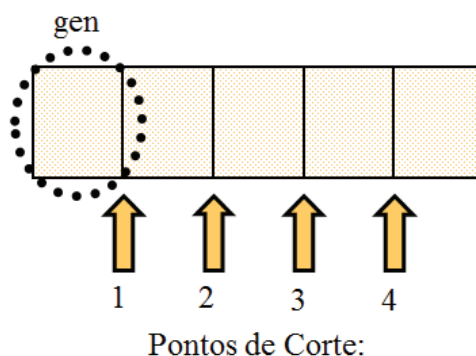


Figura 13 – Exemplo de pontos de corte
Fonte: (LINDEN, 2012)

A Figura 14 apresenta o *crossover* com as seguintes características:

- Após a seleção dos dois pais, é escolhido um ponto de corte, que constitui de dois genes de um cromossomo;
- Os pais são separados em duas partes, após o sorteio do ponto de corte: uma à esquerda e outra à direita;
- O primeiro filho surge através da combinação da parte esquerda do 1.º pai com a parte direita do 2.º pai;
- O segundo filho é composto através da combinação das partes que restaram, isto é, a metade à esquerda do segundo pai com a metade à direita do primeiro pai;
- Não é necessário que as duas partes possuam o mesmo tamanho.

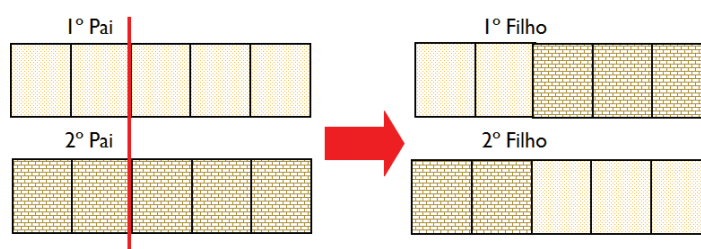


Figura 14 – Exemplo de *crossover*
(LINDEN, 2012)

O operador de *crossover* é um método de compartilhamento de informação entre os cromossomos que combina as características de dois cromossomos pais para formar dois filhos, com a possibilidade de bons cromossomos gerarem filhos melhores.

O operador de recombinação não é normalmente aplicado a todos os pares de cromossomos da população e a escolha é realizada aleatoriamente. A probabilidade de cruzamento a ser aplicada depende da probabilidade definida por uma taxa de *crossover*, conhecida como probabilidade de cruzamento (HERRERA, LOZANO e VERDEGAY, 1998).

2.3.8. Operador de Mutação

Os algoritmos genéticos realizam uma busca aleatório-direcionada, e a mutação garante o caráter aleatório da busca (GOLDBERG, 1989). No decorrer das gerações a população pode convergir para um ponto ótimo local e assim o *crossover* não consegue retirar a população estagnada do local, dessa forma, é necessário que aconteça mutação para que isto não ocorra. A mutação também é controlada por uma taxa, que deve ser menor que a do *crossover*. Ao contrário, o processo de busca torna-se totalmente aleatório. O processo de mutação mais comum em uma cadeia de *bits* é trocar um *bit* 1 por um *bit* 0.

O operador responsável pela mutação altera arbitrariamente um ou mais componentes (de genes) de um cromossomo selecionado de tal forma a aumentar a variabilidade estrutural da população. O papel principal da mutação é o de restaurar material genético perdido ou não explorado na população para evitar a convergência prematura do algoritmo genético para soluções abaixo do ideal. Ele certifica que a probabilidade de alcançar qualquer ponto no espaço de busca nunca é zero. A posição de cada cromossomo na população sofre uma mudança aleatória de acordo com uma probabilidade definida por uma taxa de mutação. (HERRERA, LOZANO e VERDEGAY, 1998).

A Figura 15 apresenta a mutação fundamentada na troca de posição de um gene qualquer e a Figura 16 mostra a mutação baseada na troca de ordem de dois cromossomos (LACERDA e CARVALHO, 1999):

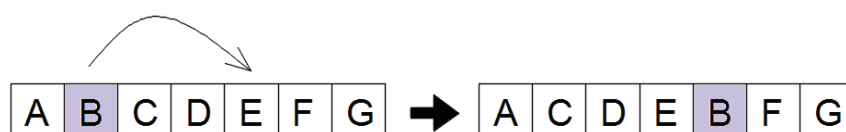


Figura 15 – Mutação baseada na troca de posição de genes (*i para j*)
Fonte: (LACERDA e CARVALHO, 1999)

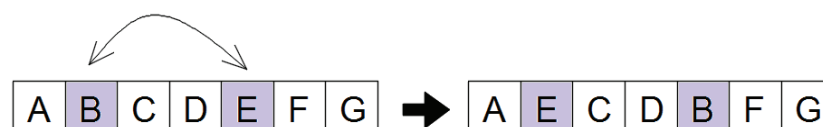


Figura 16 – Mutação baseada na troca de cromossomos
Fonte: (LACERDA e CARVALHO, 1999)

2.3.9. Elitismo

O elitismo não ocorre no processo natural de evolução, por isso é tido como técnica de auxílio à convergência (GOLDBERG, 1989). Baseia-se em manter a melhor solução encontrada em uma geração para a seguinte, para que não aconteça a perda de boas soluções.

Segundo Andries Engelbrecht (2007), o elitismo refere-se ao processo de assegurar que os melhores indivíduos da população atual sobrevivam para a próxima geração. As características dos melhores indivíduos são para a nova população sem ser mutado e quanto mais indivíduos sobrevivam à geração seguinte, menor será a diversidade da nova população.

2.4. A BIBLIOTECA DE PROJETOS DE REDES FIXAS DE TELECOMUNICAÇÕES - SNDlib

A SNDlib (*Survivable Network Design Library*) é uma biblioteca com instâncias e dados de redes de telecomunicações sendo apresentada pela primeira vez, em 2005, na Conferência Internacional DRCN (*Design of Reliable Communications Networks*) por Michał Pióro e Roland Wessäly (ZUSE INSTITUTE BERLIN (ZIB), 2005).

A biblioteca disponibiliza dados de projetos de redes fixas de telecomunicações cuja finalidade é fazer com que instâncias de rede estejam disponíveis para a comunidade científica servindo como um ponto de referência para que pesquisadores e profissionais da área possam testar avaliar e comparar modelos de projetos de rede e algoritmos (ZUSE INSTITUTE BERLIN (ZIB), 2005).

Há muitas pessoas envolvidas e que têm ajudado a construir a biblioteca, fornecendo exemplos de projetos e até mesmo programação. É atualmente mantida pelo ZIB (*Zuse Institute Berlin*) por Christian Raack e Kai Hennig com a ajuda de Sebastian Orłowski e Klähne Roman (ORŁOWSKI, WESSÄLY, *et al.*, 2010).

A biblioteca de redes fixas de telecomunicações disponibiliza dados em duas partes. A primeira parte é designada para modelagem das redes de acordo com modelo de demanda, *links*, capacidades, custo, roteamento, caminhos, número máximo de arcos e sobrevivência. A segunda parte corresponde aos dados da rede que apresenta o grafo com *links* (arcos), nós (vértices) e valores fixos de demanda e o custo de implantação para cada arco.

Após selecionar o modelo da instância, é possível encontrar os resultados obtidos por métodos determinísticos, bem como os detalhes desses resultados conforme ilustrado na Tabela 6, para uma determinada instância, no caso a rede Atlanta:

Tabela 6 – Resultados da rede atlanta--D-B-M-N-C-A-N-N

Custo Total	37.213.479,00
Total de <i>links</i> instalados	22
Capacidade total de <i>links</i> instalados	162.000,00
Uso da capacidade média dos <i>links</i>	91,48%
Comprimento médio do caminho	2,67 <i>links</i>
Número total de caminhos	224
Média de caminhos por demanda	1,07
Soluções estatísticas	mostrar detalhes...
Solução de exportação	GML (ver formato GML)
Contribuidor	Christian Raack, Zuse-Institute Berlin
Data da submissão	06/11/2006
Informação adicional	Fluxo de borda formulação, CPLEX 10.0, tempo limite: 1h

Fonte: SNDlib (ZUSE INSTITUTE BERLIN (ZIB), 2005)

A Figura 17 apresenta o processo de planejamento da biblioteca SNDlib. A Interface SNDlib permite que a Instância de Problema SNDlib possa ser escolhida dentre as 22 instâncias disponibilizadas. Os parâmetros planejados e os dados da rede são enviados através de uma Interface Matemática estabelecendo um Problema de Otimização. A solução para o problema de otimização é apresentada após o processo de otimização realizado pelo algoritmo.

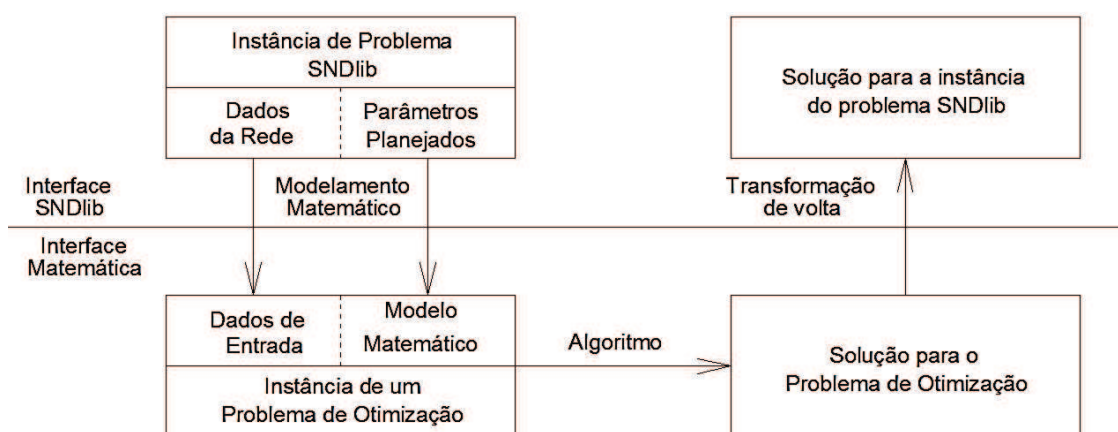


Figura 17 - Processo de Planejamento da biblioteca SNDlib
(ORLOWSKI, WESSÁLY, *et al.*, 2010)

A Figura 18 apresenta a interface gráfica disponível em <http://sndlib.zib.de/home.action> e após a seleção dos parâmetros da instância pdh, a solução SNDlib é disponibilizada com informações resumidas da rede otimizada no item *Table Legend*.

Model filter

Demand model: Undirected demands (U) | pdh
 Link model: Undirected links (U) | abilene
 Link capacity model: Explicit link capacities (E) | atlanta
 Fixed-charge model: No fixed-charge cost (N) | brain
 Routing model: Single path (S) | cost266
 Admissible path model: All paths (A) | dfn-bwin
 Hop-limit model: No hop-limits (N) | dfn-gwin
 Survivability model: No survivability (N) | di-yuan
 | france
 | geant
 | germany50
 | giul39
 | india35
 | janos-us
 | janos-us-ca
 | newyork
 | nobel-eu
 | nobel-germany
 | nobel-us
 | norway
 | pdh

Apply selection

Selection: pdh--U-U-E-N-S-A-N-N

Table Legend

status	network--model	V	E
●	pdh--U-U-E-N-S-A-N-N	11	34

D	best-sol	best-dual	gap
24	14883708.00	11114202.00	33.92%

Figura 18 – Página da biblioteca de redes SNDlib
 Fonte: (ZUSE INSTITUTE BERLIN (ZIB), 2005)

A biblioteca de redes SNDlib, disponibiliza os arquivos salvos na forma *native*, *xml*, ou por meios de pacotes compactados nos formatos *zip/tgz*, conforme apresentado na Figura 19:

Problem pdh--U-U-E-N-S-A-N-N

Problem	download: native / zip / tgz, xml / zip / tgz
Network	show source: native / xml
Model	show source: native / xml
Nodes	11
Links	34
Demands	24
Problem statistics	show details...
Network image	show image...
Problem export	GML (see GML format)
Originator	-
Contributor	-
Submission date	17.10.2005
Submit new solution or dual bound	go...

Solutions provided to SNDlib:

objective	contributor	submission date	additional information
14883708.00	Sebastian Orłowski	20.12.2005	path-flow LP with column generation, BSC with metric inequalities, time limit 1h

[statistics about all solutions.](#)

Dual bounds provided to SNDlib:

dual bound	contributor	submission date	additional information
11114202.00	Christian Raack	06.11.2005	taken from pdh--U-U-E-N-C-A-N-N
9348465.47	Sebastian Orłowski	19.12.2005	taken from pdh--U-U-E-N-C-A-N-N

Note: we encourage you to submit dual bounds to a problem if you have them. However, in contrast to primal solutions, we cannot check dual bounds for validity, which is why dual bounds are only listed here together with their contributor.

Figura 19 – Página da biblioteca de redes SNDlib
 Fonte: (ZUSE INSTITUTE BERLIN (ZIB), 2005)

CAPÍTULO III

3. METODOLOGIA

3.1. CLASSIFICAÇÃO DA PESQUISA

Com o objetivo de analisar e testar a complexidade de redes de telecomunicações, a abordagem para esta pesquisa é classificada como quantitativa e todos os dados coletados na biblioteca de redes SNDlib foram determinantes para o desenvolvimento do modelo matemático. Quanto ao método de pesquisa aplicado a este trabalho, a modelagem e simulações permitiram o uso de técnicas matemáticas para descrever os sistemas de telecomunicações, bem como as simulações computacionais foram essenciais para representar os sistemas já referidos. A Figura 20 exemplifica um processo simplificado da abordagem de solução de um problema utilizando a modelagem matemática (CAUCHICK MIGUEL, FLEURY, *et al.*, 2012).

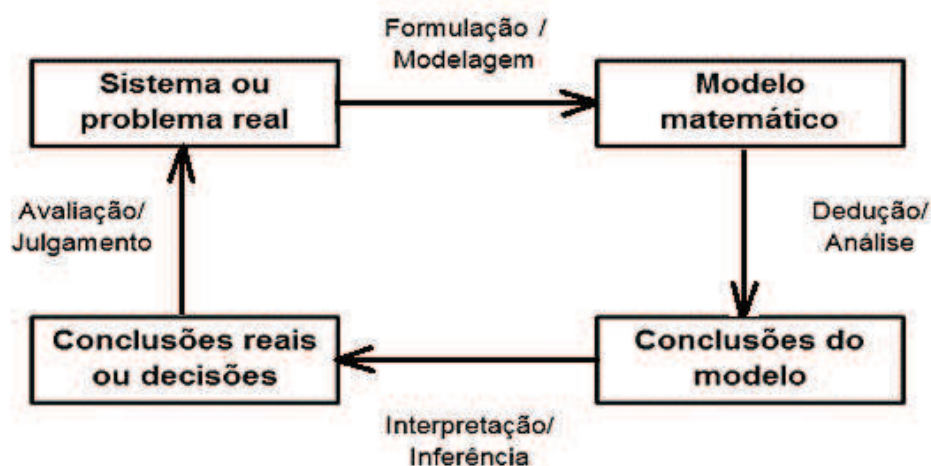


Figura 20 – Processo de modelagem

Fonte: (CAUCHICK MIGUEL, FLEURY, *et al.*, 2012), extraído de Arenales *et al.* (2007)

O processo de modelagem pode agregar duas grandes etapas:

Para os modelos de programação matemática considera-se um modelo analítico que representa uma alternativa ou escolha desse problema com variáveis de decisão, e procura por valores dessas variáveis denominada função objetivo (por exemplo, minimizar o custo total das redes de telecomunicações). A forma geral de um modelo de programação matemática é minimizar ou maximizar função objetivo, sujeito a restrições.

Os modelos de programação estocástica (otimização probabilística) consideram incertezas nos parâmetros do problema. Estes modelos são construídos para obter uma solução que maximize, por exemplo, a probabilidade de atendimento das demandas ou a probabilidade de utilizar todas as rotas (*links*) estabelecidas no planejamento de redes de telecomunicações. A Figura 21 apresenta um fluxograma simplificado para a metodologia aplicada.

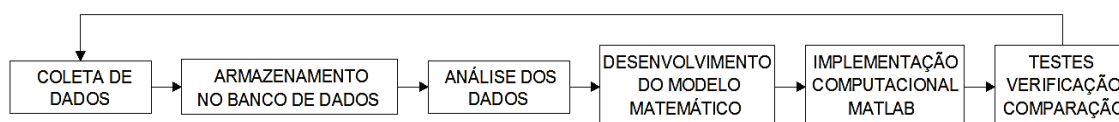


Figura 21 – Fluxograma simplificado da metodologia aplicada

3.2. PROCEDIMENTOS METODOLÓGICOS

Os procedimentos metodológicos utilizados para a realização da pesquisa tiveram o propósito de cumprir todos os objetivos gerais e específicos conforme apresentados no item 1.2 e 1.2.1 e foram divididos em quatro etapas:

Primeira etapa: Foi realizada a pesquisa bibliográfica para entender o problema e iniciar o desenvolvimento da produção científica. O início da pesquisa teve início à base de periódicos da CAPES e outras bases tais como: *IEEE Xplore Digital Library*, *SciELO Scientific Electronic Library Online*.

A visita na EMBRATEL, conforme carta de apresentação em anexo (ANEXO A) desta dissertação, contribuiu para o conhecimento de novas tecnologias e topologias aplicadas em sistemas de telecomunicações. Foi possível verificar os tipos de equipamentos e como os sinais digitais estão sendo transmitidos, bem como a velocidade empregada atualmente.

Segunda etapa: Teve início a pesquisa e estudos referentes à biblioteca SNDlib, conforme item 2.4. Para a rede PDH utilizou-se as instâncias disponibilizadas na biblioteca como base de dados para formar a codificação e as matrizes de incidência.

Terceira etapa: Elaboração do código-fonte com a ferramenta computacional MATLAB® e importação de dados armazenados em planilhas eletrônicas do software EXCEL para redes de pequeno porte.

Quarta etapa: Foi realizada uma nova estrutura de codificação com dados de redes de telecomunicações para redes de maior porte com maiores números de nós, *links* e demandas.

3.3. SELEÇÃO DE INSTÂNCIA SNDLIB

As instâncias da SNDlib são dadas em uma formulação arco-caminho da seguinte forma: $G = (V, E, D)$ onde V é o número de nós, E o número de arcos e D as demandas a serem escoadas. Para rotear as demandas é utilizada uma lista com os caminhos admissíveis para escoar determinada demanda.

As modularidades consideradas para cada arco da rede são disponibilizadas pela biblioteca de redes SNDlib e compreendem o custo existente para cada par de nós.

Foi desenvolvido um algoritmo genético para o problema de planejamento de redes de telecomunicações intitulado NDGA (*Network Design Genetic Algorithm*). Esse algoritmo foi elaborado pelo grupo de pesquisa do Laboratório de Estatísticas e Modelagem Matemática – LEMM da Pontifícia Universidade Católica de Goiás – PUC GO, composto por Emerson de Souza Campos, Victor Honorato Cunha, Lucas Carvalho Guimarães e Maria José Pereira Dantas para resolver as instâncias da biblioteca SNDlib com as seguintes características: as demandas e os *links* devem obedecer a um roteamento não direcional, a capacidade dos *links* é explícita, não possui custo fixo de instalação, as demandas podem ser escoadas apenas por um caminho, todos os caminhos existentes podem ser utilizados e não existe sobrevivência da rede. A sobrevivência da rede é um atributo que especifica o conceito de sobrevivência considerando um tipo de recurso e os valores para este atributo são:

- *No_Survivability*: Sem proteção de falhas;
- *One_Plus_One_Protection*: Uma proteção e uma proteção adicional;
- *Shared_Path_Protection*: Proteção compartilhada do arco (*link*);
- *Unrestricted_Flow_ReconFiguration*: Reconfiguração de fluxo irrestrito;

No caso mais simples, a rede não fornece nenhuma proteção contra falhas de nó e *link* e a opção a escolher deverá ser: *No_Survivability*. Caso contrário, vários recursos são considerados para proteger o tráfego contra falhas. (ZUSE INSTITUTE BERLIN (ZIB), 2005). A Tabela 7 resume a instância descrita.

Tabela 7 – Instância da Biblioteca de redes SNDlib (U-U-E-N-S-A-N-N).

Filtro de Modelo	
Modelo de Demanda: Demanda não direcional	(U)
Modelo de <i>Link</i> : <i>Links</i> não direcionais	(U)
Modelo de Capacidade do <i>Link</i> : Capacidades de <i>links</i> explícitos	(E)
Modelo de carga fixa: Sem custo de carga fixa	(N)
Modelo de roteamento: Caminho único	(S)
Modelo de caminho admissível: Todos os caminhos	(A)
Modelo de <i>hop-limit</i> : Ausência <i>hop limits</i>	(N)
Modelo de sobrevivência: Sem sobrevivência	(N)

Fonte: Zuse Institute Berlin (2005).

A formulação matemática com as funções objetivo e restrições foram definidas por (Orlowski, Wessály, Pióro, & Tomaszewski, 2010).

As demandas a serem atendidas devem obedecer cada modularidade que representa a capacidade de escoamento em canais E1. Foram realizadas simplificações considerando a instância U-U-E-N-S-A-N-N, descritas a seguir.

A Função Objetivo é apresentada na Equação (4) e representa o custo de implantação da rede.

$$\min \sum_{e \in \varepsilon} \sum_{t \in T_e} k_e^t \cdot y_e^t \quad (4)$$

Onde:

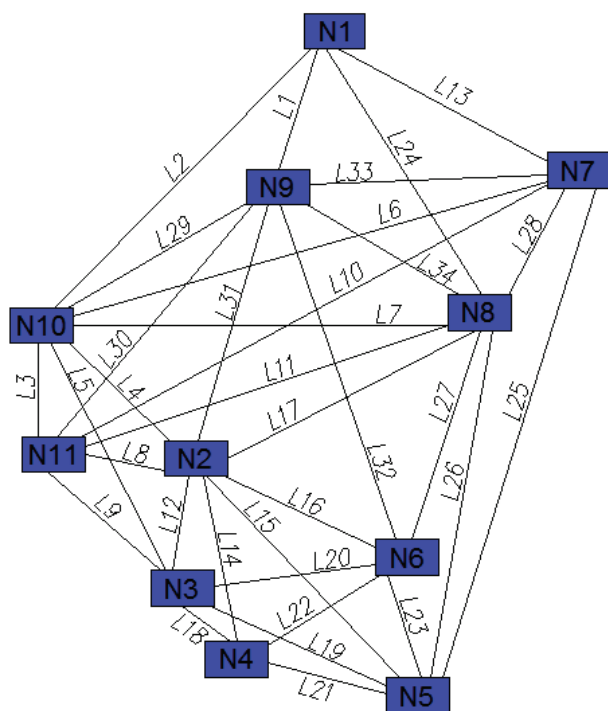
- ε : Conjunto de *links* de escoamento de demanda da rede, representa todas as possíveis conexões que podem ser feitas entre os nós da rede;
- e : Elemento do conjunto de *links* da rede [$e \in \varepsilon$];
- T_e : Conjunto das modularidades disponíveis para a instalação, sendo que cada *link* deve ser associado a uma modularidade dentro deste conjunto;
- t : Elemento do conjunto de modularidades da rede;
- k_e^t : Custo do módulo t instalado no *link* e , representa o custo de alocação de uma capacidade em um determinado *link*;
- y_e^t : Variável binária que representa a escolha da modularidade t instalada no *link* e . Decide se o custo k_e^t será contabilizado ou não;
- T_e : Conjunto das modularidades disponíveis para a instalação;
- Y_e : Fluxo de demanda em canais E1 no arco e .

A restrição de capacidade técnica é apresentada na Equação (5) e representa a capacidade instalada em um *link*. Esta capacidade deve ser maior que a soma das demandas que estão sendo escoadas através desse *link*, isto é, as demandas escoadas por cada arco que deve obedecer à capacidade do *link* alocado.

$$\sum_{e \in \varepsilon} \sum_{t \in T_e} t \cdot y_e^t \geq Y_e \quad (5)$$

O fluxo máximo de demanda que pode passar por um arco é determinado pela máxima modularidade disponível (restrição de capacidade técnica). Dessa forma, caso a soma das demandas no arco ultrapasse esse valor, a solução é dita inactível, visto que não existem módulos a serem instalados.

Para a apresentação e validação do NDGA foi escolhida uma rede PDH (*Plesiochronous Digital Hierarchy*) composta de 11 nós (vértices) e 34 *links* (arestas), conforme Figura 22.



Legenda:

N_i – Nós de demanda;

L_i – *Link* não-direcional.

Figura 22 – Rede Candidata para a instância PDH – Topologia.
Fonte: SNDlib, Zuse Institute Berlin (2005).

No total, 24 demandas previstas devem ser atendidas integralmente, conforme Tabela 8:

Tabela 8 – Valores para cada demanda prevista da rede PDH

Demanda Prevista	Nó (saída)	Nó (chegada)	Valor médio em canais E1
D1	N9	N2	384
D2	N3	N4	215
D3	N7	N8	212
D4	N10	N3	197
D5	N9	N11	160
D6	N2	N8	160
D7	N2	N5	144
D8	N1	N7	138
D9	N3	N5	124
D10	N6	N8	115
D11	N1	N10	115

D12	N10	N11	364
D13	N5	N8	108
D14	N4	N5	105
D15	N9	N8	100
D16	N11	N3	99
D17	N5	N6	95
D18	N2	N3	278
D19	N11	N2	266
D20	N9	N10	263
D21	N1	N9	258
D22	N9	N7	247
D23	N2	N6	237
D24	N2	N4	237

Fonte: SNDlib, (ZUSE INSTITUTE BERLIN (ZIB), 2005).

O custo total da rede apresentado na Equação 4 é calculado com base nos custos modulares apresentados na Tabela 9, que mostra o custo de instalação de cada modularidade em canais E1 (30, 480 e 1920) em relação a cada *link*.

Tabela 9 – Dados da rede PDH

<i>Links</i>	Nós (Saída / Chegada)		Custo do módulo em canais E1		
			30	480	1920
L1	N1	N9	88.933	691.901	1.625.091
L2	N1	N10	149.283	988.231	2.498.805
L3	N10	N11	59.431	530.267	1.162.431
L4	N10	N2	106.353	781.681	1.886.283
L5	N10	N3	129.408	892.606	2.215.230
L6	N10	N7	185.588	1.162.906	3.016.802
L7	N10	N8	164.388	1.060.906	2.714.322
L8	N11	N2	94.297	721.289	1.709.211
L9	N11	N3	114.038	818.656	1.995.932
L10	N11	N7	190.093	1.184.581	3.081.079
L11	N11	N8	164.918	1.063.456	2.721.884
L12	N2	N3	84.016	664.962	1.547.981
L13	N1	N7	121.988	856.906	2.109.362
L14	N2	N4	107.148	785.506	1.897.626
L15	N2	N5	139.743	942.331	2.362.689
L16	N2	N6	113.773	817.381	1.992.151
L17	N2	N8	136.828	928.306	2.321.098

L18	N3	N4	70.159	589.043	1.330.671
L19	N3	N5	124.638	869.656	2.147.172
L20	N3	N6	112.978	813.556	1.980.808
L21	N4	N5	103.438	767.656	1.844.692
L22	N4	N6	102.643	763.831	1.833.349
L23	N5	N6	91.168	704.146	1.660.141
L24	N1	N8	132.588	907.906	2.260.602
L25	N5	N7	187.973	1.174.381	3.050.831
L26	N5	N8	151.933	1.000.981	2.536.615
L27	N6	N8	119.603	845.431	2.075.333
L28	N7	N8	93.403	716.391	1.695.191
L29	N9	N10	121.193	853.081	2.098.019
L30	N9	N11	130.203	896.431	2.226.573
L31	N9	N2	124.638	869.656	2.147.172
L32	N9	N6	150.343	993.331	2.513.929
L33	N9	N7	125.698	874.756	2.162.296
L34	N9	N8	117.218	833.956	2.041.304

Fonte: SNDlib (ZUSE INSTITUTE BERLIN (ZIB), 2005)

O custo de instalação de um mesmo módulo em *links* distintos não é igual, pois além do custo do equipamento o custo com fibra óptica também é contabilizado. Sendo assim, o custo do *link* varia proporcionalmente ao seu comprimento.

Para a segunda etapa de testes e validação do método proposto, foi escolhida uma instância da rede ta1 de tecnologia SDH. Como pode ser observado na Figura 23 a rede da biblioteca SNDlib possui 34 nós e 55 *links*. A rede ta1 (SDH) possui alguns nós (N4, N7, N13, N16 e N17) que representam um central tandem que não geram ou recebem uma demanda, podendo apenas transmiti-las. Para esta instância a biblioteca SNDlib, possui 396 demandas previstas que devem ser atendidas integralmente, conforme Tabelas 10, 11, 12 e 13:

Legenda:

N_i - Nós de demanda;

L_i - Link não-direcional.

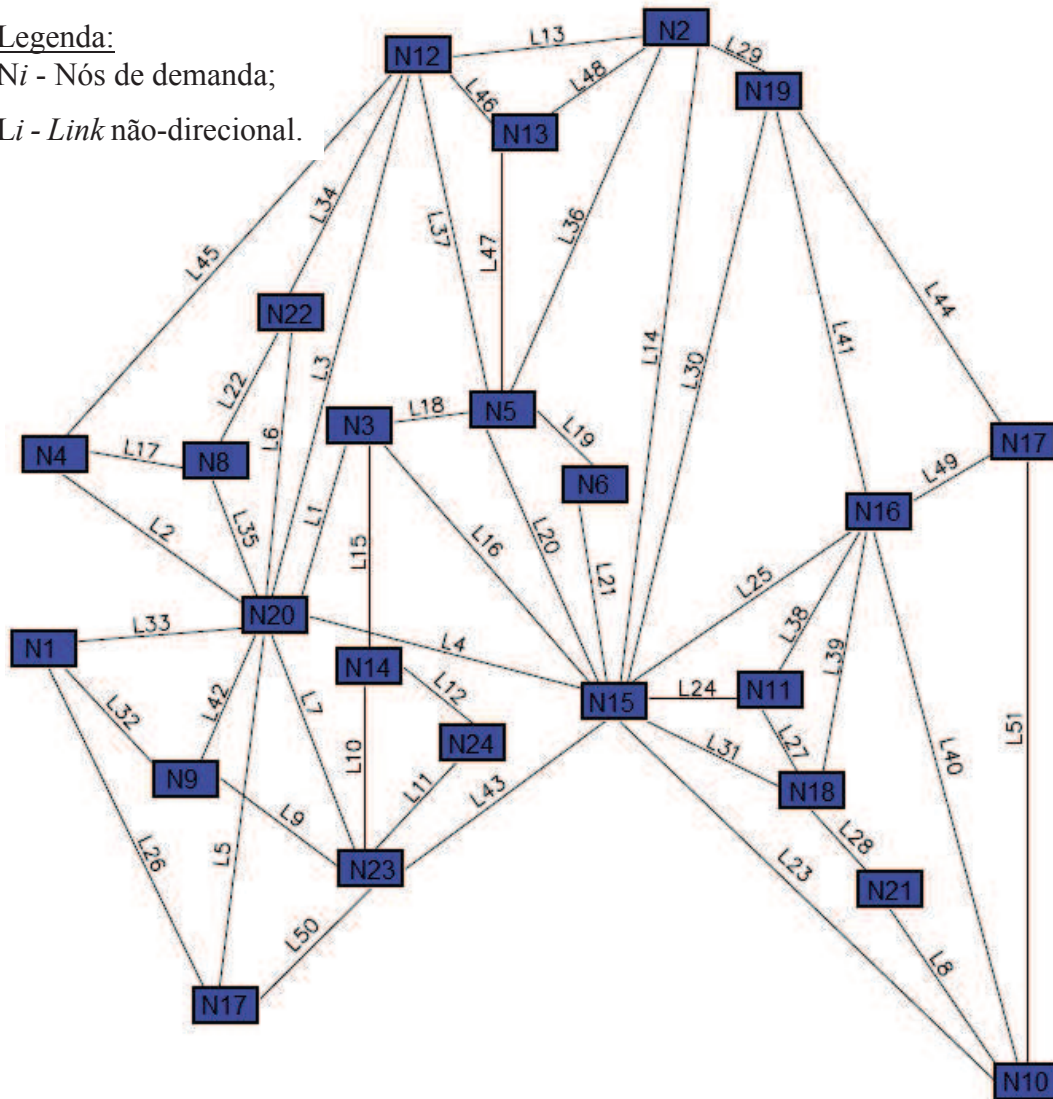


Figura 23 – Topologia de rede ta1 (SDH)
 Fonte: (ZUSE INSTITUTE BERLIN (ZIB), 2005)

Tabela 10 – Valores de demandas previstas da rede SDH (D1 a D102)

Demanda Prevista	Nó (saída)	Nó (chegada)	Valor médio em canais E1	Demanda Prevista	Nó (saída)	Nó (chegada)	Valor médio em canais E1
D1	N2	N15	819.678	D52	N11	N18	7.646
D2	N1	N20	192.434	D53	N5	N3	7.646
D3	N8	N20	15.292	D54	N3	N22	7.646
D4	N23	N3	15.120	D55	N3	N5	7.646
D5	N3	N23	15.120	D56	N2	N2	7.280
D6	N5	N15	14.934	D57	N15	N2	141.444
D7	N18	N10	14.160	D58	N2	N18	7.280
D8	N10	N18	14.160	D59	N15	N5	7.094
D9	N19	N15	14.001	D60	N20	N8	7.040
D10	N22	N8	11.469	D61	N3	N2	6.800
D11	N20	N12	11.469	D62	N2	N3	6.800
D12	N20	N8	11.469	D63	N20	N9	6.640
D13	N14	N3	191.040	D64	N14	N9	6.560
D14	N20	N1	11.469	D65	N9	N14	6.560
D15	N15	N9	11.469	D66	N15	N19	6.081
D16	N15	N6	11.469	D67	N20	N22	6.080
D17	N9	N15	11.469	D68	N15	N6	134.554
D18	N8	N22	11.469	D69	N10	N2	6.000
D19	N6	N15	11.469	D70	N3	N14	6.000
D20	N12	N1	10.465	D71	N2	N10	6.000
D21	N1	N12	10.465	D72	N6	N3	5.760
D22	N24	N15	9.921	D73	N3	N6	5.760
D23	N15	N24	9.921	D74	N20	N11	5.600
D24	N15	N3	179.672	D75	N18	N14	5.600
D25	N20	N18	9.681	D76	N14	N18	5.600
D26	N18	N20	9.681	D77	N11	N20	5.600
D27	N24	N9	9.200	D78	N3	N1	5.440
D28	N9	N24	9.200	D79	N20	N3	133.798
D29	N12	N2	9.029	D80	N1	N3	5.440
D30	N2	N12	9.029	D81	N20	N2	5.360
D31	N18	N3	8.960	D82	N18	N12	5.360
D32	N3	N18	8.960	D83	N12	N18	5.360
D33	N23	N18	8.880	D84	N2	N20	5.360
D34	N18	N23	8.880	D85	N10	N3	5.200
D35	N3	N15	179.672	D86	N3	N10	5.200
D36	N24	N14	8.640	D87	N20	N10	5.120
D37	N14	N24	8.640	D88	N11	N3	5.120
D38	N20	N14	8.400	D89	N10	N20	5.120
D39	N14	N20	8.400	D90	N3	N20	133.798
D40	N22	N8	8.320	D91	N3	N11	5.120
D41	N8	N22	8.320	D92	N24	N23	5.040
D42	N23	N1	8.160	D93	N23	N24	5.040
D43	N1	N23	8.160	D94	N11	N10	4.800
D44	N23	N12	8.000	D95	N10	N11	4.800
D45	N12	N23	8.000	D96	N23	N10	4.320
D46	N9	N20	151.520	D97	N10	N23	4.320
D47	N22	N20	7.646	D98	N9	N3	4.320
D48	N22	N3	7.646	D99	N3	N9	4.320
D49	N19	N15	7.646	D100	N18	N6	4.080
D50	N18	N11	7.646	D101	N23	N15	125.964
D51	N15	N19	7.646	D102	N6	N18	4.080

Fonte: SNDlib, (ZUSE INSTITUTE BERLIN (ZIB), 2005).

Tabela 11 – Valores de demandas previstas da rede SDH (D103 a D196)

Demanda Prevista	Nó (saída)	Nó (chegada)	Valor médio em canais	Demanda Prevista	Nó (saída)	Nó (chegada)	Valor médio em canais
D103	N20	N6	3.840	D150	N12	N6	2.480
D104	N6	N20	3.840	D151	N10	N21	2.480
D105	N20	N22	3.823	D152	N6	N12	2.480
D106	N15	N14	3.823	D153	N12	N11	2.400
D107	N14	N15	3.823	D154	N11	N12	2.400
D108	N3	N2	3.823	D155	N6	N2	2.400
D109	N2	N3	3.823	D156	N2	N6	2.400
D110	N23	N6	3.680	D157	N15	N20	122.097
D111	N6	N23	3.680	D158	N14	N1	2.240
D112	N18	N15	629.120	D159	N2	N1	2.240
D113	N15	N23	125.964	D160	N1	N14	2.240
D114	N23	N11	3.600	D161	N1	N2	2.240
D115	N23	N2	3.600	D162	N20	N24	2.000
D116	N18	N9	3.600	D163	N14	N10	2.000
D117	N11	N23	3.600	D164	N14	N2	2.000
D118	N9	N18	3.600	D165	N10	N14	2.000
D119	N2	N23	3.600	D166	N2	N14	2.000
D120	N11	N2	3.360	D167	N19	N2	1.920
D121	N2	N11	3.360	D168	N15	N10	119.842
D122	N12	N10	3.200	D169	N2	N19	1.920
D123	N10	N12	3.200	D170	N11	N1	1.840
D124	N15	N11	125.624	D171	N8	N3	1.840
D125	N22	N12	3.120	D172	N3	N8	1.840
D126	N22	N3	3.120	D173	N1	N11	1.840
D127	N14	N12	3.120	D174	N23	N8	1.760
D128	N12	N22	3.120	D175	N14	N6	1.760
D129	N12	N14	3.120	D176	N9	N6	1.760
D130	N3	N22	3.120	D177	N8	N23	1.760
D131	N11	N6	3.040	D178	N6	N14	1.760
D132	N6	N11	3.040	D179	N15	N3	119.038
D133	N18	N1	2.880	D180	N6	N9	1.760
D134	N1	N18	2.880	D181	N9	N2	1.680
D135	N21	N15	123.168	D182	N2	N9	1.680
D136	N22	N18	2.800	D183	N22	N2	1.600
D137	N18	N22	2.800	D184	N10	N9	1.600
D138	N14	N11	2.800	D185	N9	N10	1.600
D139	N12	N8	2.800	D186	N5	N3	1.600
D140	N11	N14	2.800	D187	N3	N5	1.600
D141	N8	N12	2.800	D188	N2	N22	1.600
D142	N9	N1	2.720	D189	N12	N9	1.520
D143	N1	N9	2.720	D190	N3	N15	119.038
D144	N10	N6	2.560	D191	N10	N1	1.520
D145	N6	N10	2.560	D192	N9	N12	1.520
D146	N20	N15	122.097	D193	N1	N10	1.520
D147	N23	N22	2.480	D194	N21	N11	1.440
D148	N22	N23	2.480	D195	N18	N8	1.440
D149	N21	N10	2.480	D196	N11	N21	1.440

Fonte: SNDlib, (ZUSE INSTITUTE BERLIN (ZIB), 2005).

Tabela 12 – Valores de demandas previstas da rede SDH (D197 a D309)

Demanda Prevista	Nó (saída)	Nó (chegada)	Valor médio em canais	Demanda Prevista	Nó (saída)	Nó (chegada)	Valor médio em canais
D197	N8	N18	1.440	D258	N11	N24	640
D198	N22	N10	1.360	D259	N8	N1	640
D199	N10	N22	1.360	D260	N1	N8	640
D200	N21	N2	1.280	D261	N24	N1	560
D201	N23	N15	118.507	D262	N21	N1	560
D202	N2	N21	1.280	D263	N20	N5	560
D203	N21	N3	1.200	D264	N19	N18	560
D204	N11	N9	1.200	D265	N18	N19	560
D205	N9	N11	1.200	D266	N15	N12	79.069
D206	N6	N1	1.200	D267	N9	N8	560
D207	N3	N21	1.200	D268	N8	N9	560
D208	N1	N6	1.200	D269	N5	N20	560
D209	N23	N21	1.120	D270	N1	N24	560
D210	N21	N23	1.120	D271	N1	N21	560
D211	N19	N10	1.120	D272	N24	N2	480
D212	N15	N23	118.507	D273	N21	N9	480
D213	N10	N19	1.120	D274	N21	N6	480
D214	N22	N6	1.040	D275	N9	N21	480
D215	N22	N1	1.040	D276	N6	N21	480
D216	N6	N22	1.040	D277	N12	N15	79.069
D217	N1	N22	1.040	D278	N2	N24	480
D218	N21	N20	960	D279	N24	N10	400
D219	N20	N21	960	D280	N18	N5	400
D220	N12	N5	960	D281	N10	N24	400
D221	N5	N12	960	D282	N10	N8	400
D222	N24	N18	880	D283	N10	N5	400
D223	N23	N20	365.703	D284	N8	N10	400
D224	N10	N15	110.861	D285	N5	N18	400
D225	N24	N3	880	D286	N5	N10	400
D226	N22	N9	880	D287	N24	N6	320
D227	N21	N12	880	D288	N15	N14	78.634
D228	N18	N24	880	D289	N23	N5	320
D229	N12	N21	880	D290	N19	N3	320
D230	N9	N22	880	D291	N6	N24	320
D231	N3	N24	880	D292	N5	N23	320
D232	N24	N12	800	D293	N3	N19	320
D233	N22	N11	800	D294	N22	N21	240
D234	N12	N24	800	D295	N21	N22	240
D235	N22	N20	91.680	D296	N20	N19	240
D236	N11	N22	800	D297	N19	N20	240
D237	N11	N8	800	D298	N24	N8	160
D238	N8	N11	800	D299	N22	N24	160
D239	N8	N6	800	D300	N22	N5	160
D240	N6	N8	800	D301	N23	N14	76.456
D250	N6	N5	800	D302	N21	N24	160
D251	N5	N6	800	D303	N19	N9	160
D252	N24	N11	640	D304	N19	N6	160
D253	N22	N14	640	D305	N14	N8	160
D254	N21	N14	640	D306	N14	N5	160
D255	N15	N10	80.279	D307	N9	N19	160
D256	N14	N22	640	D308	N9	N5	160
D257	N14	N21	640	D309	N8	N24	160

Fonte: SNDlib, (ZUSE INSTITUTE BERLIN (ZIB), 2005).

Tabela 13 – Valores de demandas previstas da rede SDH (D10 a D396)

Demanda Prevista	Nó (saída)	Nó (chegada)	Valor médio em canais	Demanda Prevista	Nó (saída)	Nó (chegada)	Valor médio em canais
D310	N8	N14	160	D354	N22	N15	39.282
D311	N6	N19	160	D355	N15	N22	39.282
D312	N14	N23	76.456	D356	N23	N2	36.532
D313	N5	N22	160	D357	N2	N23	36.532
D314	N5	N14	160	D358	N23	N20	34.406
D315	N5	N9	160	D359	N3	N1	30.583
D316	N5	N2	160	D360	N1	N3	30.583
D317	N2	N5	160	D361	N20	N12	29.749
D318	N24	N5	80	D362	N24	N20	29.520
D319	N23	N19	80	D363	N24	N23	26.760
D320	N21	N19	80	D364	N12	N20	236.431
D321	N19	N23	80	D365	N23	N24	26.760
D322	N19	N21	80	D366	N21	N15	26.760
D323	N11	N15	68.811	D367	N12	N3	26.080
D324	N19	N11	80	D368	N3	N12	26.080
D325	N19	N1	80	D369	N20	N23	24.469
D326	N11	N19	80	D370	N23	N9	23.025
D327	N11	N5	80	D371	N9	N23	23.025
D328	N5	N24	80	D372	N15	N12	22.937
D329	N5	N11	80	D373	N12	N15	22.937
D330	N1	N19	80	D374	N20	N3	22.865
D331	N10	N15	358.152	D375	N15	N18	233.920
D332	N18	N15	64.988	D376	N3	N20	22.865
D333	N12	N3	64.988	D377	N23	N14	21.025
D334	N3	N12	64.988	D378	N14	N23	21.025
D335	N15	N2	60.789	D379	N23	N1	19.114
D336	N15	N9	59.896	D380	N15	N21	19.114
D337	N9	N15	59.896	D381	N12	N20	19.114
D338	N15	N18	57.342	D382	N12	N2	19.114
D339	N9	N20	57.342	D383	N2	N12	19.114
D340	N20	N15	55.646	D384	N1	N23	19.114
D341	N15	N20	55.646	D385	N21	N18	18.210
D342	N11	N15	346.370	D386	N2	N15	206.431
D343	N23	N3	53.520	D387	N18	N21	18.210
D344	N15	N11	53.520	D388	N15	N8	18.070
D345	N3	N23	53.520	D389	N8	N15	18.070
D346	N15	N21	51.573	D390	N20	N1	17.954
D347	N15	N1	48.047	D391	N18	N11	15.760
D348	N1	N15	48.047	D392	N11	N18	15.760
D349	N1	N20	48.001	D393	N21	N18	15.292
D350	N8	N20	44.240	D394	N20	N23	15.292
D351	N23	N9	42.051	D395	N18	N21	15.292
D352	N9	N23	42051	D396	N14	N3	15.292
D353	N6	N15	299.994				

Fonte: SNDlib, (ZUSE INSTITUTE BERLIN (ZIB), 2005).

O custo total da rede apresentado na Equação 4 é calculado com base nos custos modulares apresentados nas Tabelas 14, 15 e 16 que mostram o custo de instalação de cada modularidade em relação a cada *link*.

Tabela 14 – Dados da rede SDH para os módulos de 1 a 4

Link	Nó (saída)	Nó (chegada)	Custo do módulo em canais E1			
			Módulo 1 504.000	Módulo 2 1.008.000	Módulo 3 1.512.000	Módulo 4 2.016.000
L1	N20	N3	162.160,02	324.319,93	486.479,95	162.160,02
L2	N20	N4	280.094,58	560.188,97	840.283,55	280.094,58
L3	N20	N12	398.029,14	796.058,01	1.194.087,15	398.029,14
L4	N20	N15	987.701,94	1.975.403,21	2.963.105,15	987.701,94
L5	N17	N20	88.450,92	176.901,78	265.352,70	88.450,92
L6	N20	N22	1.105.636,50	2.211.272,25	3.316.908,75	1.105.636,50
L7	N20	N23	619.156,44	1.238.312,46	1.857.468,90	619.156,44
L8	N21	N10	309.578,22	619.156,23	928.734,45	309.578,22
L9	N23	N9	456.996,42	913.992,53	1.370.988,95	456.996,42
L10	N23	N14	280.094,58	560.188,97	840.283,55	280.094,58
L11	N23	N24	117.934,56	235.869,04	353.803,60	117.934,56
L12	N24	N14	176.901,84	353.803,56	530.705,40	176.901,84
L13	N2	N12	796.058,28	1.592.116,02	2.388.174,30	796.058,28
L14	N2	N15	884.509,20	1.769.017,80	2.653.527,00	884.509,20
L15	N3	N14	545.447,34	1.090.894,31	1.636.341,65	545.447,34
L16	N3	N15	840.283,74	1.680.566,91	2.520.850,65	840.283,74
L17	N4	N8	648.640,08	1.297.279,72	1.945.919,80	648.640,08
L18	N3	N5	398.029,14	796.058,01	1.194.087,15	398.029,14
L19	N5	N6	206.385,48	412.770,82	619.156,30	206.385,48
L20	N5	N15	456.996,42	913.992,53	1.370.988,95	456.996,42
L21	N6	N15	250.610,94	501.221,71	751.832,65	250.610,94
L22	N8	N22	191.643,66	383.287,19	574.930,85	191.643,66
L23	N10	N15	456.996,42	913.992,53	1.370.988,95	456.996,42
L24	N11	N15	368.545,50	737.090,75	1.105.636,25	368.545,50
L25	N15	N16	29.483,64	58.967,26	88.450,90	29.483,64
L26	N17	N1	353.803,68	707.607,12	1.061.410,80	353.803,68
L27	N18	N11	280.094,58	560.188,97	840.283,55	280.094,58
L28	N18	N21	280.094,58	560.188,97	840.283,55	280.094,58
L29	N2	N19	398.029,14	796.058,01	1.194.087,15	398.029,14
L30	N15	N19	486.480,06	972.959,79	1.459.439,85	486.480,06
L31	N15	N18	235.869,12	471.738,08	707.607,20	235.869,12
L32	N1	N9	943.476,48	1.886.952,32	2.830.428,80	943.476,48
L33	N1	N20	427.512,78	855.025,27	1.282.538,05	427.512,78
L34	N12	N22	751.832,82	1.503.665,13	2.255.497,95	751.832,82
L35	N8	N20	928.734,66	1.857.468,69	2.786.203,35	928.734,66
L36	N2	N5	781.316,46	1.562.632,39	2.343.948,85	781.316,46
L37	N5	N12	855.025,56	1.710.050,54	2.565.076,10	855.025,56
L38	N11	N16	339.061,86	678.123,49	1.017.185,35	339.061,86
L39	N18	N16	221.127,30	442.254,45	663.381,75	221.127,30
L40	N16	N10	427.512,78	855.025,27	1.282.538,05	427.512,78
L41	N16	N19	471.738,24	943.476,16	1.415.214,40	471.738,24
L42	N9	N20	1.076.152,86	2.152.304,99	3.228.457,85	1.076.152,86
L43	N15	N23	1.606.858,38	3.213.715,67	4.820.574,05	1.606.858,38
L44	N7	N19	265.352,76	530.705,34	796.058,10	265.352,76
L45	N4	N12	117.934,56	235.869,04	353.803,60	117.934,56
L46	N12	N13	442.254,60	884.508,90	1.326.763,50	442.254,60
L47	N5	N13	427.512,78	855.025,27	1.282.538,05	427.512,78
L48	N2	N13	368.545,50	737.090,75	1.105.636,25	368.545,50
L49	N7	N16	206.385,48	412.770,82	619.156,30	206.385,48
L50	N17	N23	530.705,52	1.061.410,68	1.592.116,20	530.705,52
L51	N10	N7	221.127,30	442.254,45	663.381,75	221.127,30

Fonte: SNDlib (ZUSE INSTITUTE BERLIN (ZIB), 2005)

Tabela 15 – Dados da rede SDH para os módulos de 5 a 8

Link	Nó (saída)	Nó (chegada)	Custo do módulo em canais E1			
			Módulo 5 2.520.000	Módulo 6 3.024.000	Módulo 7 4.032.000	Módulo 8 4.536.000
L1	N20	N3	324.319,93	486.479,95	324.319,93	486.479,95
L2	N20	N4	560.188,97	840.283,55	560.188,97	840.283,55
L3	N20	N12	796.058,01	1.194.087,15	796.058,01	1.194.087,15
L4	N20	N15	1.975.403,21	2.963.105,15	1.975.403,21	2.963.105,15
L5	N17	N20	176.901,78	265.352,70	176.901,78	265.352,70
L6	N20	N22	2.211.272,25	3.316.908,75	2.211.272,25	3.316.908,75
L7	N20	N23	1.238.312,46	1.857.468,90	1.238.312,46	1.857.468,90
L8	N21	N10	619.156,23	928.734,45	619.156,23	928.734,45
L9	N23	N9	913.992,53	1.370.988,95	913.992,53	1.370.988,95
L10	N23	N14	560.188,97	840.283,55	560.188,97	840.283,55
L11	N23	N24	235.869,04	353.803,60	235.869,04	353.803,60
L12	N24	N14	353.803,56	530.705,40	353.803,56	530.705,40
L13	N2	N12	1.592.116,02	2.388.174,30	1.592.116,02	2.388.174,30
L14	N2	N15	1.769.017,80	2.653.527,00	1.769.017,80	2.653.527,00
L15	N3	N14	1.090.894,31	1.636.341,65	1.090.894,31	1.636.341,65
L16	N3	N15	1.680.566,91	2.520.850,65	1.680.566,91	2.520.850,65
L17	N4	N8	1.297.279,72	1.945.919,80	1.297.279,72	1.945.919,80
L18	N3	N5	796.058,01	1.194.087,15	796.058,01	1.194.087,15
L19	N5	N6	412.770,82	619.156,30	412.770,82	619.156,30
L20	N5	N15	913.992,53	1.370.988,95	913.992,53	1.370.988,95
L21	N6	N15	501.221,71	751.832,65	501.221,71	751.832,65
L22	N8	N22	383.287,19	574.930,85	383.287,19	574.930,85
L23	N10	N15	913.992,53	1.370.988,95	913.992,53	1.370.988,95
L24	N11	N15	737.090,75	1.105.636,25	737.090,75	1.105.636,25
L25	N15	N16	58.967,26	88.450,90	58.967,26	88.450,90
L26	N17	N1	707.607,12	1.061.410,80	707.607,12	1.061.410,80
L27	N18	N11	560.188,97	840.283,55	560.188,97	840.283,55
L28	N18	N21	560.188,97	840.283,55	560.188,97	840.283,55
L29	N2	N19	796.058,01	1.194.087,15	796.058,01	1.194.087,15
L30	N15	N19	972.959,79	1.459.439,85	972.959,79	1.459.439,85
L31	N15	N18	471.738,08	707.607,20	471.738,08	707.607,20
L32	N1	N9	1.886.952,32	2.830.428,80	1.886.952,32	2.830.428,80
L33	N1	N20	855.025,27	1.282.538,05	855.025,27	1.282.538,05
L34	N12	N22	1.503.665,13	2.255.497,95	1.503.665,13	2.255.497,95
L35	N8	N20	1.857.468,69	2.786.203,35	1.857.468,69	2.786.203,35
L36	N2	N5	1.562.632,39	2.343.948,85	1.562.632,39	2.343.948,85
L37	N5	N12	1.710.050,54	2.565.076,10	1.710.050,54	2.565.076,10
L38	N11	N16	678.123,49	1.017.185,35	678.123,49	1.017.185,35
L39	N18	N16	442.254,45	663.381,75	442.254,45	663.381,75
L40	N16	N10	855.025,27	1.282.538,05	855.025,27	1.282.538,05
L41	N16	N19	943.476,16	1.415.214,40	943.476,16	1.415.214,40
L42	N9	N20	2.152.304,99	3.228.457,85	2.152.304,99	3.228.457,85
L43	N15	N23	3.213.715,67	4.820.574,05	3.213.715,67	4.820.574,05
L44	N7	N19	530.705,34	796.058,10	530.705,34	796.058,10
L45	N4	N12	235.869,04	353.803,60	235.869,04	353.803,60
L46	N12	N13	884.508,90	1.326.763,50	884.508,90	1.326.763,50
L47	N5	N13	855.025,27	1.282.538,05	855.025,27	1.282.538,05
L48	N2	N13	737.090,75	1.105.636,25	737.090,75	1.105.636,25
L49	N7	N16	412.770,82	619.156,30	412.770,82	619.156,30
L50	N17	N23	1.061.410,68	1.592.116,20	1.061.410,68	1.592.116,20
L51	N10	N7	442.254,45	663.381,75	442.254,45	663.381,75

Fonte: SNDlib (ZUSE INSTITUTE BERLIN (ZIB), 2005)

Tabela 16 – Dados da rede SDH para os módulos de 9 a 11

<i>Link</i>	Nó (saída)	Nó (chegada)	Custo do módulo em canais E1		
			Módulo 9 5.040.000	Módulo 10 6.048.000	Módulo 11 11.088.000
L1	N20	N3	648.639,86	486.479,95	1.621.599,76
L2	N20	N4	1.120.377,94	840.283,55	2.800.945,04
L3	N20	N12	1.592.116,02	1.194.087,15	3.980.290,32
L4	N20	N15	3.950.806,42	2.963.105,15	9.877.016,72
L5	N17	N20	353.803,56	265.352,70	884.508,96
L6	N20	N22	4.422.544,50	3.316.908,75	11.056.362,00
L7	N20	N23	2.476.624,92	1.857.468,90	6.191.562,72
L8	N21	N10	1.238.312,46	928.734,45	3.095.781,36
L9	N23	N9	1.827.985,06	1.370.988,95	4.569.962,96
L10	N23	N14	1.120.377,94	840.283,55	2.800.945,04
L11	N23	N24	471.738,08	353.803,60	1.179.345,28
L12	N24	N14	707.607,12	530.705,40	1.769.017,92
L13	N2	N12	3.184.232,04	2.388.174,30	7.960.580,64
L14	N2	N15	3.538.035,60	2.653.527,00	8.845.089,60
L15	N3	N14	2.181.788,62	1.636.341,65	5.454.471,92
L16	N3	N15	3.361.133,82	2.520.850,65	8.402.835,12
L17	N4	N8	2.594.559,44	1.945.919,80	6.486.399,04
L18	N3	N5	1.592.116,02	1.194.087,15	3.980.290,32
L19	N5	N6	825.541,64	619.156,30	2.063.854,24
L20	N5	N15	1.827.985,06	1.370.988,95	4.569.962,96
L21	N6	N15	1.002.443,42	751.832,65	2.506.108,72
L22	N8	N22	766.574,38	574.930,85	1.916.436,08
L23	N10	N15	1.827.985,06	1.370.988,95	4.569.962,96
L24	N11	N15	1.474.181,50	1.105.636,25	3.685.454,00
L25	N15	N16	117.934,52	88.450,90	294.836,32
L26	N17	N1	1.415.214,24	1.061.410,80	3.538.035,84
L27	N18	N11	1.120.377,94	840.283,55	2.800.945,04
L28	N18	N21	1.120.377,94	840.283,55	2.800.945,04
L29	N2	N19	1.592.116,02	1.194.087,15	3.980.290,32
L30	N15	N19	1.945.919,58	1.459.439,85	4.864.799,28
L31	N15	N18	943.476,16	707.607,20	2.358.690,56
L32	N1	N9	3.773.904,64	2.830.428,80	9.434.762,24
L33	N1	N20	1.710.050,54	1.282.538,05	4.275.126,64
L34	N12	N22	3.007.330,26	2.255.497,95	7.518.326,16
L35	N8	N20	3.714.937,38	2.786.203,35	9.287.344,08
L36	N2	N5	3.125.264,78	2.343.948,85	7.813.162,48
L37	N5	N12	3.420.101,08	2.565.076,10	8.550.253,28
L38	N11	N16	1.356.246,98	1.017.185,35	3.390.617,68
L39	N18	N16	884.508,90	663.381,75	2.211.272,40
L40	N16	N10	1.710.050,54	1.282.538,05	4.275.126,64
L41	N16	N19	1.886.952,32	1.415.214,40	4.717.381,12
L42	N9	N20	4.304.609,98	3.228.457,85	10.761.525,68
L43	N15	N23	6.427.431,34	4.820.574,05	16.068.579,44
L44	N7	N19	1.061.410,68	796.058,10	2.653.526,88
L45	N4	N12	471.738,08	353.803,60	1.179.345,28
L46	N12	N13	1.769.017,80	1.326.763,50	4.422.544,80
L47	N5	N13	1.710.050,54	1.282.538,05	4.275.126,64
L48	N2	N13	1.474.181,50	1.105.636,25	3.685.454,00
L49	N7	N16	825.541,64	619.156,30	2.063.854,24
L50	N17	N23	2.122.821,36	1.592.116,20	5.307.053,76
L51	N10	N7	648.639,86	663.381,75	2.211.272,40

A rede tal (SDH) foi fornecida pelo operador de redes Telekom Austria, sendo que os dados foram levemente modificados com finalidade de proteger as informações do operador (ORLOWSKI, WESSÄLY, *et al.*, 2010).

Nas tabelas 14 e 15 e 16 é possível observar que para um mesmo *link* existem custos iguais, apesar da diferença de modularidade existente. A igualdade é mantida para todos os *links* obedecendo o seguinte padrão: o custo do módulo 1 é igual ao custo do módulo 4, o custo 2 é igual ao custo 5, 7 e o custo 3 é o mesmo dos módulos 6, 8 e 10. Os demais custos (11 e 9) não apresentaram semelhanças.

Essa igualdade pode acontecer pela modificação dos dados do operador de rede, como também pode representar uma situação atípica de mercado. Devido a essa igualdade é possível instalar um módulo de maior capacidade pelo mesmo valor de um módulo menor, por exemplo, é mais vantajoso instalar o módulo 4 em detrimento do 1, pois ambos têm o mesmo custo com uma diferença de capacidade de quatro vezes.

Dessa forma, dos onze módulos disponíveis na instância da biblioteca são considerados para instalação os módulos 4, 7 e 10, economizando tempo de processamento.

3.4. NETWORK DESIGN GENETIC ALGORITHM (NDGA)

O algoritmo proposto NDGA resolve as instâncias da biblioteca de redes SNDlib para as tecnologias PDH e SDH. O código foi desenvolvido em MATLAB®, por ser uma linguagem científica destinada a cálculos matemáticos. O algoritmo implementado é descrito de acordo com o fluxograma da Figura 24.

Todo o processo do NDGA ocorre em dois ambientes: MATLAB® e EXCEL®. Todos os dados relativos aos nós, *links*, demandas, custo, caminhos, e parâmetros a serem definidos pelo usuário; são dispostos em uma planilha eletrônica, para que o MATLAB consiga processar esses dados é necessário que eles sejam importados para o ambiente de programação, e esse é o primeiro passo do algoritmo.

Feita a importação dos dados começa o processo de formação da população inicial do NDGA. Dessa forma, as primeiras propostas de soluções são formadas e a melhor delas é mantida através da rotina de elitismo. No próximo passo do algoritmo os indivíduos passam pela ação dos operadores genéticos de cruzamento, seleção e mutação para que as boas características dos indivíduos sejam trabalhadas. As rotinas de elitismo

e operadores genéticos são repetidas até que um número máximo de iterações, definida pelo usuário, seja atingido.

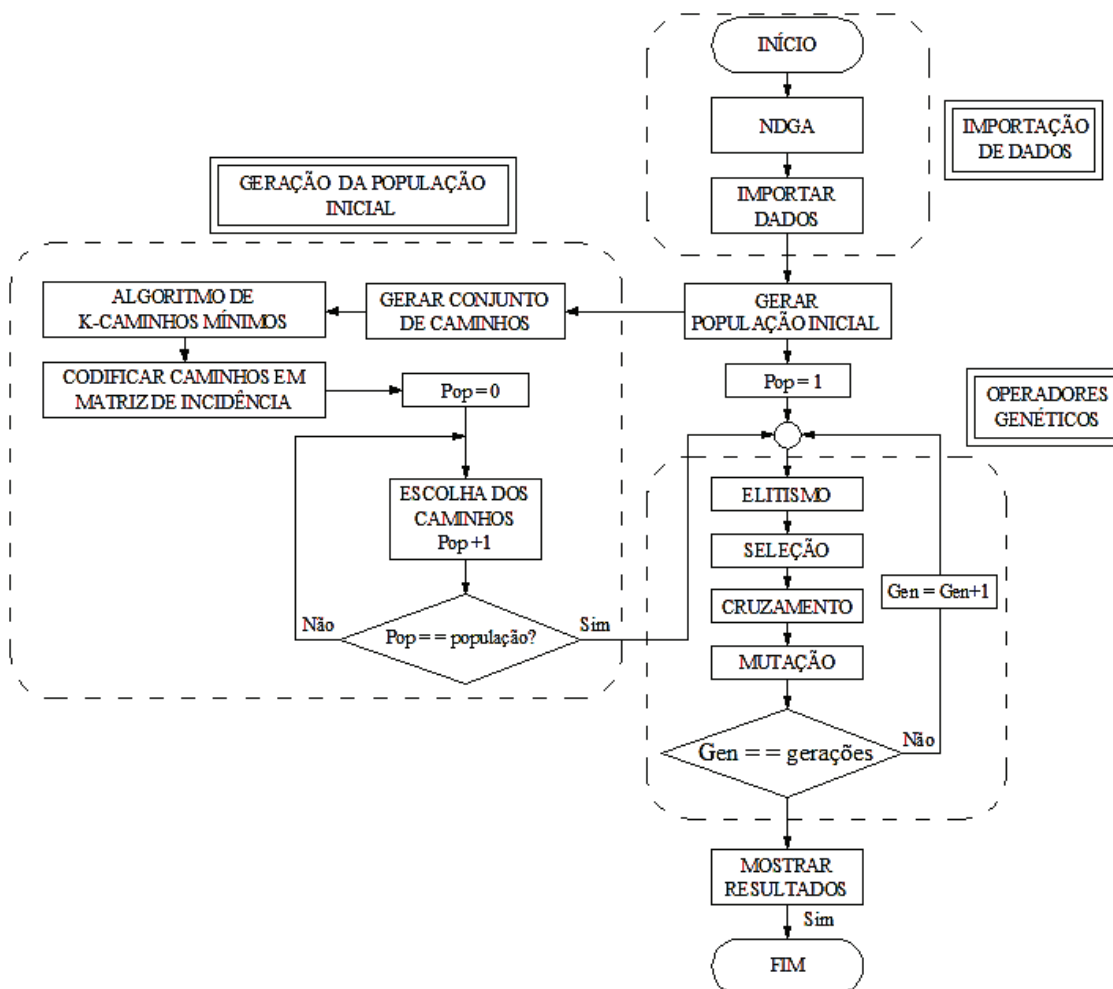


Figura 24 – Fluxograma algoritmo proposto - NDGA

Após todo o procedimento iterativo do NDGA os resultados são mostrados através de planilha eletrônica (exportação da solução encontrada), curva de convergência (mostra a evolução da solução) e a plotagem da rede otimizada (solução encontrada).

3.4.1. Codificação, Infactibilidade e População Inicial

A codificação da rede é feita através de uma matriz de incidência de ordem $N \times L$, onde N é o número de nós e L é o número de *links*. A matriz é composta de 0's e 1's pois os *links* existentes são bidirecionais, onde 0 representa ausência de *link* e 1 mostra a incidência do *link* em determinado nó.

Já as soluções da rede são uma matriz de ordem $D \times L$, onde D é o número de demandas a serem escoadas (Figura 25). Cada demanda deve ser escoada por um caminho

formado por um conjunto de *links* (do nó origem até o nó destino). Assim cada linha da matriz representa os *links* utilizados para escoar determinada demanda, sendo que a coluna correspondente ao *link* utilizado é preenchida com o número 1. Posteriormente os elementos da matriz que são iguais a um, são substituídos pelo valor correspondente a cada demanda. Matematicamente:

$$a_{dl} = \begin{cases} D_d, & \text{se } a_{dl} = 1 \\ 0, & \text{se } a_{dl} = 0 \end{cases}$$

Onde: d , número da demanda.

l , número do *link*.

a_{dl} , elemento da matriz correspondente ao indivíduo

D_d , valor da demanda d

Uma proposta de solução do problema já codificada pode ser vista da Figura 25. A linha da matriz que se encontra separada corresponde à soma de todas as demandas que utilizaram um determinado *link*. Se a soma for igual a zero, o *link* não é instalado, caso a soma seja maior que o limite técnico (modularidade disponível para instalação). Neste caso, o indivíduo é considerado ineficaz e descartado dando início à formação de um novo indivíduo.

	L1	L2	L3	L4	L5	...	L30	L31	L32	L33	L34
D1	0	0	0	0	0		0	384	0	0	0
D2	0	0	221	0	377		322	0	0	0	0
D3	0	0	0	195	0		0	0	0	212	212
⋮											
D22	0	0	124	0	0		222	0	0	247	0
D23	0	0	0	128	0		0	66	0	0	0
D24	0	0	0	0	302		0	0	0	0	0
	121	297	345	323	679		544	450	670	459	212

Figura 25 – Codificação das soluções

Para cada demanda existe uma lista de caminhos disponíveis, esses caminhos são criados utilizando-se o algoritmo de *k-caminhos mínimos*, pois como parte do custo resulta da distância de fibra óptica utilizada. Não é interessante que os caminhos (descritos como uma sequência de arcos) sejam longos. Assim, faz parte da modelagem, a escolha da quantidade de caminhos para cada demanda.

Para a formação do indivíduo, um desses caminhos deve ser selecionado aleatoriamente. Assim é gerado um número finito de indivíduos que correspondem a população inicial do NDGA. Como apresentado na Figura 26 a população pode ser

descrita e indexada (na implementação) como uma matriz de ordem $3 D \times L \times P$ sendo P o tamanho da população que é constante no decorrer das gerações.

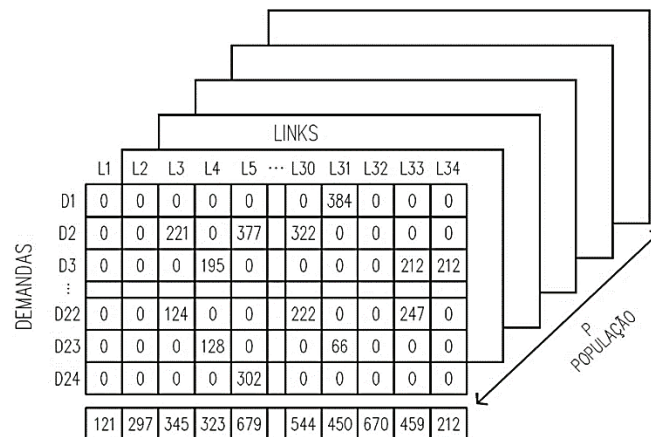


Figura 26 – Exemplo de população inicial

3.4.2. Aptidão e Seleção

O custo total da rede é utilizado como função de aptidão. Cada arco da rede possui um custo devido à fibra óptica e a modularidade do equipamento escolhido para transmissão de dados. A biblioteca SNDLib fornece uma tabela com o custo correspondente para cada arco da rede, dessa forma, caso um arco seja utilizado em uma solução, o seu custo é contabilizado (Equação 4).

A seleção dos indivíduos é feita pelo método do torneio. Dois indivíduos são selecionados aleatoriamente, os seus custos são comparados e a solução de menor custo é então selecionada para passar suas informações para a próxima geração por meio do *crossover*. Esse método é simples, porém eficiente, e não possui custo computacional excessivo, já que o custo é parte do indivíduo e não precisa ser calculado unicamente para a seleção (MILLER e GOLDBERG, 1995).

3.4.3. Crossover

No NDGA é utilizado o *crossover* de n pontos. Os pais que realizarão cruzamento são seccionados em n pontos aleatórios e têm suas informações genéticas trocadas. São gerados dois filhos, porém apenas o indivíduo de menor custo entra para a próxima geração (Figura 27). Os filhos gerados podem ainda ser infactíveis devido à capacidade máxima disponível de canais. Se isto acontecer, o processo é repetido novamente até que se consiga um indivíduo factível.

Este operador é controlado por uma probabilidade que define a presença ou ausência do operador. Caso não haja *crossover* o indivíduo de uma geração é apenas transferido para a próxima.

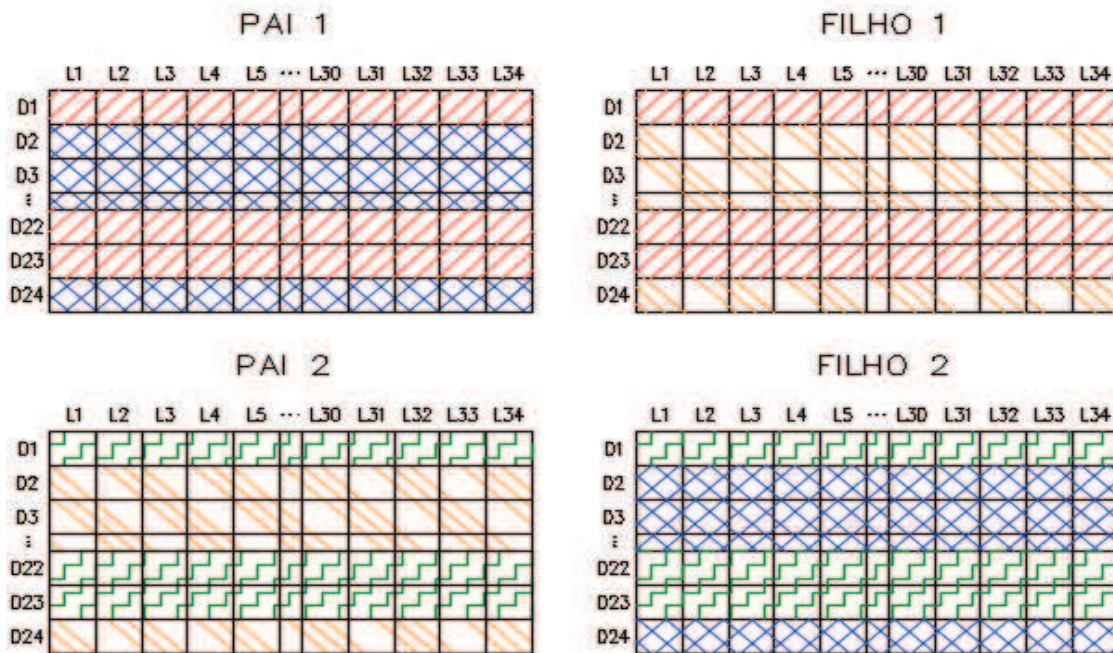


Figura 27 – *Crossover* de n pontos para o NDGA

3.4.4. Mutação

Para o NDGA, foi proposto um novo operador. Cada linha representa o caminho pelo qual uma demanda está sendo escoada e o processo de mutação consiste em trocar este caminho por outro aleatoriamente escolhido dentro do conjunto disponíveis de caminhos (Figura 28). O número de trocas a serem realizadas é parte dos parâmetros a serem definidos no NDGA.

	L1	L2	L3	L4	L5	...	L30	L31	L32	L33	L34
D1	/ / / / / / / / / / / /										
D2	0	0	221	0	377		322	0	0	0	0
D3	0	0	0	195	0		0	0	0	212	212
⋮											
D22	/ / / / / / / / / / / /										
D23	/ / / / / / / / / / / /										
D24	0	0	0	0	302		0	0	0	0	0

CAMINHO ALTERADO

Figura 28 – Mutação com a ocorrência de 03 trocas

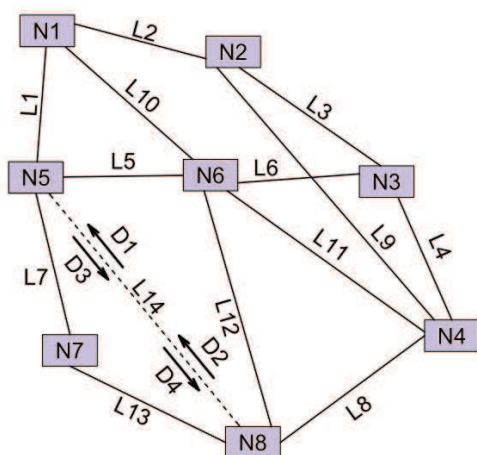
3.4.5. Elitismo

O elitismo implementado no NDGA se resume em manter a melhor solução encontrada em cada geração. Assim é possível garantir que o pior indivíduo de uma geração posterior seja no mínimo igual ao da anterior. A análise de convergência do NDGA é realizada com base no indivíduo elitista da última geração, ou seja, a rede otimizada corresponde ao indivíduo elitista da última geração.

3.5. ALGORITMO DE PÓS-OTIMIZAÇÃO

Na literatura, vários trabalhos aplicam o algoritmo de pós-otimização com o objetivo de obter melhores soluções. Belgacem, Charon e Hudry (2014). Descrevem um algoritmo de pós otimização para ser utilizado ao final do método heurístico (AG), afim de melhorar a solução encontrada.

A solução apresentada pelo NDGA pode ainda ser trabalhada com o algoritmo de pós-otimização baseado no rompimento dos *links* da rede otimizada. Um *link* faz parte do caminho de várias demandas (Figura 29). Dessa forma, a ideia consiste em “romper” um *link* e remanejar as demandas que o utilizava, por caminhos diferentes na rede. Caso a alteração de caminhos seja possível para todas as demandas a serem remanejadas, o *link* “rompido” é considerado dispensável e pode ser removido da rede, reduzindo assim o custo.



Legenda:

- N_i – Nós de demanda;
- L_i – *Link* não-direcional.
- - - - *Link* a ser retirado.

Figura 29 – Exemplo de rede de telecomunicações

Todo *link* que esteja presente na solução do NDGA passa pela simulação de rompimento. Como o objetivo da otimização é unicamente financeiro os *links* são organizados em ordem decrescente de custo. Dessa forma, existe uma maior probabilidade de que um *link* de alto custo possa ser retirado da solução final.

Após o “rompimento” do *link* são necessários que duas condições sejam satisfeitas para que o *link* seja realmente retirado da rede:

1. Exista no mínimo um caminho alternativo para cada uma das demandas que utilizam o *link* rompido;

2. Caso a primeira condição exista, as demandas são enviadas pelos seus novos caminhos, de forma que a capacidade já instalada não seja ultrapassada.

Caso as condições 1 ou 2 não sejam inteiramente satisfeitas, o *link* não é removido e o algoritmo segue para o próximo *link* de acordo com a ordem estabelecida. O processo é repetido até que todos os *links* sejam testados conforme fluxograma da Figura 30.

Como será apresentado na seção de resultados esse processo melhora o custo final da rede (objetivo da otimização), porém minimiza as folgas existentes. De um lado a rede é melhor aproveitada, de outro ela não permite atender novas demandas em um cenário futuro. Como o objetivo deste trabalho é a redução de custos, o método se apresenta adequado à definição da função objetivo.

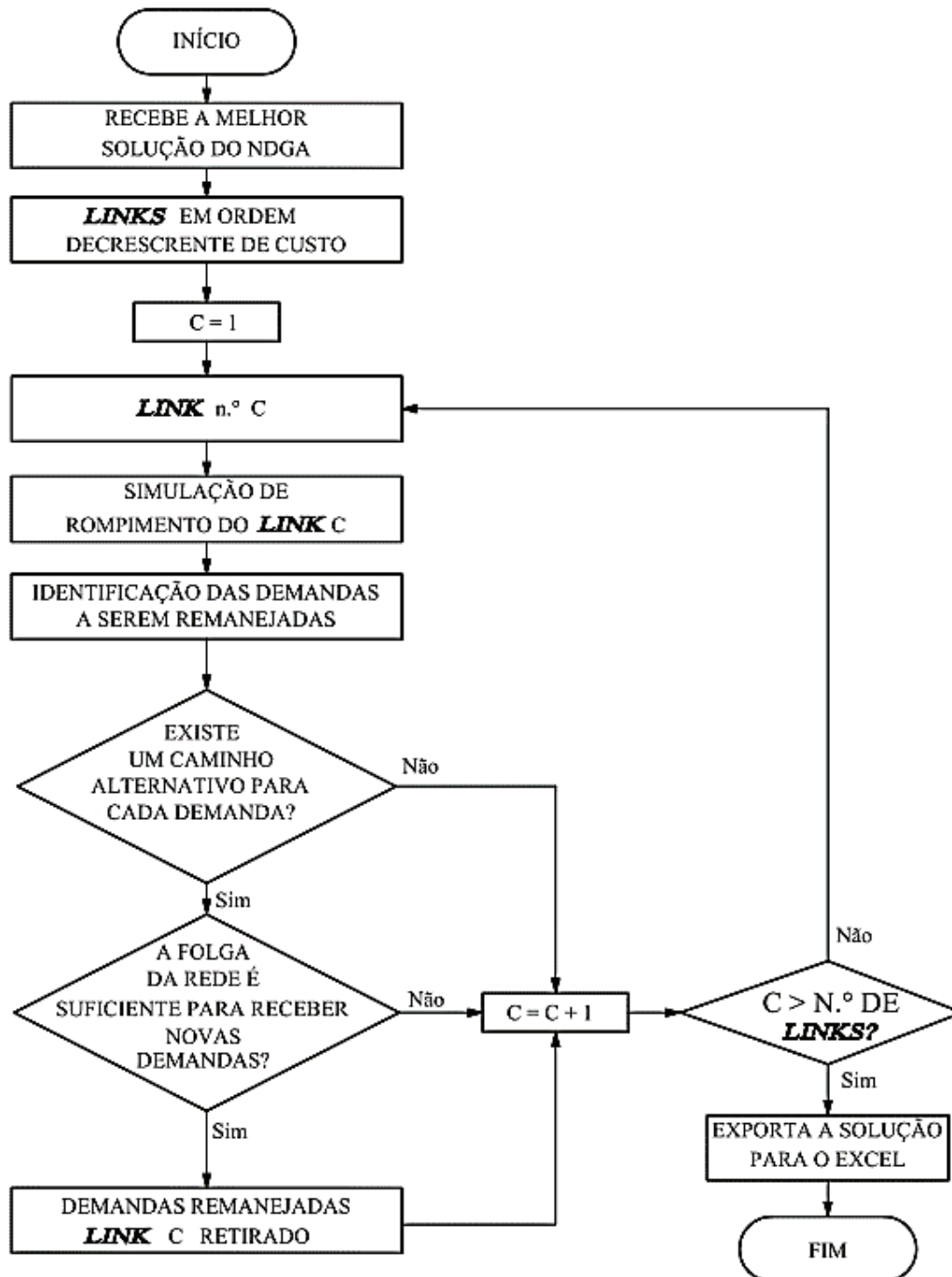


Figura 30 – Fluxograma após a otimização

CAPÍTULO IV

4. RESULTADOS E DISCUSSÕES

Nas seções subsequentes serão apresentados os resultados para a rede PDH utilizando o NDGA e depois combinando o NDGA com o algoritmo de pós otimização descrito na seção 3.5. A rede PDH é uma rede de pequeno porte que foi utilizada nos testes iniciais dos dois algoritmos. Logo após será apresentado o resultado do NDGA combinado com o algoritmo de pós-otimização aplicado na otimização da rede tal.

A descrição dos testes feitos é específica para cada rede e algoritmo. Sendo assim, a descrição e justificativa dos testes feitos é apresentada na seção correspondente. Os resultados encontrados pelos algoritmos desenvolvidos são comparados com os disponíveis na biblioteca SNDlib para a instância selecionada. Pelo fato de serem probabilísticos, os AGs podem não repetir os resultados, mesmo que os parâmetros genéticos sejam mantidos.

4.1. NDGA aplicado à rede PDH

Para os testes iniciais e análise individual, o NDGA é aplicado à rede PDH de 11 nós, 34 *links* e 24 demandas a serem escoadas. Primeiramente, foram realizados 100 testes para um total de 11 configurações de parâmetros diferentes (Tabela 17). O conjunto de testes para cada configuração é chamado de experimento, e o resultado final de cada experimento corresponde ao melhor resultado encontrado entre os testes realizados.

Tabela 17 – Configuração dos Experimentos

Experimentos	População	Gerações	<i>Crossover</i> (Probabilidade /Cortes)	<i>Mutação</i> (Probabilidade /Trocas)
1	100	100	0,9 / 5	0,06 / 2
2	200	100	0,9 / 5	0,06 / 2
3	300	100	0,9 / 5	0,06 / 2
4	400	100	0,9 / 5	0,06 / 2
5	400	100	0,9 / 5	0,06 / 2
6	100	100	0,9 / 5	0,06 / 5
7	200	200	0,9 / 5	0,06 / 2
8	300	300	0,9 / 5	0,07 / 2
9	400	400	0,9 / 5	0,08 / 2
10	500	500	0,9 / 5	0,08 / 2
11	1000	500	0,95 / 10	0,08 / 5

A escolha de cada parâmetro deve ser feita considerando as características e complexidade de cada problema especificamente, pois não existe um conjunto de parâmetros que produzem bons resultados para todos os tipos de problemas. Porém, existem valores que se tornaram clássicos na literatura, como por exemplo, probabilidade de crossover e mutação, pois em muitos problemas pequenas alterações nessas probabilidades em torno do valor clássico não alteram significativamente o desempenho do AG. Para o problema de redes aqui considerado o espaço de busca é vasto, de forma de que o tamanho da população e a quantidade máxima de gerações do NDGA interferem significativamente no resultado. Após vários testes, verificou-se o conjunto de parâmetros com bons resultados para a otimização da rede PDH. Desta forma, os experimentos foram configurados com valores para ter uma visualização do esforço computacional necessário para a otimização deste problema.

Os valores de aptidão de todos os indivíduos são conhecidos, porém a análise de convergência é feita através das informações do indivíduo mais apto. Como os indivíduos são avaliados de acordo com seu custo, esse dado é utilizado para gerar uma curva de convergência que mostra evolução da melhor solução no decorrer das gerações. O resultado final do NDGA é o indivíduo mais apto da última geração, e o resultado de cada experimento é o melhor teste realizado entre as 100 execuções.

A codificação escolhida foi adequada para o problema de modo, que nenhuma informação seja perdida no processo de codificação e decodificação dos cromossomos. A seleção e o *crossover* propostos foram eficientes no processo de troca de informações entre os indivíduos e também apresentou flexibilidade considerando que a quantidade de indivíduos da seleção e o número de cortes a serem realizados podem ser definidos.

A mutação foi fundamental para a solução do problema, pois o espaço de busca é extenso e multimodal, de forma que a mutação ameniza a tendência do AG a convergir para ótimos locais. Após alguns testes verificou-se a necessidade do uso de elitismo, pois boas soluções estavam sendo perdidas durante o processo evolutivo comprometendo a solução final encontrada pelo algoritmo.

Após comparações com soluções encontradas por outras técnicas de otimização, o AG desenvolvido apresentou redução significativa no custo da rede.

Nos relatórios da SNDlib são disponibilizadas seis informações principais: o custo da rede, o número de arcos utilizados, a capacidade total em canais E1 instalada na rede, a porcentagem utilizada dessa capacidade, o comprimento médio dos caminhos utilizados para escoar as demandas e o tempo gasto na otimização.

A Tabela 18 mostra os resultados disponíveis na biblioteca e os obtidos para os 11 experimentos realizados para fim de comparação. Enquanto que, nos resultados da SNDlib, dos 34 *links* disponíveis 13 foram instalados sendo 8 de 480E1 e 5 de 1920E1.

Para os experimentos realizados o NDGA utilizou 17 *links* com modularidades de 480E1. O maior uso de *links* pode gerar maiores custos, mas as modularidades de 1920E1 que interferem significativamente no custo não foram utilizadas, garantindo uma redução de custo de 15,15% nos experimentos 4, 9, 10 e 11, em comparação com a solução da biblioteca.

Este percentual representa uma diferença monetária de \$1.958.024,00, mas por outro lado, a capacidade utilizada de 72,48% está acima da biblioteca que apresentou um valor de 65,08%.

A capacidade utilizada em valores percentuais mostra a melhor utilização da capacidade instalada. Apesar de não ser o melhor valor encontrado, o experimento 6 tem maior flexibilidade em relação as folgas, utilizando apenas 64,66% da capacidade instalada. Dessa forma, o experimento 6 também pode ser considerado um bom resultado, pois tem maior folga e menor custo em relação a solução da SNDlib.

Tabela 18 – Resultados das simulações e da biblioteca SNDlib

Experimentos	Custo (\$)	Capacidade Instalada (Canais E1)	Capacidade Utilizada (%)	Tamanho médio dos caminhos (<i>links</i>)	Tempo (s)	
pdh ¹	14.883.708,00	13.440	65,08	2,00	3600	
Valores Simulados do algoritmo proposto NDGA	1	13.353.844,00	8.160	74,01	1,38	0,70
	2	13.398.469,00	8.160	67,11	1,33	1,40
	3	13.138.609,00	8.160	68,62	1,29	2,86
	4	12.925.684,00	8.160	72,48	1,38	3,00
	5	13.290.334,00	8.160	67,76	1,29	4,27
	6	13.738.754,00	9.600	64,66	1,46	0,88
	7	13.246.744,00	8.160	71,32	1,38	3,52
	8	12.966.484,00	8.160	70,67	1,38	8,39
	9	12.925.684,00	8.160	72,48	1,38	17,05
	10	12.925.684,00	8.160	72,48	1,38	18,97
	11	12.925.684,00	8.160	72,48	1,38	52,61

¹ Resultado da biblioteca SNDlib

O tempo gasto na otimização é importante tanto quanto o custo final da rede, pois na resolução de problemas ainda mais complexos é necessário que a otimização seja feita em tempo hábil, de acordo com as definições do usuário.

A biblioteca utiliza um modelo determinístico de resolução do problema. Caso o usuário deixe o algoritmo executar em sua totalidade, o método testa todas as soluções encontrando a melhor delas. Porém, os autores da biblioteca definem um tempo de otimização de uma hora para o modelo, justificando o fato de o algoritmo probabilístico ser melhor do que o determinístico.

Considerando unicamente o custo, a melhor solução encontrada foi a apresentada no experimento 4, que chegou ao menor tempo devido ao tamanho da população e número de gerações. O gráfico de convergência (Figura 31) mostra a evolução da solução no decorrer das gerações.

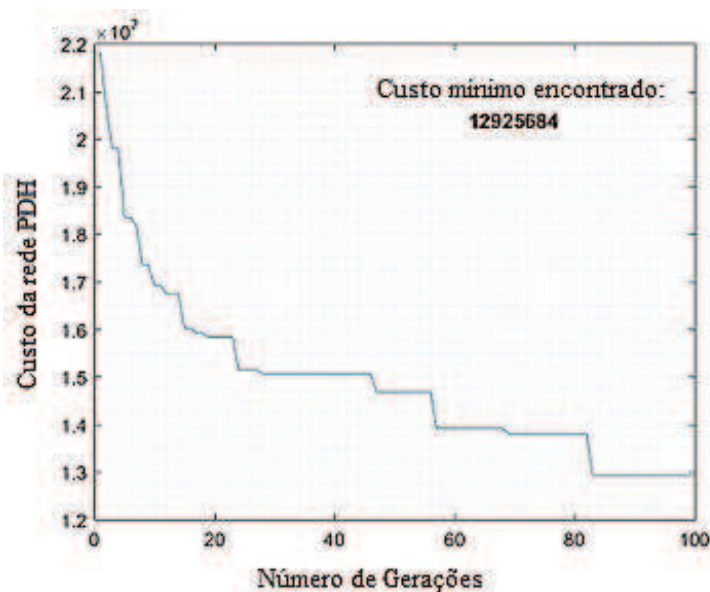


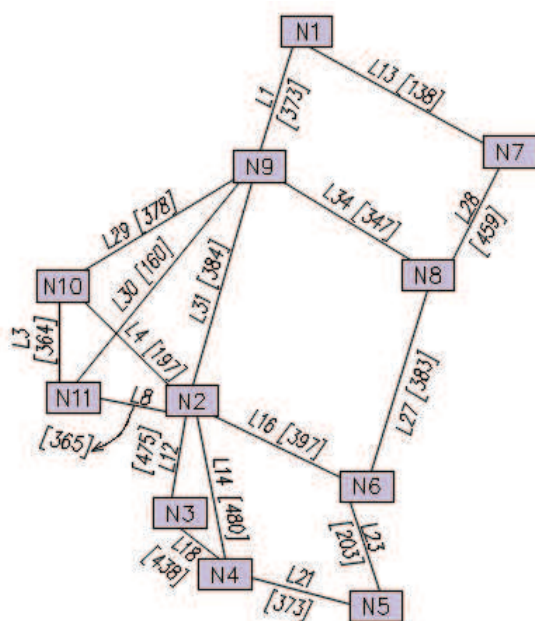
Figura 31 – Convergência do algoritmo proposto NDGA

A Tabela 19 apresenta os detalhes dos resultados que foram obtidos no experimento 04.

Tabela 19 – Resultados do experimento 04

Custo:	\$ 12.925.684,00
<i>Links</i> Utilizados:	17
Capacidade Instalada:	8160 E1
Capacidade Utilizada:	72,48 (%)
Tamanho médio dos caminhos:	1,38 <i>links</i>
Tempo	3 s

Os *links* utilizados e os respectivos valores de demanda em canais E1 passando por eles são mostrados na Figura 32 da rede otimizada.



Legenda:

N_i – Nós de demanda;

L_i - *Link* não-direcional;

[xE1] - Demanda prevista para o nó em canais E1.

Figura 32 – Rede PDH otimizada

A Tabela 20 apresenta as características principais mostrando as folgas e a capacidade instalada para a instância PDH da rede de bibliotecas SNDlib.

Tabela 20 – Características principais do melhor resultado apresentado pelo algoritmo

Origem	Destino	Link Utilizado	Capacidade Instalada	Capacidade Utilizada	Capacidade Utilizada (%)	Folga	Folga (%)
1	7	L13	480	342	71,25	138	28,75
1	9	L1	480	373	77,71	107	22,29
9	8	L34	480	347	72,29	133	27,71
7	8	L28	480	459	95,63	21	4,38
10	9	L29	480	378	78,75	102	21,25
11	9	L30	480	160	33,33	320	66,67
10	11	L3	480	364	75,83	116	24,17
11	2	L8	480	365	76,04	115	23,96
2	9	L31	480	384	80,00	96	20,00
2	6	L16	480	397	82,71	83	17,29
6	8	L27	480	383	79,79	97	20,21
10	2	L4	480	197	41,04	283	58,96
2	4	L14	480	480	100	0	0
2	3	L12	480	475	98,96	5	1,04
3	4	L18	480	438	91,25	42	8,75
4	5	L21	480	373	77,71	107	22,29
6	5	L23	480	203	42,29	277	57,71

4.2. NDGA e algoritmo de pós-otimização aplicado à rede ta1

A rede ta1 possui 34 nós e 55 *links* com 396 demandas a serem escoadas. A relação combinatorial entre o número de nós, *links* e demandas faz da rede ta1 um problema de alta complexidade, logo para essa rede foram utilizadas 12 configurações diferentes, descrito na Tabela 21.

Tabela 21 – Configuração dos Experimentos

Experi- Mentos	População	Gerações	Crossover (Probabilidade /Cortes)	Mutação (Probabilidade /Trocac)	Número de Competidores
1	100	100	0,9 / 5	0,06 / 2	3
2	200	100	0,9 / 5	0,06 / 2	3
3	300	100	0,9 / 5	0,06 / 2	3
4	400	100	0,9 / 5	0,06 / 2	3
5	400	100	0,9 / 5	0,06 / 2	3
6	100	100	0,9 / 5	0,06 / 5	3
7	200	200	0,9 / 5	0,06 / 2	3
8	300	300	0,9 / 5	0,07 / 2	3
9	400	400	0,9 / 5	0,08 / 2	3
10	500	500	0,9 / 5	0,08 / 2	3
11	1.000	500	0,95 / 10	0,08 / 5	3
12	1.000	500	0,95 / 30	0,1 / 10	5

A fim de verificar-se a variação dos resultados devido a alteração na quantidade de caminhos criados para cada demanda, os testes foram realizados com cinco e dez caminhos. Para o primeiro caso segue os valores na Tabela 22:

Tabela 22 – Resultados das simulações e da biblioteca SNDlib para 5 caminhos

Experimento	Custo (\$)	Quantidade de <i>links</i> utilizados	Capacidade instalada (Canais E1)	Capacidade utilizada (%)	Tamanho médio do caminho	Tempo (s)	
ta1 (SDH) ¹	7.606.778,96	24	44.856.000	58,95%	3,97	3600	
Valores Simulados do algoritmo proposto NDGA	1	10.039.179,26	28	60.480.000	58,01	4,67	796,53
	2	9.832.793,79	28	58.464.000	58,70	4,82	288,63
	3	9.759.084,82	28	58.464.000	57,48	4,37	424,98
	4	9.375.798,52	26	52.416.000	60,37	4,49	493,01
	5	9.375.798,52	26	52.416.000	60,37	4,49	483,42
	6	9.626.408,46	29	58.464.000	52,80	4,10	767,84
	7	9.818.052,12	29	58.464.000	55,29	4,25	419,65
	8	9.832.793,94	28	56.448.000	59,03	4,13	724,63
	9	9.508.473,79	27	56.448.000	61,76	4,35	507,99
	10	9.700.117,56	28	56.448.000	56,25	4,28	698,72
	11	9.508.473,90	26	52.416.000	64,88	4,18	514,90
	12	9.213.637,50	27	54.432.000	63,44	4,51	734,24

¹ Resultado da biblioteca SNDlib

A Tabela 23 apresenta os resultados da biblioteca SNDlib e os 12 experimentos. Os seguintes resultados foram obtidos para uma lista de 10 caminhos atribuídos para cada demanda.

Tabela 23 – Resultados das simulações e da biblioteca SNDlib para 10 caminhos

Experimento	Custo (\$)	Quantidade de <i>links</i> utilizados	Capacidade instalada (Canais E1)	Capacidade utilizada (%)	Tamanho médio do caminho	Tempo (s)	
ta1 (SDH) ¹	7.606.778,96	24	44.856.000	58,95%	3,97	3600	
Valores Simulados do algoritmo proposto NDGA	1	9.773.826,18	27	64.512.000	58,00	4,93	495,49
	2	9.847.535,44	28	62.496.000	70,00	4,74	475,20
	3	9.375.797,50	29	60.480.000	51,05	4,42	786,98
	4	9.316.830,24	29	58.464.000	56,40	4,39	1016,77
	5	9.316.830,24	29	58.464.000	56,40	4,39	430,56
	6	9.655.891,91	29	60.480.000	58,76	4,63	441,22
	7	9.169.412,04	26	52.416.000	69,20	4,85	910,94
	8	9.965.470,32	29	58.464.000	50,60	3,93	535,36
	9	8.918.801,10	27	54.432.000	69,23	4,85	547,65
	10	9.316.830,24	27	54.432.000	66,85	4,74	792,65
	11	9.552.699,36	29	58.464.000	54,59	4,22	639,81
	12	9.405.281,16	28	56.448.000	54,93	4,66	890,67

¹ Resultado da biblioteca SNDlib

Nos resultados para a rede ta1 nota-se um aumento considerável no tempo de resolução do problema. Isso ocorre devido ao algoritmo de pós-otimização que faz a simulação de falha em todos os *links* da rede.

Mesmo com a utilização de duas técnicas em sequência não foi possível chegar em uma solução melhor do que a apresentada na biblioteca. A melhor solução encontrada foi no experimento nove da Tabela 21 com dez caminhos disponíveis para o escoamento das demandas.

A Figura 33 mostra a convergência da solução que é dividida em duas partes: o NDGA representado na região anterior a reta vermelha pontilhada, e o algoritmo de pós-otimização na região posterior. Como indicado, o eixo das abscissas representa o número de gerações do NDGA (até geração quatrocentos) e também mostra o número de *links* retirados pela simulação de falha (no caso um total de 20 *links*).

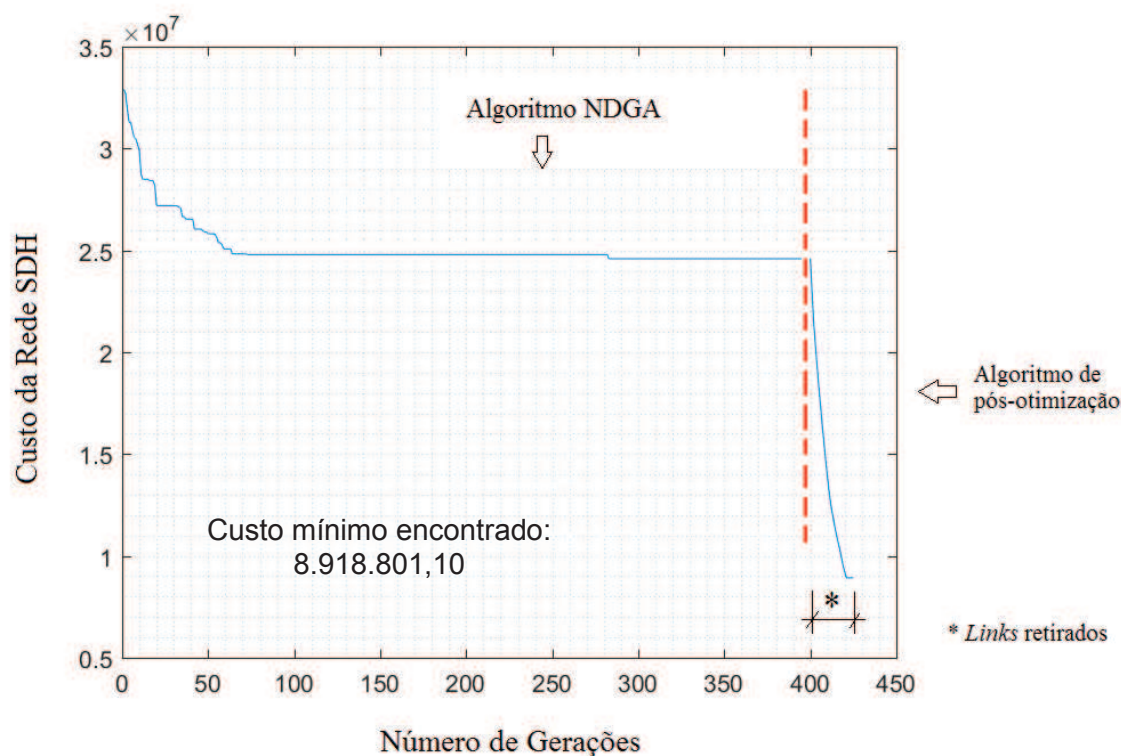


Figura 33 – Convergência do algoritmo proposto NDGA

A Tabela 24 apresenta os detalhes dos resultados com referência ao experimento 04.

Tabela 24 – Resultados do experimento 04

Custo:	\$ 8.918.801,10
Links Utilizados:	27
Capacidade Instalada:	54.432.000 u.r.
Capacidade Utilizada:	69,23 (%)
Tamanho médio dos caminhos:	4,85 links
Tempo	547,65 s

As informações detalhadas para cada *link* da rede são mostradas na Tabela 25.

Tabela 25 – Características principais do melhor resultado apresentado pelo algoritmo

Origem	Destino	Link Utilizado	Capacidade Instalada	Capacidade Utilizada	Capacidade Utilizada (%)	Folga	Folga (%)
1	20	L33	2.016.000	1.184.114	58,736	831.886	41,264
2	13	L48	2.016.000	1.111.637	55,141	904.363	44,859
2	19	L29	2.016.000	1.955.525	97,000	60.475	3,000
3	5	L18	2.016.000	1.313.025	65,130	702.975	34,870
3	14	L15	2.016.000	1.821.520	90,353	194.480	9,647
3	15	L16	2.016.000	2.012.111	99,807	3.889	0,193

3	20	L1	2.016.000	1.996.709	99,043	19.291	0,957
4	8	L17	2.016.000	990.078	49,111	1.025.922	50,889
4	12	L45	2.016.000	1.221.712	60,601	794.288	39,399
4	20	L2	2.016.000	835.712	41,454	1.180.288	58,546
5	6	L19	2.016.000	1.952.018	96,826	63.982	3,174
5	13	L47	2.016.000	1.969.857	97,711	46.143	2,289
6	15	L21	2.016.000	2.015.965	99,998	35	0,002
8	22	L22	2.016.000	538.741	26,723	1.477.259	73,277
9	23	L9	2.016.000	850.962	42,210	1.165.038	57,790
10	21	L8	2.016.000	1.034.480	51,313	981.520	48,687
11	18	L27	2.016.000	826.234	40,984	1.189.766	59,016
12	13	L46	2.016.000	1.320.661	65,509	695.339	34,491
14	24	L12	2.016.000	1.372.458	68,078	643.542	31,922
15	16	L25	2.016.000	1.547.991	76,785	468.009	23,215
15	18	L31	2.016.000	1.602.981	79,513	413.019	20,487
15	19	L30	2.016.000	1.710.541	84,848	305.459	15,152
16	18	L39	2.016.000	1.547.991	76,785	468.009	23,215
17	20	L5	2.016.000	970.646	48,147	1.045.354	51,853
17	23	L50	2.016.000	1.051.315	52,149	964.685	47,851
18	21	L28	2.016.000	1.415.253	70,201	600.747	29,799
23	24	L11	2.016.000	1.514.945	75,146	501.055	24,854

A solução encontrada gerou a topologia da Figura 34 abaixo. O módulo 4 apresentado anteriormente na Tabela 14 foi instalado em todos os *links*, sendo este de modularidade de 2.016.000 unidades de roteamento. Como o nó sete é uma central *tandem* não é necessário que ele esteja conectado ao restante da rede, pois não gera nem recebe nenhuma demanda.

Legenda:

N_i - Nós de demanda;
 L_i - Link não-direcional.

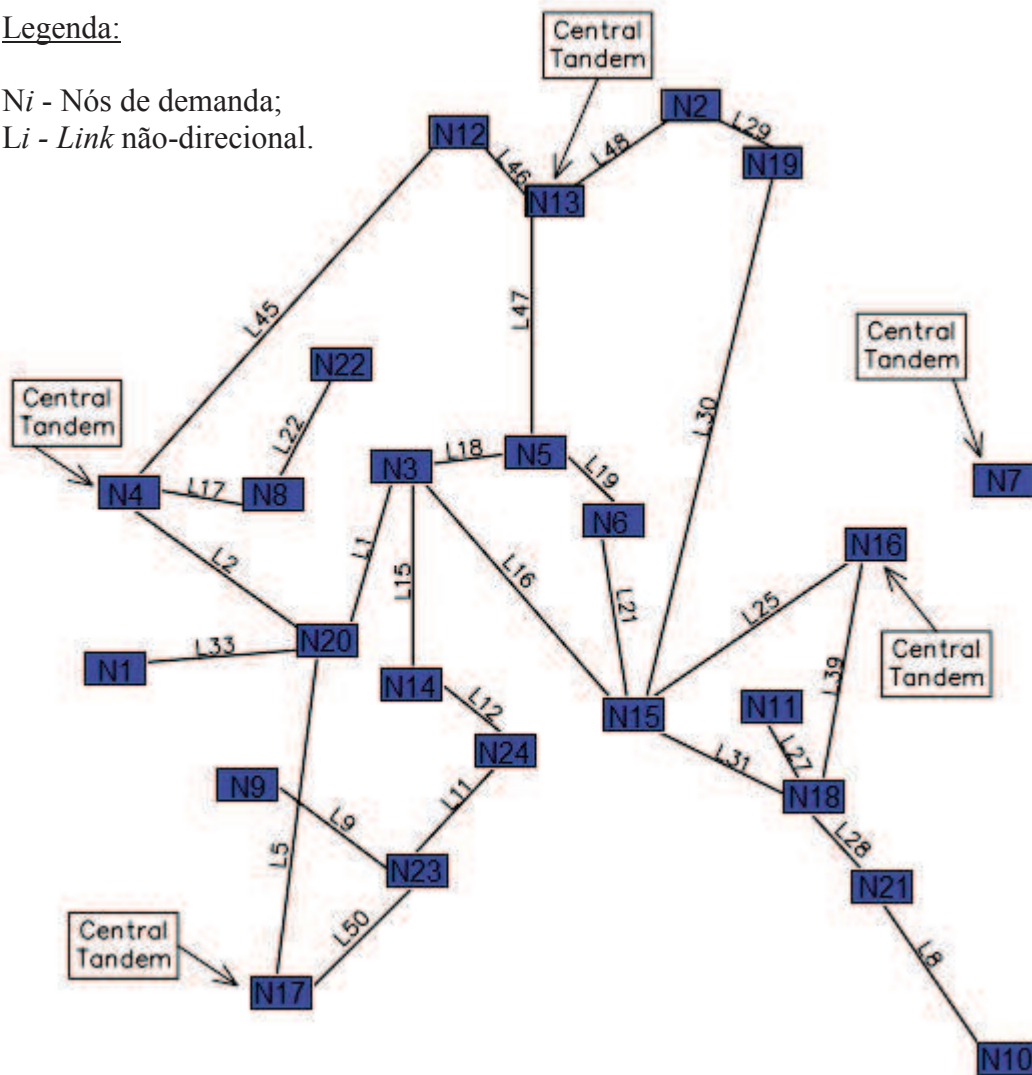


Figura 34 – Rede ta1 (SDH) otimizada

Em um grafo não é possível mostrar a dinâmica das demandas sendo escoadas na rede. Seguindo o padrão de solução da SNDlib a Tabela 26 mostra alguns caminhos utilizados para escoar cada demanda e a Tabela completa pode ser visualizada no Apêndice E (Tabela 27).

Tabela 26 – Caminhos utilizados para escoamento de demandas

Demandas	Links				
D1	29	30			
D2	19	21	33	47	48
⋮					
D395	5	28	29	48	50
D396	15				

CAPÍTULO V

Conclusão

O planejamento estratégico e a tomada de decisões são fundamentais para solucionar problemas de fluxo e garantir maior viabilidade e confiabilidade no transporte de informações e dados. Os projetos de redes de telecomunicações são altamente influenciados pela quantidade de nós e arcos, pela demanda a ser escoada e pelas modularidades disponíveis para alocação, tornando o planejamento da rede altamente complexo.

Para o *design* da rede foi proposto um algoritmo genético intitulado NDGA como alternativa aos métodos determinísticos, que trabalha com uma população de possíveis soluções da rede, e através dos operadores genéticos realiza a evolução da solução. As simulações realizadas permitiram uma visualização do esforço computacional requerido pelo problema de otimização da rede e também o conjunto de parâmetros a serem utilizados em uma instância específica.

Nos testes de uma rede PDH o NDGA se mostrou eficiente, pois apresentou uma redução do custo de até 15,15% e também de tempo computacional, quando comparado ao valor disponível na biblioteca SNDlib.

Após os resultados obtidos com a rede PDH, o algoritmo foi utilizado na otimização de uma rede de maior porte, a rede ta1. Os resultados obtidos apenas com o NDGA estavam longe dos desejados. Então, uma nova técnica foi introduzida para trabalhar com a solução fornecida pelo NDGA, um algoritmo de pós-otimização.

Com o algoritmo de pós-otimização foi possível uma redução considerável no custo, através da simulação de falha nos *links* da rede. Essa simulação de rompimentos de *links* é adequada para objetivos que envolvam apenas custos, pois ela diminui as folgas existentes na rede retirando a flexibilidade de atender demandas futuras não previstas.

Apesar de não conseguir um valor melhor do que o da biblioteca SNDlib, o NDGA associado ao algoritmo de pós-otimização chegou em um valor próximo em um menor tempo de processamento.

As duas redes são bem diferentes em relação ao número de demandas a serem escoadas, a rede PDH possui apenas 24 e a rede ta1 396. Através dessa diferença foi possível perceber que o NDGA trabalha melhor quando existem poucas demandas com

valores altos. E o algoritmo de pós-otimização opera melhor quando existem muitas demandas com baixos valores. Mas novos testes devem ser realizados.

Observou-se que o algoritmo proposto NDGA tem a característica de trabalhar melhor com poucas demandas de altos valores, devido a forma que os indivíduos são formados no processo de geração da população inicial: as demandas são enviadas pelos caminhos disponíveis e só depois os *links* são alocados. Sendo assim quando em redes como a ta1, a melhor forma de gerar a população inicial é atribuindo primeiramente capacidades aos *links*, e só depois escoar as demandas.

4.3. DESDOBRAMENTOS DO TRABALHO

Artigo aprovado e apresentado na modalidade oral, com a premiação de melhor artigo:

- CAMPOS, E. S., CUNHA, V. H., GUIMARÃES, L. C., DANTAS, M. J. P., LIMA, C. A. M., OLIVEIRA, B.Q. *Network Design Algorithm Genetic* (NDGA) aplicado em redes de telecomunicações. In: Conferências Ibero-Americanas WWW/Internet e Computação Aplicada 2016, 2016, Lisboa. **Atas das Conferências IADIS Ibero-Americanas**. Lisboa: IADIS Press, 2016. v.1. p.269 – 280.

Artigo aprovado e apresentado na modalidade oral:

- CUNHA, V. H., GUIMARÃES, L. C., CAMPOS, E. S., DANTAS, M. J. P. Algoritmo genético de codificação real aplicado à otimização de funções de *Benchmark*. Simpósio Brasileiro de Pesquisa Operacional - SBPO, Vitória - ES, 2016.

4.4. TRABALHOS FUTUROS

Como trabalhos futuros os estudos poderão ter continuidade com a hibridização do algoritmo com outras técnicas, como por exemplo, lógica *Fuzzy*, para o tratamento impreciso de demandas e custos, considerando as variações de mercado, ou mesmo, para inserção de regras no processo de evolução.

Neste trabalho, apenas duas instâncias foram avaliadas e podem ser realizados novos testes para outras instâncias da biblioteca de redes SNDlib, uma vez que há 23 topologias de redes disponíveis.

Identificou-se, também, a necessidade da evolução do modelo matemático atual para um modelo multiobjectivo, incluindo a avaliação de outras variáveis, além do custo.

REFERÊNCIAS:

ANDRÉ, J. et al. Design and dimensioning of hydrogen transmission pipeline networks. **European Journal of Operational Research**, v. 229, p. 239-251, Março 2013. www.elsevier.com/locate/ejor.

ANDRÉ, P. S. D. B. et al. Compensadores Adaptativos de Dispersão Cromática para Sistemas de Comunicação por Fibra Óptica. **IEEE Latin América Transactions**, v. 4, n. 5, p. 309-314, Setembro 2006.

BABATUNDE, O.; MBAROUK, S. A review of Plesiochronous Digital Hierarchy (PDH) and Synchronous Digital Hierarchy (SDH). **International Journal of Scientific Research Engineering & Technology (IJSRET)**, 3, Junho 2014. 677-681. www.ijret.org.

BANOV, J. T. M. **Abordagem Evolutiva para o Planejamento Multi-Período da Expansão da Rede de Acesso aos Serviços de Telecomunicações**. Campinas: Universidade de Campinas (UNICAMP), Faculdade de Engenharia Elétrica e de Computação (FEEC), Departamento de Engenharia de Sistemas (DENSIS), 2005. Tese (Doutorado em Engenharia Elétrica).

BAUMGARTNER, A.; REDDY, V. S.; BAUSCHERT, T. Combined Virtual Mobile Core Network Function Placement and Topology Optimization with Latency Bounds. **2015 Fourth European Workshop on Fourth Software Defined Networks - EWSDN**, Bilbao, 2015. 97-102. Disponível em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7313623&isnumber=7313575>. Data da Conferência: 30 de Setembro a 02 de Outubro de 2015.

BELGACEM, L.; CHARON, ; HUDRY,. A post-optimization method for the routing and wavelength assignment problem applied to scheduled lightpath demands. **European Journal of Operational Research**, v. 232, p. 298-306, 2014. ISSN 0377-2217. Disponível em: [journal homepage: www.elsevier.com/locate/ejor](http://journal.homepage: www.elsevier.com/locate/ejor).

BEN-AMEUR, W.; KERIVIN, H. Routing of Uncertain Traffic Demands. **International Multidisciplinary Journal to Promote Optimization Theory & Applications in Engineering Sciences**, v. 6, n. 3, p. 283-313, 01 Março 2005. ISSN 1389-4420.

BOAVENTURA NETTO, P.. **Grafos: Teoria, Modelos, Algoritmos**. 5ª. ed. São Paulo: Blücher, 2011. 326 p. ISBN 978-85-212-0680-4.

CANTODOSSANTOS, J. V.; BARRETO, M. B. Computacional System Based on Genetic Algorithm for Planning and Restoration of Telecommunication Networks. **International Conference Applied Computing (IADIS)**, Rio de Janeiro, Único, 2011. 241-248.

CAUCHICK MIGUEL, P. A. et al. **Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações**. 2ª. ed. Rio de Janeiro: Elsevier-Campus, 2012. 280 p. ISBN 978-85-352-4891-3.

CIRINO, R. B. Z.; SANTOS, ; DELBEM, A. C.. Aplicação da metaheurística AGC para o problema de alocação de aulas à salas. **XLVII SBPO - Simpósio Brasileiro de Pesquisa Operacional**, Porto de Galinhas, 25-28 Agosto 2015. 1745-1755.

COUËTOUX, B. et al. Labeled Traveling Salesman Problems: Complexity and approximation. **Discrete Optimization**, 7, n. 1, 11 Março 2010. 74-85. ISSN 15725286.

DESOUSA, M. A. et al. Interconexão de ERBs em um SMC - UM MILP para alocação e dimensionamento de enlaces SDH. **SBrT - Simpósio Brasileiro de Telecomunicações**, Fortaleza, XXXI, 1 a 4 Setembro 2013.

ELSAYED, S. M.; SARKE, A.; ESSAM, D.. A new genetic algorithm for solving optimization problems. **Engineering Applications of Artificial Intelligence**, p. 57-69, 26 Outubro 2014.

ENGELBRECHT, A. P. **Computational Intelligence: an introduction**. 2^a. ed. [S.l.]: John Wiley & Sons Ltd, 2007. ISBN 978-0-470-03561-0.

GOLDBARG, E.; GOLDBARG, M. C. **Grafos: conceitos, algoritmos e aplicações**. Rio de Janeiro: Elsevier, 2012. ISBN 978-85-352-5716-8.

GOLDBARG, M. C.; GOLDBARG, G.; LUNA, H. P. L. **Otimização combinatória e meta-heurísticas: algoritmos e aplicações**. 1^a. ed. Rio de Janeiro: Elsevier, 2016. ISBN 978-85-352-7812-5.

GOLDBERG, D. E. **Genetic algorithms in search, optimization, and machine learning**. [S.l.]: Addison-Wesley Publishing Company, Inc, 1989. 412 p. ISBN 0-201-15767-5.

GUIOTTI, F. G. **Otimização de recursos através da gestão integrada da rede de transporte**. Brasília: Universidade de Brasília, 2007. 96 p. Mestrado Profissional (Engenharia Elétrica). PPGENE.DM-045A/07.

HERRERA, F.; LOZANO, M.; VERDEGAY, J. L. Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. **Artificial Intelligence Review**, 12, n. 4, 1998. 265-319. ISSN 1573-7462.

HILLIER, F. S.; LIEBERMAN, G. J. **Introduction to operations research**. 9^a. ed. New York: Mc Graw-Hill Education, 2009.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**. [S.l.]: MIT Press, 1975. 211 p.

ITU-T (INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR). **G.702 - General Aspects of Digital Transmission Systems - Digital Hierarchy Bit Rates**. [S.l.]: [s.n.], v. Fascicle III.4 - Rec. G.702, 1988, 1993. 4 p.

ITU-T (INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR). **G.707 - Network node interface for the Synchronous Digital Hierarchy (SDH)**. [S.l.]: [s.n.], 2003.

JYOTHIRMAI, N.; VALLI, R. M.; KRISHNA, A. R. SDH and Its Future Trends. **IJITEE - International Journal of Innovative Technology and Exploring Engineering**, v. 1, n. 6, p. 74-78, Novembro 2012. ISSN 2278-3075.

LACERDA, E. G. M.; CARVALHO, A. C. P. L. F. D. Introdução aos algoritmos genéticos. In: GALVÃO, C. O.; VALENÇA, M. J. S. **Sistemas Inteligentes: aplicações**

a recursos hídricos e ciências ambientais. 1ª. ed. Porto Alegre: Universidade Federal do Rio Grande do Sul - UFRGS, 1999. Cap. 3, p. 99-150.

LIENIG, J.; THULASIRAMAN, K. A Genetic Algorithm for Channel Routing in VLSI Circuits. **MIT Press Journals - Evolutionary Computation**, Montreal, Quebec, 1, n. 4, 1993. 293-311.

LINDEN, R. **Algoritmos Genéticos**. 3ª. ed. Rio de Janeiro: Ciência Moderna Ltda., 2012. ISBN 978-85-399-0195-1.

LUKE, S. **Essentials of Metaheuristics**. 2ª. ed. [S.l.]: [s.n.], 2015. ISBN 978-1-300-54962-8. Disponível em: <<http://cs.gmu.edu/~sean/book/metaheuristics/>>. Acesso em: 21 Janeiro 2017. Versão Online 2.2.

MELO, D. W. F. **Empregos de Algoritmos Genéticos na Generalização de Modelos Digitais de Superfície**. Rio de Janeiro: Instituto Militar de Engenharia. Dissertação (Mestrado em Engenharia Cartográfica), 2005.

MELO, J. F. D. **Aplicação da Teoria dos Grafos e Simulação Computacional para Dimensionamento de Redes Hidráulicas Industriais**. Recife: Universidade Federal de Pernambuco (UFPE) - Centro de Tecnologia e Geociências, 2013. Dissertação (Mestrado em Materiais e Fabricação).

MILLER, B. L.; GOLDBERG, D. E. Genetic algorithms, tournament selection, and the effects of noise. **Complex Systems**, 9, n. 3, 1995. 193- 212.

MITCHELL, M. **An Introduction to Genetic Algorithms**. Londres: MIT Press Journals, 1996.

OBITKO, M.; SLAVÍK, P. Visualization of Genetic Algorithms in a Learning Environment. **Spring Conference on Computer Graphics, SCCG'99**, Bratislava, 1999. 101-106. ISBN 80-223-1357-2.

OLIVEIRA, B. Q. D. **Modelo de Otimização PLIM-Fuzzy para o dimensionamento de rede e seleção de serviços em telecomunicações**. Goiânia: Pontifícia Universidade Católica de Goiás - PUC GO, Mestrado em Engenharia de Produção e Sistemas - MEPROS, 2015. Dissertação (Mestrado em Engenharia de Produção e Sistemas).

ORLOWSKI, S. et al. SNDlib 1.0 - Survivable Network Design Library. **Networks - An International Journal**, v. 55, n. 3, p. 276-286, 10 Maio 2010. doi 10.1002/net.20371.

PALLE, A. V. SDH (Synchronous Digital Hierarchy) & Its Architecture. **IJSRD - International Journal for Scientific Research & Development**, Hyderabad, v. 1, n. 7, 2013. ISSN 2321-0613.

PASTOR, E. T. L. **Algoritmo de RWA com considerações de sobrevivência baseado em heurística-algoritmo genético para redes IP/WDM**. Brasília: Universidade de Brasília (UnB), Faculdade de Tecnologia (FT), Departamento de Engenharia Elétrica, 2007. 217 p. Tese (Doutorado em Engenharia Elétrica).

PIÓRO, M.; MEDHI, D. **Routing, Flow, and Capacity Design in Communication and Computer Networks**. São Francisco: Elsevier, 2004. ISBN: 0-12-557189-5.

PRATA, B. D. A. A Hybrid Genetic Algorithm for the Vehicle and Crew Scheduling in Mass Transit Systems. **IEEE LATIN AMERICA TRANSACTIONS**, 13, n. 9, Setembro 2015. 3020-3025.

PRIMETRICA. TeleGeography. www.primetrica.com, 1989. Disponível em: <<https://www.telegeography.com>>. Acesso em: 14 fev. 2017.

ROCHA, H. X. et al. Statistical Decision Support for Fitness Function Formulation to Multi-Objective Evolutionary Algorithms: a Case Study on Computer Automated Exterior Lighting Design. **IEEE America Latina Transactions**, v. 14, n. 1, p. 281-288, Janeiro 2016. ISSN 1548-0992.

RODRÍGUEZ-MARTÍN, I.; SALAZAR-GONZÁLEZ, J.-J.; YAMAN,. Hierarchical Survivable Network Design Problems. **Electronic Notes in Discrete Mathematics**, 52, 2016. 229-236. Disponível em: <www.elsevier.com/locate/endm>.

TANOMARU, J. Motivação, Fundamentos e Aplicações de Algoritmos Genéticos. **II Congresso Brasileiro de Redes Neurais - III Escola de Redes Neurais**, Curitiba, 28 a 01 Outubro a Novembro 1995.

TRIPATHY, P. K.; DASH, R. K.; TRIPATHY, C. R. A Genetic Algorithm Based Approach for Topological Optimization of Interconnection Networks. **Procedia Technology - 2nd International Conference on Communication, Computing & Security [ICCCS-2012]**, 6, 2012. 196-205.

VASSEUR, J.-P.; PICKAVET, M.; DEMEESTER, P. **Network recovery: protection and restoration of optical, SONET-SDH, IP, and MPLS**. [S.l.]: Elsevier, 2004. ISBN: 0-12-715051-x.

ZUSE INSTITUTE BERLIN (ZIB). **SNDlib**. [S.l.]: [s.n.], 2005. Disponível em: <<http://sndlib.zib.de/home.action>>. Acesso em: 26 Julho 2016.

APÊNDICE - A

Artigo apresentado à 4.^a Conferência Ibero Americana Computação Aplicada 2016 promovida pela IADIS - International Association for Development of the Information Society em Lisboa, Portugal no período de 10 a 11 de dezembro de 2016.

NETWORK DESIGN GENETIC ALGORITHM (NDGA) APLICADO EM REDES DE TELECOMUNICAÇÕES

Emerson de Souza Campos^{1,2}, Victor Honorato Cunha³, Dr.^a Maria José Pereira Dantas¹
Lucas de Carvalho Guimarães³, Celso Aurélio Lima¹ e Bruno Quirino de Oliveira³

¹Mestrado em Engenharia de Produção e Sistemas (MEPROS), Pontifícia Universidade Católica de Goiás (PUC-Goiás)
Av. Universitária, 1069, Setor Universitário, CEP 74.650-010, Caixa Postal 86, Goiânia-GO, Brasil

²Faculdade de Tecnologia SENAI Ítalo Bologna (FATEC SENAI – Departamento Regional de Goiás)
Rua Armogaste José da Silveira, 612 – Setor Centro Oeste – CEP 74.560-020, Goiânia-GO, Brasil

³Departamento de Engenharia (ENG), Pontifícia Universidade Católica de Goiás (PUC-Goiás)
Av. Universitária, 1440, Área III, Bloco G, Sala 100, CEP 74.605-010, Goiânia-GO, Brasil

RESUMO

O objetivo deste artigo é propor e avaliar um Algoritmo Genético para a otimização de redes de telecomunicações. O algoritmo é mono-objetivo e o custo da rede é utilizado como função de aptidão. A codificação das soluções é feita através de matrizes de incidência, respeitando as características da rede (demandas e as conexões entre nós) e os recursos disponíveis para instalação (solução técnica disponível). Os operadores genéticos propostos atuam sobre matrizes. O algoritmo foi aplicado a uma instância da biblioteca SNDlib (*Survivable Network Design Library*). Os resultados obtidos mostraram uma redução de 15,15% no custo total da rede utilizada para fins de comparação. Os resultados são promissores para avaliações de redes de maior porte.

PALAVRAS-CHAVE

Sistemas de Telecomunicações, otimização, algoritmos genéticos, Biblioteca SNDlib

1. INTRODUÇÃO

O planejamento insere-se numa gama de problemas complexos a serem solucionados por sistemas computacionais. Este trabalho apresenta um algoritmo intitulado Network Design Genetic Algorithm (NDGA) para a otimização de redes de telecomunicações. As redes estão presentes em diversos ambientes com uma vasta variedade de cenários tais como: transporte, eletricidade, gás, redes de comunicação, produção, distribuição, dentre outros.

As representações em rede fornecem informações de um grupo de componentes inter-relacionados com o objetivo de produzir resultados em um processo organizado de modo a possibilitar maior eficiência e maior interação entre sistemas de comunicação.

Os sistemas de telecomunicações atuais apresentam demanda crescente de usuários, com o surgimento de novos serviços tais como aplicações multimídia, jogos *online*, compartilhamento de arquivos, mensagens instantâneas de dados e vídeo e outros agregaram novas tecnologias (Ben-Ameur & Kerivin, 2005).

O problema de planejamento de redes é complexo e de difícil solução. As diversas demandas devem ser escoadas entre os nós, por meio de *links* com uma capacidade associada. Isso se explica pela relação entre o número de nós de uma rede com a complexidade computacional do problema, pois na medida em que se aumenta a quantidade de nós também há um aumento da quantidade de variáveis. Essa relação entre número de nós, quantidade de *links* e valores de demanda caracteriza o problema como combinatorial.

As técnicas de otimização determinísticas, que encontram a solução ótima, apresentam algumas limitações quanto às características de um problema de redes. Isso ocorre devido às funções a serem otimizadas, que não são contínuas, são não convexas, apresentam multimodalidade (vários pontos ótimos); existe a necessidade de se trabalhar com valores discretos para as variáveis, entre outras (CantoDosSantos & Barreto, 2011).

ALGORITMO GENÉTICO DE CODIFICAÇÃO REAL APLICADO À OTIMIZAÇÃO DE FUNÇÕES DE *BENCHMARK*

Victor Honorato Cunha

Pontifícia Universidade Católica de Goiás
Departamento de Engenharia, Av. Universitária, 1440, Área III, Goiânia, Goiás - 74605010
victor.puc.engel@gmail.com

Emerson de Souza Campos

Pontifícia Universidade Católica de Goiás
MEPROS, Av. Universitária, 1440, Área III, Goiânia, Goiás - 74605010
emersoncampos.engel@gmail.com

Lucas de Carvalho Guimarães

Pontifícia Universidade Católica de Goiás
Departamento de Engenharia, Av. Universitária, 1440, Área III, Goiânia, Goiás - 74605010
lucascg1995@gmail.com

Maria José Pereira Dantas

MAF - Pontifícia Universidade Católica de Goiás
MEPROS, Av. Universitária, 1440, Área III, Goiânia, Goiás - 74605010
mjpdantas@gmail.com

RESUMO

O trabalho compara resultados obtidos por um algoritmo genético elitista de codificação real, com trabalhos da literatura que utilizam diferentes técnicas e operadores, na otimização das seguintes funções de *benchmark*: Goldstein-Price, Rosenbrock, Easom, Zakharov e Shubert. Dois parâmetros são utilizados na avaliação de eficiência: o número de avaliações da função objetivo e a taxa de sucesso de convergência. Para a visualização da complexidade das funções utilizadas é fornecido a sua representação matemática e gráfica, juntamente com um gráfico de convergência, que mostra a evolução da solução no decorrer das gerações do algoritmo genético. O algoritmo convergiu em cinco das sete funções utilizadas, com taxa de sucesso entre 47% e 90%. A comparação com outros trabalhos mostra a adequação dos algoritmos genéticos na solução de problemas de otimização, e também a necessidade do uso de técnicas específicas adjacentes para a solução de cada problema.

PALAVRAS CHAVE. Algoritmos Genéticos, Codificação Real, Funções de *Benchmark*.
TÓPICOS. MP (Modelos Probabilísticos); MH (Metaheurísticas).

ABSTRACT

This work compares the results obtained by a real-coded elitist genetic algorithm, with literature works that uses different techniques and operators, in the optimization of the following benchmark functions: Goldstein-Price, Rosenbrock, Easom, Zakharov and Shubert. Two parameters are used in the evaluation of efficiency: the number of objective function evaluations and convergence success rate. To the visualization of the complexity of the used functions is provided its mathematical and graphical representation, along with a convergence chart, which shows the evolution of the solution through the generations of the genetic algorithm. The algorithm converges in five of the seven functions used, with a success rate between 47% and 90%. Comparison with other studies shows the adequacy of genetic algorithms to solve optimization problems, and also the need to use specific adjacent techniques to the solution of each problem.

KEYWORDS. Genetic Algorithm, Real Coded, *Benchmark* Functions.
PAPER TOPICS. MP (Probabilistic Models); MH (Metaheuristics)

1. Introdução

A modelagem matemática de problemas reais de planejamento são utilizados modelos de otimização. Otimizar significa encontrar o equilíbrio entre o objetivo a ser alcançado e as restrições envolvidas, de forma que o resultado seja satisfatório.

Quando o espaço de busca onde estão contidas as soluções é relativamente pequeno é possível avaliar todas as soluções e apresentar a melhor depois dessa avaliação. Esses problemas são ditos como tratáveis, “em termos práticos, um problema é tratável se o seu limite superior de complexidade é polinomial” [Linden 2012]. Método da bisseção e método simplex são exemplos de modelos determinísticos utilizados na resolução de problemas tratáveis. Por outro lado, a grande maioria dos problemas tem o seu nível superior de complexidade de ordem exponencial, o que caracteriza um problema intratável. Um exemplo clássico é o do caixeiro viajante [Couëtoux *et al.*, 2010].

Na resolução de problemas intratáveis (NP-Completo), existem métodos heurísticos tendo base na natureza e conceitos sociais [Mollinetti 2013]. Não avaliam totalmente o espaço de busca, no entanto utiliza-se de uma informação anterior para chegar a uma boa solução. Esses algoritmos existem porque as resoluções de problemas de ordem exponencial por métodos determinísticos levariam um tempo muito grande para serem solucionados [Linden 2012], fora do limite aceitável. Os métodos heurísticos são utilizados quando as ferramentas determinísticas tornam-se inviáveis, isso geralmente ocorre quando o número de variáveis envolvidas é consideravelmente grande [Cirino *et al.* 2015].

Algoritmos Genéticos (AG's), objetivo de estudo deste trabalho, faz parte desses algoritmos supracitados. São dotados para encontrar soluções baseadas no processo biológico de evolução e seleção natural de Charles Robert Darwin [Linden 2012], onde, por meio de operadores genéticos, as características favoráveis são evidenciadas em gerações sucessivas e as características desfavoráveis tendem a ser minimizadas ou extinguir-se.

Os AG's têm se mostrado eficientes na solução de muitos problemas como, por exemplo, na engenharia [Lienig e Thulasiraman 2007], [Peralta 2013], [Chang *et al.* 2010], ciência da computação [Gordon 1988], ciências sociais [Fisher 2008], medicina [Tosatti *et al.* 2008] e outras. O grande número de aplicações em diversas áreas se justifica por sua facilidade de implementação, por sua performance em sua busca por melhores soluções e pelo fato de não serem fundamentalmente limitados por restrições do espaço de busca [Goldberg 1989].

Os algoritmos genéticos são propostos de acordo com o problema que irão resolver [Doerr *et al.* 2012]. No ano de 1975 Jhon Holland criou os AG's de codificação binária. Seus trabalhos foram popularizados por seu aluno Goldberg [Goldberg 1989]. Mais tarde Kenneth A De Jong [Jong 1975] em sua tese de doutorado analisa o comportamento dos algoritmos genéticos em diversas situações utilizando funções reais, mostrando a eficiência dos AG's.

A fim de avaliar a eficiência de algoritmos, existem as chamadas funções de *Benchmark* [Molga 2005], que incluem também as funções utilizadas por De Jong [Jong 1975]. Elas oferecem um meio pelo qual esses algoritmos probabilísticos podem ser avaliados. O intuito deste trabalho é submeter um algoritmo genético a essas funções e comparar os resultados obtidos com outros trabalhos da literatura, contribuindo com novos resultados e eficiência de técnicas utilizadas; conteúdo indispensável para pesquisadores da área que desejam avaliar a eficiência de seus algoritmos.

Na seção 2 é apresentada uma breve revisão da teoria dos AG's. A descrição das escolhas feitas para o desenvolvimento do algoritmo é apresentada na seção 3. A seção 4 define como são realizados os experimentos. Os resultados são apresentados na seção 5. Por fim, na seção 6 é feita uma análise seguida de discussão do algoritmo genético desenvolvido.

2. Origem

O termo genético vem de Darwin e sua teoria da evolução das espécies descrita em sua obra “A Origem das Espécies”. Onde é dito que a evolução dos indivíduos ocorre através da seleção natural, ou seja, o indivíduo que melhor se adapta ao meio em que vive tem mais chances de sobreviver e repassar suas características para a próxima geração; sendo que os indivíduos menos adaptados ainda podem se reproduzir, no entanto, com menor sucesso. A seleção é feita a partir de vários acontecimentos na vida de um indivíduo de uma determinada população, como, por exemplo, disputa por alimento.

Em 1975 o propósito de John Holland em utilizar a evolução natural como método de resolução de problemas de otimização, resultou nos algoritmos genéticos, que por sua vez são uma simulação computacional da evolução natural. Esses algoritmos são uma técnica heurística, baseados na experiência, utilizam-se desta informação para se guiar no conjunto das possíveis soluções de um problema. Através de operados genéticos, tais como seleção, cruzamento e mutação; realizam uma busca aleatório-direcionada no espaço de busca aleatória para encontrar soluções “otimizadas”.

As possíveis soluções são representadas por cromossomos, isto é, população de indivíduos (pais) que submetida a certas modificações, geram novos candidatos (filhos) com a tendência de melhorar a desempenho de algo em direção a algum ponto ou pontos otimizados [Serrada 1996].

A estreita relação dos AG’s e a genética faz com que muitos termos sejam utilizados na teoria dos algoritmos genéticos. A Tabela 1 faz uma analogia com a biologia, referente a tais termos.

Tabela 1 – Terminologia utilizada nos algoritmos genéticos

Biologia	Algoritmo Genético
Cromossomo	Indivíduo
Gene	Característica
Alelo	Valor de cada variável
Genótipo	Estrutura
Fenótipo	Conjunto de parâmetros
Indivíduo	Solução candidata de um problema
População	Conjunto de indivíduos de uma geração
Geração	Ciclo de criação e de transformação de um problema
Adequabilidade	Função de avaliação

Fonte: Adaptado de Calixto, 2008

2.1. Codificações Binária e Real (ou decimal)

Existem duas codificações tradicionais nos AG’s, a binária e a real. A codificação binária foi utilizada nos trabalhos de John Holland, sendo, portanto, historicamente importante. Nesse tipo de codificação os dados são armazenados em cadeias de bits (cromossomos). Quando os dados são parâmetros discretos, a codificação binária é a mais indicada; sua simplicidade de análise também justifica o seu uso.

Quando as variáveis do problema são contínuas, é mais eficiente se utilizar a codificação real. No espaço de busca contínuo a precisão da solução é motivo de atenção; ao representarmos as soluções através de números reais, a precisão é aumentada facilmente pela simples adição de mais casas decimais. Já na codificação binária se quisermos aumentar a precisão em uma casa, devemos acrescentar 3,3 bits em cada indivíduo.

É de fundamental importância entender que o comportamento dos algoritmos genéticos é influenciado diretamente pelos seus operadores e parâmetros. Os operadores genéticos fundamentais são a seleção, o cruzamento e a mutação, enquanto que, o tamanho da população inicial, a taxa de cruzamento e a taxa de mutação são os parâmetros genéticos mais importantes. [Catarina e Bach 2003].

A dificuldade de se utilizar cadeias binárias para representação de um espaço contínuo é principalmente o seu impacto no tempo de processamento. Os dados contínuos devem ser codificados e decodificados, demandando maior tempo de execução. E muitos dos problemas a serem resolvidos têm o seu domínio no espaço multidimensional, por exemplo: se utilizarmos 22 bits para cada variável de decisão do problema, os indivíduos terão $22 \cdot D$ bits, sendo D o número de variáveis; e novamente o desempenho do AG binário é reduzido.

3. Metodologia

3.1. Codificação e Funcionamento

Cada codificação possui os seus operadores genéticos. O algoritmo desenvolvido (AG_Real) neste trabalho para o teste de desempenho é de codificação real, sendo assim, os operadores a seguir apresentados, são mais comumente utilizados em codificação decimal. Porém seus princípios e conceitos podem ser utilizados para qualquer codificação.

O AG_Real foi implementado em MATLAB®, por ser uma linguagem de alto nível dedicada a cálculos matemáticos, e sua simplicidade de programação devido à utilização de matrizes, e segue o modelo inicial proposto por Holland, exceto pela inclusão da técnica de elitismo Figura 1. O algoritmo é dedicado à maximização de funções reais, cujo domínio esteja no R^n , ou seja, abrange funções multivariáveis [Jamil and Yang 2013].

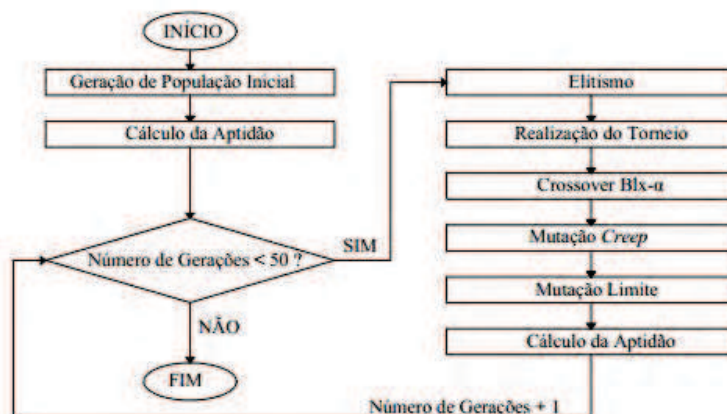


Figura 1 – Fluxograma AG_Real

3.2. Seleção

Utilizada em vários trabalhos [Miller and Goldberg 1995] a seleção por torneio foi escolhida como critério de seleção, ela determina, em parte, a pressão de seleção do AG_Real, pois os indivíduos melhores adaptados ao meio, dizemos os que possuem boas aptidões, têm maiores chances de serem escolhidos para o *crossover* (cruzamento). São escolhidos n cromossomos da população (conjunto das possíveis soluções do problema) aleatoriamente e com mesma probabilidade. O indivíduo com maior aptidão é então selecionado para o *crossover*.

3.3. Crossover

O *crossover* é o meio pelo qual os bons indivíduos, escolhidos na seleção, repassam sua característica para a próxima geração. Em uma analogia com o meio natural, o *crossover* pode ser comparado à reprodução sexuada dos seres. No AG_Real o cruzamento é feito pelo *crossover* BLX- α .

O *crossover* BLX- α tem uma interessante propriedade. A localização do novo indivíduo depende da diferença entre os pais. Se a diferença entre eles é pequena, a diferença entre pais e filhos é pequena também e vice-versa [Chen e Wang 2011].

Os cromossomos pais ($p1$ e $p2$) possuem números decimais, cada qual relacionado a uma variável da função utilizada, e são submetidos à equação (1):

$$c = p1 + \beta * (p2 - p1) \quad (1)$$

Tradicionalmente é definido $\alpha = 0,5$, ele estende em 0,5 unidades o intervalo entre os dois cromossomos pais, ou seja, os que irão participar do cruzamento. O parâmetro β aleatoriamente escolhido dentro do intervalo $[-\alpha; 1 + \alpha]$, esse processo ameniza a tendência de que os filhos sejam gerados próximos ao centro do intervalo entre os dois pais. Esse tipo de *crossover* explora melhor o espaço de busca das funções em comparação com outros operadores de cruzamento [Ariyarat e Kanazaki 2015].

Nem todos os indivíduos passam pelo operador de *crossover*. Existe uma probabilidade relacionada a esse parâmetro e o processo é repetido até formar todos os indivíduos (filhos) da próxima geração que ainda podem passar pelo operador de mutação.

3.4. Mutação

A mutação é responsável por evitar que a população permaneça apenas em uma pequena parte do espaço de busca, ou ainda, que a solução encontrada seja local e não global. O AG_Real utiliza duas mutações: *creep* e limite.

A mutação *creep* é uma mutação local onde cada gene do cromossomo (valor numérico relacionado a cada variável), é multiplicado por um número (aleatório) próximo de 1, no intervalo $[0,95; 1,05]$. De modo que, se o indivíduo está próximo do máximo global, ele irá rapidamente encontrar a melhor solução do problema. Por não ser muito destrutiva, a mutação *creep* pode ser associada a uma probabilidade relativamente alta. É adequada para representações reais dos genes [Soni e Kumar 2014].

Já a mutação limite é altamente destrutiva, ela escolhe um indivíduo da população e move-o para o limite inferior ou superior do domínio considerado da função. Assim, a probabilidade da mutação limite deve ser baixa, para evitar que se tenha perda de progresso durante as gerações. As probabilidades são definidas de forma que o processo de busca não se torne totalmente aleatório.

3.5. Elitismo

O elitismo é um mecanismo que mantém bons indivíduos nas gerações futuras. É utilizado para assegurar que as soluções sejam iguais ou melhores que as alcançadas anteriormente, contribuindo para uma melhor convergência do AG_Real. Para isso, o melhor indivíduo é mantido na próxima geração sem sofrer alteração dos operadores genéticos na próxima geração [Herrera et al. 1998].

4. Experimentos

A avaliação do AG_Real é feita a partir de sua aplicação em funções *benchmark* através de comparação com outros trabalhos da literatura. As funções de *benchmark* são funções em que seus máximos ou mínimos já são conhecidos. Elas são um meio pelo qual o desempenho dos AG's pode ser avaliado. Outra vantagem de se utilizar essas funções, é que elas “desafiam” os algoritmos genéticos; pois oferecem uma variedade de topologias diferentes na sua superfície, de maneira que a robustez, ou seja, o balanço entre eficiência e eficácia para que ele obtenha bons resultados em diferentes “meio ambientes” também pode ser avaliada [Goldberg 1989].

Por ser um algoritmo probabilístico, mesmo encontrando boas soluções para um determinado problema, os AG's podem não encontrá-las novamente, ainda que as condições (parâmetros) de execução sejam mantidas. Dessa forma, realizar apenas um teste para cada função, não garante a robustez do algoritmo, já que pode-se obter diferentes soluções a cada execução.

Os parâmetros utilizados nas simulações são mostrados na Tabela 2. Esses valores foram escolhidos após a realização de vários testes [Bessaou e Siarry 2001].

Tabela 2 – Configuração dos parâmetros.

Parâmetros	Configurações
Tamanho da População	50
Número de Competidores no Torneio	3
Probabilidade de <i>Crossover</i> (Blx-0,5)	90%
Probabilidade de Mutação (Creep)	40%
Probabilidade de Mutação (Limite)	0,5%
Número de Gerações	50
Elitismo	1

Cada função é testada cem vezes. É considerada a taxa de sucesso do AG_Real, que fornece porcentagem de sucesso de convergência do algoritmo no universo dos cem testes realizados. É considerado convergência quando a lacuna entre o mínimo global analítico da função e o mínimo encontrado pelo algoritmo atinge uma precisão de 3 casas decimais. A precisão considerada depende das exigências do usuário e da grandeza com a qual se trabalha, a aplicação em funções é geral, logo essa escolha foi feita através da observação dos resultados apresentados por Chelouah e Siarry [2000]. É feita uma média do número de avaliações da função objetivo de todas as execuções que convergiram, uma dessas execuções é aleatoriamente utilizada para demonstrar através de gráfico com deu a convergência do AG_Real para o mínimo global da respectiva função [Chelouah e Siarry 2000].

5. Resultados

Primeiramente é mostrada a formulação matemática, depois uma visualização no espaço das funções utilizadas seguido dos resultados encontrados, o gráfico de convergência e comentários. São sete funções no total. Uma tabela de comparação é apresentada no fim desta seção.

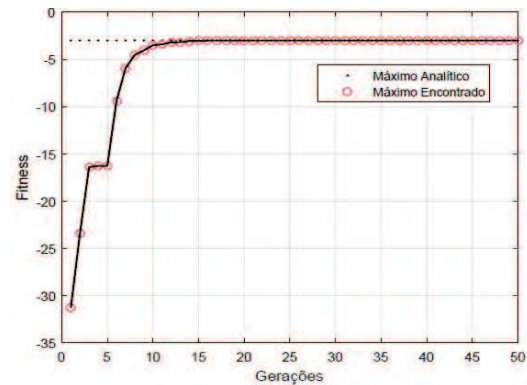
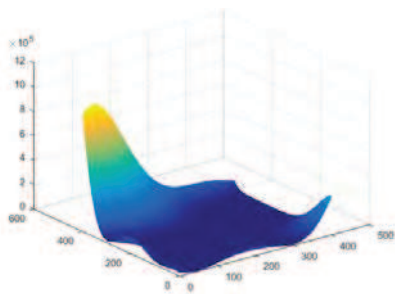
Função de Goldstein-Price (GP):

$$GP(x_1, x_2) = [1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 13x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 - 48x_2 - 36x_1x_2 + 27x_2^2)] \quad (2)$$

Domínio de busca: $-2 \leq x_j \leq 2, j = 1, 2;$

4 mínimos locais e um mínimo global $(x_1, x_2) = (-1, 0);$

Mínimo Global: $GP[(x_1, x_2)] = 3$



Taxa de sucesso: 87%
 Avaliações da função objetivo: 695

(a) Representação Gráfica

(b) Convergência

Figura 2 – Função Goldstein-Price

A função de Goldstein-Price (Eq. 2) (Figura 2a), tem duas variáveis e é uma função de teste de otimização global, pois possui apenas um mínimo global. O algoritmo apresenta bons resultados para esta função. A rápida convergência do algoritmo (Figura 2b) se deve ao fato de que o domínio considerado desta função é relativamente pequeno para o tamanho da população utilizada (cinquenta indivíduos), assim é possível varrer o espaço de busca como um todo. Encontrar o máximo desta função é simples, e pode-se utilizar técnicas como subida de encosta. Porém quando se busca sua minimização, ela apresenta um desafio para o algoritmo, pois possui 4 mínimos globais.

Função de Rosenbrock (R_n (variáveis)):

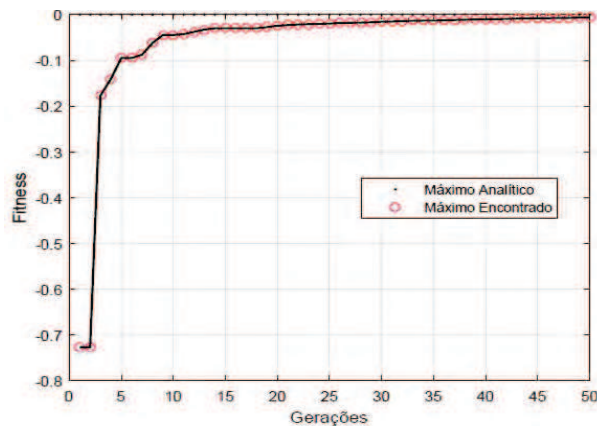
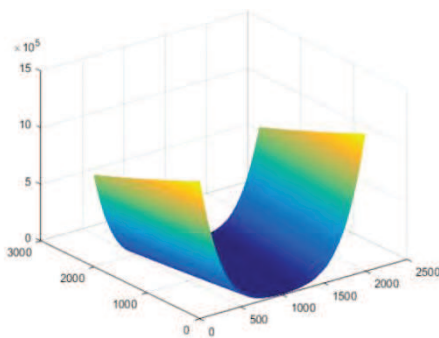
$$R_n(x) = \sum_{j=1}^{n-1} [100(x_j^2 - x_{j+1})^2 + (x_j - 1)^2] \quad (3)$$

Utilizada nas três funções: R_2, R_5 ;

Domínio de busca: $-5 \leq x_j \leq 10, j = 1, \dots, n$;

1 mínimo global: $x = (1, \dots, 1)$;

Mínimo Global: $R_n(x^*) = 0$.



Taxa de sucesso (n= 2): 47%
 Avaliações função objetivo (n= 2): 610
 Não convergiu para n= 5

(a) Representação Gráfica

(b) Convergência

Figura 3 – Função Rosenbrock

A função de Rosenbrock (Eq. 3) (Figura 3a) é clássica e conhecida com função banana. O mínimo global encontra-se em um vale, longo, estreito, e de formato aplanado. Esta função de teste tem sido bastante utilizada para a avaliação de desempenho de algoritmos de otimização global [Cavalcanti 2004]. Para duas variáveis o AG_Real encontra o mínimo global em 47% dos casos, onde percebe-se uma dificuldade, e como esperado o algoritmo não convergiu para cinco variáveis pois quando se aumenta o número de variáveis aumenta também o espaço de busca, e como os parâmetros foram mantidos constantes para futura comparação, uma população de cinquenta indivíduos se torna pequena para varrer todo o espaço de busca. A convergência (Figura 3b) para o mínimo global é lenta devido á topologia da função, logo um aumento no número de gerações ajudaria no aumento da taxa de sucesso.

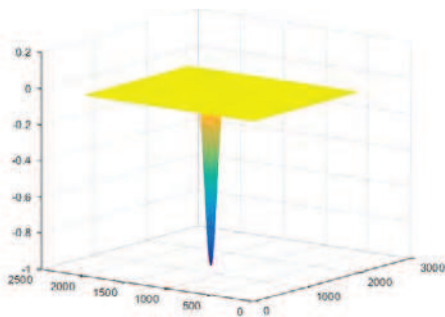
Função de Easom (ES):

$$ES(x_1, x_2) = -\cos(x_1) \times \cos(x_2) \times \exp \{ -[(x_1 - \pi)^2 + (x_2 - \pi)^2] \} \tag{4}$$

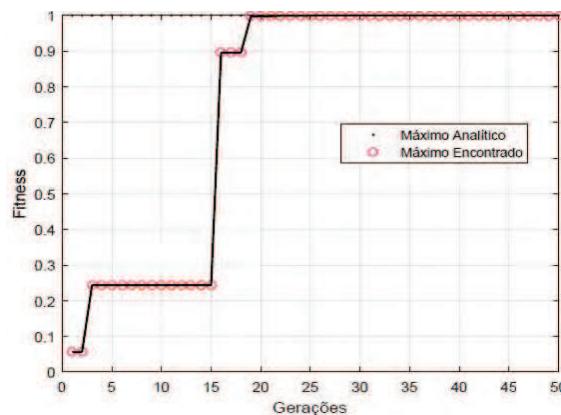
Domínio de busca: $-100 \leq x_j \leq 100, j = 1, 2;$

1 mínimo global: $(x_1, x_2)^* = (\pi, \pi);$

Mínimo Global: $ES[(x_1, x_2)^*] = -1.$



Taxa de sucesso: 73%
Avaliações função objetivo: 1040



(a) Representação Gráfica

(b) Convergência

Figura 4 – Função Easom

A função Easom (Eq. 4) (Figura 4a) que é uma função unimodal, ou seja, possui apenas um mínimo global, pode parecer simples de otimizar, porém o seu mínimo se encontra apenas em uma pequena área com relação ao seu ao seu espaço de busca, o que dificulta a otimização. Quando alguns indivíduos da população encontram essa pequena região do domínio a convergência se dá de forma rápida como mostrado nas gerações de quinze a vinte do gráfico de convergência (Figura 4b).

Função de Zakharov (R_n (variáveis)):

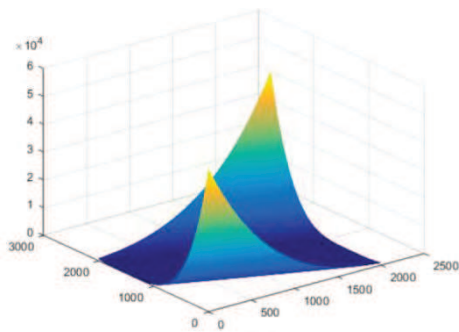
$$Z_n(x) = \left(\sum_{j=1}^n x_j^2 \right) + \left(\sum_{j=1}^n 0,5j \times x_j \right)^2 + \left(\sum_{j=1}^n 0,5j \times x_j \right)^4 \tag{5}$$

Utilizada nas três funções: $Z_2, Z_5;$

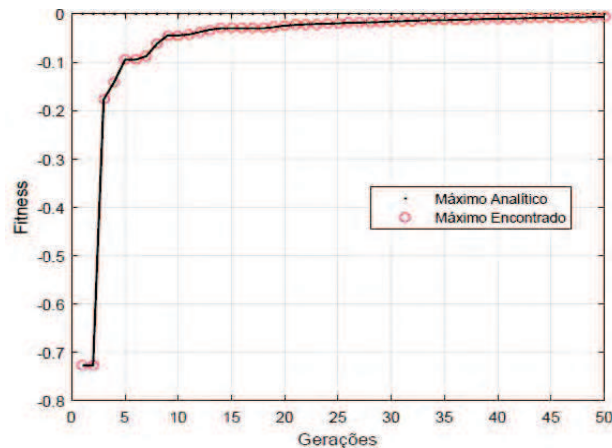
Domínio de busca: $-5 \leq x_j \leq 10, j = 1, \dots, n;$

1 mínimo global: $x^* = (0, \dots, 0);$

$Z_n(x^*) = 0.$



Taxa de sucesso (n= 2): 90%
 Avaliações função objetivo (n= 2): 540
 Não convergiu para n= 5



(a) Representação Gráfica

(b) Convergência

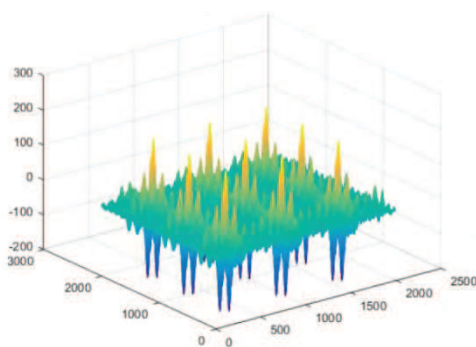
Figura 5 – Função Zakharov

A função de Zakharov (Eq. 5) (Figura 5a) é considerada relativamente simples para duas variáveis, como pode ser observado na Figura 5b porém a medida que aumentamos a quantidade de variáveis a topologia da função começa a ficar mais complexa assim também como o espaço de busca, dessa forma essa função com cinco variáveis representa um desafio multivariáveis para o algoritmo. Os melhores resultados do AG_Real foram obtidos nesta função com duas variáveis, porém a convergência não aconteceu para cinco variáveis devido a alta complexidade do problema.

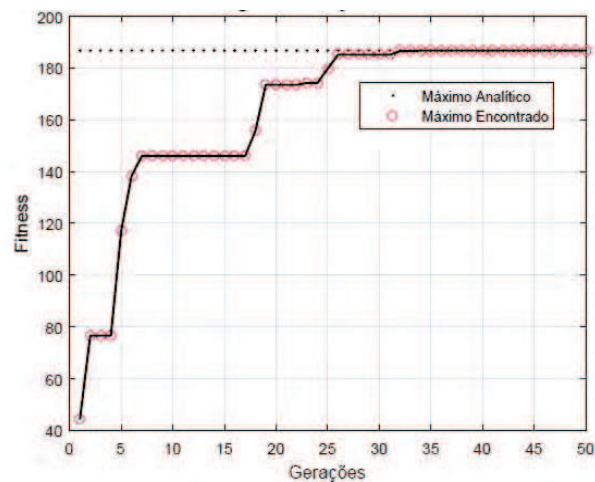
Função de Shubert (SH):

$$SH(x_1, x_2) = \left(\sum_{j=1}^5 j \cdot \cos[(j + 1) \cdot x_1 + j] \right) \cdot \left(\sum_{j=1}^5 j \cdot \cos[(j + 1) \cdot x_2 + j] \right) \quad (6)$$

Domínio de busca: $-10 \leq x_j \leq 10, j = 1, 2;$
 760 mínimos locais e 18 mínimos globais
 Mínimo Global: $SH[x_1, x_2] = -186.7309$



Taxa de sucesso: 51%
 Avaliações função objetivo: 1050



(a) Representação Gráfica

(b) Convergência

Figura 6 – Função Shubert

Shubert (Eq. 6) é uma função de teste multimodal (Figura 6a) de duas variáveis. Funções multimodais como essa são utilizadas em testes de desempenho de otimizadores, pois muitos problemas de planejamento possuem a característica de multimodalidade. O AG_Real convergiu em 51% das execuções; para fins de comparação os parâmetros genéticos foram mantidos os mesmos durante todos os testes. Porém no caso da função de Shubert, um aumento do número de gerações nos levaria a um aumento da taxa de sucesso, bem como no de avaliações da função objetivo. O gráfico de convergência mostra o algoritmo saindo de mínimos locais, dando esse formato de “escada” para a Figura 6b.

Comparação AG_Real, CGA e RCGA

O AG_Real é comparado com dois outros algoritmos: o RCGA (Real-Coded Genetic Algorithm) de Bessaou e Siarry [2001] e o CGA (Continous Genetic Algorithm) de Chelouah e Siarry [2000]. Na Tabela 3 são mostradas duas informações, primeiro o número médio de avaliações da função objetivo (FOBJ_{aval}) e depois a taxa de sucesso (TS).

Tabela 3 - Comparação da média do número de avaliações da função objetivo

AG	AG_Real		RCGA		CGA	
Função	FOBJ _{aval}	TS (%)	FOBJ _{aval}	TS (%)	FOBJ _{aval}	TS (%)
GP	695	87	270	100	410	100
R ₂	610	47	596	100	960	100
ES	1040	73	642	100	1504	100
Z ₂	540	90	437	100	620	100
SH	1050	51	946	100	575	100
R ₅	-	0	4150	60	3990	100
Z ₅	-	0	1115	100	1350	100

Quando comparado ao RCGA, o AG_Real não conseguiu melhores resultados de taxa de sucesso e número de avaliações da função objetivo. Em relação ao CGA, o número de avaliações da função objetivo do AG_Real foi menor para as seguintes funções: Rosenbrock de duas variáveis, Easom e Zakharov de duas variáveis.

Ambos os algoritmos utilizam técnicas adjacentes para melhoria da aptidão de cada geração, tais como: composição de subpopulações baseado na teoria de entropia, decrescimento da probabilidade de mutação e da quantidade de indivíduos na população, existe a proposição de um novo método de *crossover*, técnicas de diversificação e intensificação de busca, etc. O AG_Real utiliza operadores simples, onde apenas o elitismo entra como técnica de auxílio à convergência.

As técnicas empregadas no RCGA e CGA, se justificam pela análise das funções Zakharov e Rosenbrock de cinco variáveis, pois o AG_Real não convergiu com os parâmetros configurados, logo técnicas como decrescimento da probabilidade de mutação é importante no controle da pressão de seleção.

6. Conclusão

Após revisão de literatura percebeu-se a importância dos otimizadores probabilísticos. O algoritmo genético elitista de codificação real desenvolvido neste trabalho convergiu para cinco das sete funções de *benchmark* utilizadas na análise, com taxa de sucesso entre 47% e 90%. A implementação foi realizada com operadores simples, onde apenas o elitismo entra como técnica de auxílio à convergência.

Os experimentos de comparação serviram para a análise da adequação dos algoritmos à complexidade das funções e mostrar a necessidade do uso de outras técnicas de acordo com a complexidade de cada problema. Os algoritmos selecionados para a comparação utilizaram técnicas adjacentes tais como composição de subpopulações baseado na teoria de entropia,

decréscimo da probabilidade de mutação e da quantidade de indivíduos na população, novo operador de *crossover*, técnicas de diversificação e intensificação de busca.

Os resultados apresentados para sete funções de *benchmark*, através de uma formulação matemática, sua representação gráfica, gráfico de convergência, taxa de sucesso e número de avaliações da função objetivo, permitem uma melhor visualização dos tipos de problemas de otimização e, também, da adequação dos algoritmos genéticos para resolvê-los.

As funções de *benchmark* representam neste artigo a complexidade de problemas reais, e os três AG's comparados demonstram que os algoritmos genéticos podem ser escolhidos como ferramenta nas soluções de problemas complexos, que envolvem muitas variáveis e restrições, para os quais as ferramentas determinísticas tornam-se inviáveis.

Agradecimentos: Este trabalho teve apoio financeiro da Fundação de Amparo à Pesquisa do Estado de Goiás (FAPEG).

Referências:

Ariyarit, A. e Kanazaki, M. (2015). Multi-modal distribution *crossover* base on two crossing segments bounded by selected parents applied to multi-objective design optimization. Journal of Mechanical Science and Technology (29) 4 p. 1443-1448.

Bessaou, M. e Siarry, P. (2001). A genetic algorithm with real- value coding to optimize multimodal continuous functions. Struct Multidisc Optim, v. 23, n. 1, p. 63-74. ISSN 1615-147X.

Calixto, W. P. (2008). Aplicação do Mapeamento Conforme no Cálculo do Fator de Carter. Dissertação (Mestrado em Engenharia Elétrica) – Escola de Engenharia Elétrica e de Computação. Goiânia: Universidade de Goiás.

Catarina, A. S. e Bach, S. L. (2003). Estudo do efeito dos parâmetros genéticos sobre a solução otimizada e sobre o tempo de convergência em algoritmos genéticos com codificações binária e real. Acta Scientiarum Technology: Maringá, v. 25 n. 2, p. 147 -152.

Chang, L.-C. et. al. (2010). Constrained genetic algorithms for optimizing multi-use reservoir operation. Journal of Hydrology. Elsevier B.V, v. 390, p. 66-74.

Chelouah, R. e Siarry, P. (2000). A Continuous Genetic Algorithm Designed for the Global Optimization of Multimodal Functions. Journal of Heuristics, v. 6, n. 2, p. 191-213. ISSN 1381-1231.

Chen, Z.-Q. e Wang, R.-L. (2011). Two efficient real-coded genetic algorithms for real parameter optimization. International Journal of Innovative Computing, Information and Control – ICIC, vol. 7, n. 8, ago.

Cirino, R. B. Z. et al. (2015). Aplicação da metaheurística *agc* para o problema de alocação de aulas à salas. XLVII Simpósio Brasileiro de Pesquisa Operacional.

Couëtoux, B. et al. (2010). Labeled Traveling Salesman Problems: Complexity and approximation. Discrete Optimization, v. 7, n. 1-2, p. 74-85. ISSN 15725286.

Doerr, B. et al. (2012). More effective *crossover* operators for the all-pairs shortest path problem. Theoretical Computer Science. ISSN 0304-3975.

Fischer, I. (2008). “Using Genetic Algorithms for Simulation of Social Dilemmas” in New Issues and Paradigms in Research on Social Dilemmas, pp. 252-264.

- Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley Publishing Company, Inc. Universidade do Alabama, Reading, MA, 412.
- Gordon, M. (1988). Probabilistic and genetic algorithm in document retrieval. Communications of the ACM, v. 31 p. 1208-1218.
- Herrera, F. et al. (1998). Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. Artificial Intelligence Review, v. 12, n. 4, p. 265-319. ISSN 1573-7462. Available at: < <http://dx.doi.org/10.1023/A:1006504901164> >.
- Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. MIT Press.
- Jamil, M. e Yang, X.-S. (2013) A literature survey of *benchmark* functions for global optimisation problems. International Journal of Mathematical Modelling and Numerical Optimisation, v. 4, n. 2, p. 150-194. ISSN 2040-3607.
- Jong, K. A. D. (1975). An analysis of the behavior of a class of genetic adaptive systems. 266 University of Michigan
- Lienig, J. e Thulasiraman, K. (2007). A Genetic Algorithm for Channel Routing in VLSI Circuits. Evolutionary Computation, 1(4):293-311.
- Linden, R. (2012). Algoritmos Genéticos. 3ª ed. Ciência Moderna.
- Tosatti M. A. et al. (2008). Algoritmo híbrido genético-fuzzy aplicado em câncer de próstata, Hifen, 32 (62) (2008) 164-171.
- Cavalcanti, A. M. (2004). Desenvolvimento e análise de algoritmos probabilísticos de otimização global.
- Miller, B. L. e Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. Complex Systems, v. 9, n. 3, p. 193-212. ISSN 0891-2513.
- Molga, M. e Smutnicki, C. (2005). Test functions for optimization needs.
- Mollinetti, M. A. F. et. al. (2013). Análise de Performance do Imperialist Competitive Algorithm através de Funções de *Benchmark*. In: Encontro Anual de Tecnologia da Informação e Semana Acadêmica de Tecnologia da Informação – EATI, ano 3, n. 1. Frederico Westphalen: EATI, p. 13-21.
- Peralta, R. et al. (2014). Multiobjective genetic algorithm conjunctive use optimization for production, cost, and energy with dynamic return flow. Journal of Hydrology, n. 511, p. 776-785.
- Serrada, A. P. (1996). Una Introducción a la Computación Evolutiva. Universidade de Oviedo, Espanha.
- Soni, N. e Kumar, T. (2014). Study of Various Mutation Operators in Genetic Algorithms. In: International Journal of Computer Science and Information Technologies. Universidade de Lingaya's. Faridabad: vol. 5, n. 3, p. 4514-4521.

As soluções para estes problemas ocorrem nas técnicas heurísticas de otimização, que proporcionam soluções boas, mas não garante que a solução encontrada seja a ótima. Assim, tais métodos, baseados na busca aleatório-controlada por critérios probabilísticos, nos últimos anos tiveram grandes avanços devido aos recursos computacionais, uma vez que necessitam de um número elevado de avaliações da função objetivo. A busca por uma solução para o problema permitirá o acesso a tópicos de estudo bastante atuais, na área de otimização combinatorial, cujo estado da arte aponta para os algoritmos genéticos (Goldberg, 1989).

A introdução dos algoritmos genéticos foi primeiramente realizada por John Holland em 1975 (Holland, 1975), com o objetivo de ser uma ferramenta para a resolução de problemas de otimização de grande complexidade, como por exemplo, o problema do caixeiro viajante. Baseiam-se no processo da seleção natural proposto por Charles Darwin, no qual uma população de indivíduos irá passar por um processo de seleção, em que os mais adaptados às condições do meio têm mais chances de repassar suas características para a próxima geração através do cruzamento, gerando filhos, que ainda podem sofrer um processo de mutação. Seleção, cruzamento e mutação são chamados de operadores genéticos, responsáveis por aprimorar a solução no decorrer das gerações do algoritmo.

Cada possível solução de um problema é codificada em uma estrutura chamada de "cromossomo", que é composta por uma estrutura de dados que descreve os genes. Estes cromossomos representam indivíduos, que são evoluídos ao longo de várias gerações, analogamente aos seres vivos, de acordo com os princípios de seleção natural e sobrevivência dos mais aptos, descritos pela primeira vez por Charles Darwin em seu livro "A Origem das Espécies".

A SNDlib (*Survivable Network Design Library*) é uma biblioteca com instâncias e dados de redes de telecomunicações sendo apresentada pela primeira vez, em 2005, na Conferência Internacional DRCN (*Design of Reliable Communications Networks*) por (Orlowski, Wessály, Pióro, & Tomaszewski, 2010). A biblioteca disponibiliza dados de projetos de redes fixas de telecomunicações cuja finalidade é fazer com que instâncias de rede estejam disponíveis para a comunidade científica servindo como um ponto de referência para que pesquisadores e profissionais da área possam testar avaliar e comparar modelos de projetos de rede e algoritmos (Zuse Institute Berlin (ZIB), 2005).

1.1 Trabalhos Correlatos

Hillier e Lieberman (Hillier & Lieberman, 2009) em seu livro, descrevem a evolução das redes de telecomunicações. Apresentam metodologias e técnicas capazes de resolver problemas de otimização com melhor eficiência de modo a permitir soluções de problemas intratáveis de décadas atrás.

Para solucionar diversos problemas em sistemas de telecomunicações, novas propostas e novas técnicas são apresentadas na literatura. Atualmente estudos no campo de projetos hierárquicos de sobrevivência de rede são aplicados. O trabalho de (Rodríguez-Martín, Salazar-González, & Yaman, 2016) que desenvolveu um algoritmo de ramificação e corte e o seu comportamento foi testado em instâncias da rede de biblioteca SNDlib.

Para a interconexão de redes móveis celulares, os trabalhos de (Baumgartner, Reddy, & Bauschert, 2015) apresentou um novo modelo de otimização matemática para encapsulamentos de redes de núcleo móveis virtuais objetivando os limites de latência diferenciando de modelos clássicos de incorporação de rede virtual que atribui equipamentos tais como *gateways* de núcleo único, sujeitas a otimização.

Em seu outro trabalho, (Baumgartner, Reddy, & Bauschert, Combined Virtual Mobile Core Network Function Placement and Topology Optimization with Latency Bounds, 2015), refere-se para a virtualização de rede como um ponto chave para novos estudos e pesquisa objetivando maior flexibilidade, adaptabilidade e capacidade de gerenciamento para futuras redes de comunicação.

(Tripathy, Dash, & Tripathy, 2012) abordaram a confiabilidade dos componentes dos sistemas ao projetar redes de telecomunicações e que problemas de otimização topológica é definido como a existência de pelo menos um caminho confiável entre um par de nós que satisfazem o custo predefinido da rede.

(Altıparmak, Gen, Lin, & Paksoy, 2006) apresentaram uma abordagem de algoritmo genético para a otimização multi-objetivo aplicado em redes da cadeia de abastecimento com concepções e análises para determinar a eficiência de um sistema, comparar sistemas alternativos e projetar sistemas propostos.

Categorizaram as medidas em qualitativas de forma a satisfazer o cliente visando a flexibilidade e a gestão eficaz do risco enquanto que as medidas quantitativas possuem objetivos baseados diretamente no custo das redes, tais como: minimização de custos, maximização de vendas e lucro e os objetivos baseados

em medidas de capacidade de resposta ao cliente que envolve a maximização da taxa, minimização do tempo de resposta do cliente, minimização do *lead time* e outros (Altıparmak, Gen, Lin, & Paksoy, 2006).

Na área de inteligência computacional, os Algoritmos Genéticos são métodos de otimização global inspirados nos processos genéticos de organismos biológicos (Couëtoux, Gourvès, Monnot, & Telelis, 2010).

(Peralta, Forghani, & Fayad, 2014) afirmam que muitos problemas reais de otimização de recursos hídricos apresentam objetivos conflitantes e que os Algoritmos Genéticos Multiobjetivos (MGA) têm sido previamente propostos para superar dificuldades.

Sendo assim, os estudos poderão ser aplicados a outros problemas de distribuição em rede de gás para a concepção e dimensionamento de redes de gasodutos de transmissão de hidrogênio (André, et al., 2013).

Outros projetos de diversas áreas são apresentados por (Lienig & Thulasiraman, 1993) que aplicou um algoritmo genético para roteamento de canais em circuitos VLSI.

2. METODOLOGIA

Um problema de otimização da biblioteca consiste de duas partes: dados da rede (nós, *links*, demandas, capacidades, parâmetros de custo e outros) e dados do modelo especificando os parâmetros de otimização (*links* direcionais ou não, divisão de demanda, sobrevivência e etc.). As instâncias da SNDlib são dadas em uma formulação arco-caminho da seguinte forma: $G = (V, E, D)$ onde V é o número de nós, E número de arcos e D as demandas a serem escoadas entre os nós.

Os *links* de uma rede são possíveis conexões entre nós que devem ser utilizadas para transportar tráfego. O conjunto de demandas representa o tráfego a ser roteado, sendo que cada demanda possui um nó de origem e um nó de destino. Para rotear as demandas é utilizado um conjunto caminhos admissíveis para escoar determinada demanda, esses caminhos são descritos como uma lista de *links* sequenciais com nós origem e destino iguais ao da demanda correspondente. Para cada *link* da rede existem modularidades disponíveis para instalação.

O problema consiste em decidir os caminhos para escoar as demandas previstas de forma que o custo de implantação da rede seja o menor possível. Obedecendo restrições de capacidade técnica e satisfação de demanda.

Foi desenvolvido um algoritmo genético para o problema de planejamento de redes de telecomunicações intitulado NDGA (*Network Design Genetic Algorithm*) descrito na seção 3. Esse algoritmo resolve instâncias da biblioteca SNDlib: as demandas e os *links* devem obedecer a mesma forma de roteamento sendo não direcionais, a capacidade dos *links* é explícita, não possui custo fixo de instalação, as demandas podem ser escoadas apenas por um caminho, todos os caminhos existentes podem ser utilizados, e não existe sobrevivência da rede. A Tabela 1 resume essas informações com as instâncias consideradas.

Tabela 1. Instância da Biblioteca de redes SNDlib (U-U-E-N-S-A-N-N)

Filtro de Modelo	
Modelo de Demanda: Demanda não direcional	(U)
Modelo de <i>Link</i> : <i>Links</i> não direcionais	(U)
Modelo de Capacidade do <i>Link</i> : Capacidades de <i>links</i> explícitos	(E)
Modelo de carga fixa: Sem custo de carga fixa	(N)
Modelo de roteamento: Caminho Único	(S)
Modelo de caminho admissível: Todos os caminhos	(A)
Modelo de <i>hop-limit</i> : Sem <i>hope limits</i>	(N)
Modelo de sobrevivência: Sem sobrevivência	(N)

Fonte: Zuse Institute Berlin (2005).

A formulação matemática com as funções objetivo e restrições foram definidas por (Orlowski, Wessály, Pióro, & Tomaszewski, 2010). Foram realizadas simplificações considerando a instância U-U-E-N-S-A-N-N, descritas a seguir. A Função Objetivo (1): representa o custo de implantação da rede.

$$\min \sum_{e \in \mathcal{E}} \sum_{t \in T_e} k_e^t \cdot y_e^t \quad (1)$$

Onde:

- ε : Conjunto de *links* de escoamento de demanda da rede. Representa todas as possíveis conexões que podem ser feitas entre os nós da rede.
- e : Elemento do conjunto de *links* da rede [$e \in \varepsilon$].
- T_e : Conjunto das modularidades disponíveis para a instalação. Cada *link* deve ser associado a uma modularidade dentro deste conjunto.
- t : Elemento do conjunto de modularidades da rede.
- k_e^t : Custo do módulo t instalado no *link* e . Representa o custo de alocação de uma capacidade em um determinado *link*.
- y_e^t : Variável binária que representa a escolha da modularidade t instalada no *link* e . Decide se o custo k_e^t será contabilizado ou não.
- T_e : Conjunto das modularidades disponíveis para a instalação
- Y_e : Fluxo de demanda em canais E1 no arco e
- t : Elemento do conjunto de modularidades da rede

Restrição de capacidade técnica (2): a capacidade instalada em um *link* deve ser maior que a soma das demandas que estão sendo escoadas através desse *link*.

$$\sum_{e \in \varepsilon} \sum_{t \in T_e} t \cdot y_e^t \geq Y_e \quad (2)$$

Onde:

- Y_e : Fluxo de demanda em canais E1 no arco e . É soma de todas as demandas escoadas em um *link*.

O fluxo máximo de demanda que pode passar por um arco é determinado pela máxima modularidade disponível. Dessa forma, caso a soma das demandas em arco ultrapasse esse valor, a solução é dita infactível visto que não existem módulos a serem instalados.

Para a apresentação e teste do NDGA foi escolhida uma rede PDH (*Plesiochronous Digital Hierarchy*) que proporciona a intercalação sequencial de *bytes* podendo a ter níveis hierárquicos de 565,148 Mbps, com 8192 canais E1, (Babatunde & Mbarouk, 2014). A rede PDH candidata é composta de 11 nós (vértices) e 34 *links* (arestas), conforme Figura 1.

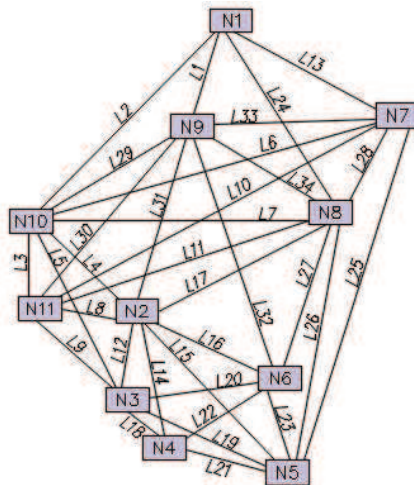


Figura 1. Rede Candidata – Topologia. Fonte: Zuse Institute Berlin (2005)

No total, 24 demandas previstas devem ser atendidas integralmente, conforme Tabela 2.

Tabela 2. Valores de demanda para cada demanda

Demanda Prevista	Nó (origem)	Nó (destino)	Valor médio em canais E1	Demanda Prevista	Nó (origem)	Nó (destino)	Valor médio em canais E1
D1	N9	N2	384	D13	N5	N8	108
D2	N3	N4	215	D14	N4	N5	105
D3	N7	N8	212	D15	N9	N8	100
D4	N10	N3	197	D16	N11	N3	99
D5	N9	N11	160	D17	N5	N6	95
D6	N2	N8	160	D18	N2	N3	278
D7	N2	N5	144	D19	N11	N2	266
D8	N1	N7	138	D20	N9	N10	263
D9	N3	N5	124	D21	N1	N9	258
D10	N6	N8	115	D22	N9	N7	247
D11	N1	N10	115	D23	N2	N6	237
D12	N10	N11	364	D24	N2	N4	237

Fonte: Zuse Institute Berlin (2005).

O custo da rede apresentado na equação 1 é calculado de acordo com a Tabela 3, que mostra o custo de instalação de cada modularidade em canais E1 (30, 480 e 1920) em relação a cada *link*.

Tabela 3. Dados da rede PDH

Link	Custo (30 E1)	Custo (480 E1)	Custo (1920 E1)	Link	Custo (30 E1)	Custo (480 E1)	Custo (1920 E1)
L1	88.933	691.901	1.625.091	L18	70.159	589.043	1.330.671
L2	149.283	988.231	2.498.805	L19	124.638	869.656	2.147.172
L3	59.431	530.267	1.162.431	L20	112.978	813.556	1.980.808
L4	106.353	781.681	1.886.283	L21	103.438	767.656	1.844.692
L5	129.408	892.606	2.215.230	L22	102.643	763.831	1.833.349
L6	185.588	1.162.906	3.016.802	L23	91.168	704.146	1.660.141
L7	164.388	1.060.906	2.714.322	L24	132.588	907.906	2.260.602
L8	94.297	721.289	1.709.211	L25	187.973	1.174.381	3.050.831
L9	114.038	818.656	1.995.932	L26	151.933	1.000.981	2.536.615
L10	190.093	1.184.581	3.081.079	L27	119.603	845.431	2.075.333
L11	164.918	1.063.456	2.721.884	L28	93.403	716.391	1.695.191
L12	84.016	664.962	1.547.981	L29	121.193	853.081	2.098.019
L13	121.988	856.906	2.109.362	L30	130.203	896.431	2.226.573
L14	107.148	785.506	1.897.626	L31	124.638	869.656	2.147.172
L15	139.743	942.331	2.362.689	L32	150.343	993.331	2.513.929
L16	113.773	817.381	1.992.151	L33	125.698	874.756	2.162.296
L17	136.828	928.306	2.321.098	L34	117.218	833.956	2.041.304

Fonte: Zuse Institute Berlin (2005).

3. ALGORITMO GENÉTICO PROPOSTO

O algoritmo genético proposto intitulado *Network Design Genetics Algorithm* (NDGA) especifica como combinações de componentes devem ser tomadas através de um conjunto de regras de um processo matemático.

Desta forma, o modelo é constituído para que os operadores de mutação e crossover possam combinar aleatoriamente variáveis e dados e definir conjuntos eficazes de processos descartando dados ineficientes. Este último processo do AG é conhecido como seleção (*fitness* do inglês), onde os melhores indivíduos são escolhidos dentro do espaço de busca (Linden, 2012).

3.1 Codificação da Rede

Antes da escolha dos operadores é necessário primeiramente definir como as propostas de solução do problema (os indivíduos) serão representadas, ou seja, definir a codificação do problema. A codificação mais

comum é a codificação binária que foi utilizada nos trabalhos iniciais de Holland, mesmo não sendo computacionalmente eficiente, possui relevância histórica e tem sido utilizada quanto a sua simplicidade. A codificação deve ser escolhida de forma que não haja perda de informações no processo de codificação e decodificação. O Fluxograma a seguir mostra 03 grandes partes: a importação de dados, a geração da população e os operadores genéticos.

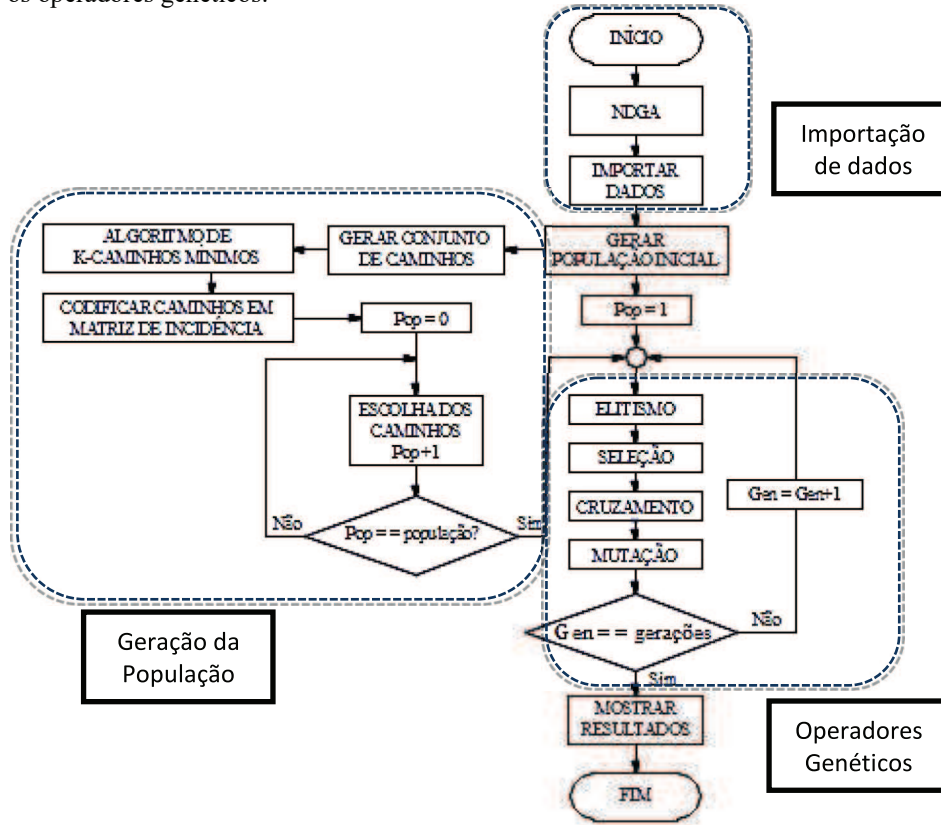


Figura 2. Fluxograma do algoritmo proposto

A codificação da rede é feita através de uma matriz de incidência de ordem $N \times L$, onde N é o número de nós e L é o número de *links*. A matriz é composta de 0's e 1's pois os *links* existentes são bidirecionais, onde 0 representa ausência de *link* e 1 mostra a incidência do *link* em determinado nó. As soluções da rede são uma matriz de ordem $D \times L$, onde D é o número de demandas a serem escoadas conforme apresentada na Figura 3.

	L1	L2	L3	L4	L5	...	L30	L31	L32	L33	L34
D1	0	0	0	0	0		0	384	0	0	0
D2	0	0	221	0	377		322	0	0	0	0
D3	0	0	0	195	0		0	0	0	212	212
:											
D22	0	0	124	0	0		222	0	0	247	0
D23	0	0	0	128	0		0	66	0	0	0
D24	0	0	0	0	302		0	0	0	0	0
	121	297	345	323	679		544	450	670	459	212

Figura 3. Codificação das soluções

Cada demanda deve ser escoada por um caminho que é formado por um conjunto de *links* do nó origem até o nó destino. Assim cada linha da matriz representa os *links* utilizados para escoar determinada demanda, sendo que a coluna correspondente ao *link* utilizado é preenchida com o número 1.

3.2 Aptidão e Seleção

O custo total da rede é utilizado como função de aptidão. Cada arco da rede possui um custo devido à fibra óptica e a modularidade do equipamento escolhido para transmissão de dados. A biblioteca SNDLib fornece uma tabela com esse custo para cada arco da rede, dessa forma caso um arco seja utilizado em uma solução o seu custo é contabilizado.

A seleção dos indivíduos é feita pelo método do torneio. Dois indivíduos são selecionados aleatoriamente, os seus custos são comparados e a solução de menor custo é então selecionada para passar suas informações para a próxima geração por meio do crossover. Esse método é simples, porém eficiente, e não possui custo computacional excessivo, já que o custo é parte do indivíduo e não precisa ser calculado unicamente para a seleção (Miller & Goldberg, 1995).

3.3 Operador de *Crossover*

O *crossover* é o operador pelo qual bons indivíduos repassam suas características para a próxima geração. Em conjunto com o operador de seleção garante que a busca do algoritmo genético não seja totalmente aleatória.

O *crossover* mais encontrado na literatura, também utilizado por Holland é *crossover* de um ponto para codificação binária, onde um indivíduo (cadeia de *bits*) é cortado em um ponto aleatório; e após o corte dois indivíduos têm suas informações cruzadas, produzindo dois novos indivíduos (filhos) que irão compor a próxima geração.

No NDGA é utilizado o *crossover* de n pontos. Os pais que realizarão cruzamento são cortados em n pontos aleatórios e têm suas informações genéticas trocadas, são gerados dois filhos, porém apenas o indivíduo de menor custo entra para a próxima geração conforme apresentado na Figura 4.

Os filhos gerados podem ainda ser ineficazes devido à capacidade máxima disponível de canais E1, se isto acontecer, o processo é feito novamente até que se consiga um indivíduo factível. Este operador é controlado por uma probabilidade que define a presença ou ausência do operador. Caso não haja *crossover* o indivíduo de uma geração é apenas copiado para a próxima.

3.4 Operador de Mutação

Os algoritmos genéticos realizam uma busca aleatório-direcionada, e a mutação garante o caráter aleatório da busca, (Goldberg, 1989).

No decorrer das gerações a população pode convergir para um ponto ótimo local e assim o *crossover* não consegue retirar a população estagnada do local, dessa forma, é necessário que aconteça mutação para que isto não ocorra.

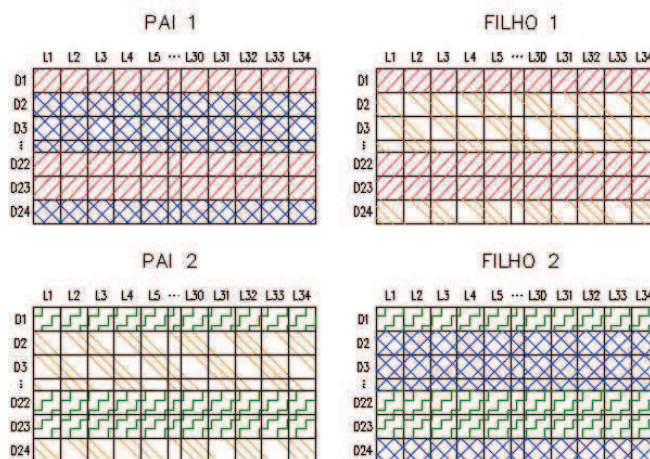


Figura 4. Crossover de n pontos

A mutação também é controlada por uma taxa, que deve ser menor que a do *crossover*, ao contrário o processo de busca torna-se totalmente aleatório (Soni & Tapas, 2014).

O processo de mutação mais comum em uma cadeia de *bits*, é trocar um *bit* 1 por um *bit* 0. Para o NDGA, foi proposto um novo operador, com base no descrito acima. Cada linha representa o caminho pelo qual uma demanda está sendo escoada e o processo de mutação consiste em trocar este caminho por outro aleatoriamente escolhido dentro dos conjuntos disponíveis de caminhos (Figura 5). O número de trocas a serem realizadas é parte dos parâmetros a serem definidos no NDGA.

	L1	L2	L3	L4	L5	...	L30	L31	L32	L33	L34
D1											
D2	0	0	221	0	377		322	0	0	0	0
D3	0	0	0	195	0		0	0	0	212	212
⋮											
D22											
D23											
D24	0	0	0	0	302		0	0	0	0	0


 CAMINHO ALTERADO

Figura 5. Mutação com a ocorrência de 03 trocas

3.5 Elitismo

O elitismo não ocorre no processo natural de evolução, por isso é tido como técnica de auxílio à convergência (Goldberg, 1989). Baseia-se em manter a melhor solução encontrada em uma geração para a seguinte, para que não aconteça a perda de boas soluções (Liang & Leung, 2011). Dessa forma o algoritmo genético guarda o indivíduo de melhor custo para a próxima geração. A análise de convergência do NDGA é feito com base no indivíduo elitista da última geração.

4. EXPERIMENTOS

Os resultados encontrados pelo NDGA serão comparados com os disponíveis na biblioteca SNDlib para a instância selecionada. Pelo fato de serem probabilísticos os AG's podem não encontrar a mesma solução mesmo que os parâmetros genéticos sejam mantidos. Sendo assim foram realizados 100 testes para um total de 11 configurações diferentes de parâmetros (Tabela 4). O conjunto de testes para cada configuração é chamado de experimento, e o resultado final de cada experimento corresponde ao melhor resultado encontrado entre os testes realizados.

Tabela 4. Configuração dos Experimentos

Experimentos	População	Gerações	Crossover (Probabilidade/Cortes)	Mutação (Probabilidade/Trocas)
1	100	100	0,9 / 5	0,06 / 2
2	200	100	0,9 / 5	0,06 / 2
3	300	100	0,9 / 5	0,06 / 2
4	400	100	0,9 / 5	0,06 / 2
5	400	100	0,9 / 5	0,06 / 2
6	100	100	0,9 / 5	0,06 / 2
7	200	200	0,9 / 5	0,06 / 2
8	300	300	0,9 / 5	0,07 / 2
9	400	400	0,9 / 5	0,08 / 2
10	500	500	0,9 / 5	0,08 / 2
11	1000	500	0,95 / 10	0,08 / 5

A escolha de cada parâmetro deve ser feita considerando as características e complexidade de cada problema especificamente, pois não existe um conjunto de parâmetros que produzem bons resultados para todos os tipos de problemas. Porém existem valores que se tornaram clássicos na literatura, como por exemplo, probabilidade de crossover e mutação, pois em muitos problemas pequenas alterações nessas probabilidades em torno do valor clássico não alteram significativamente o desempenho do AG. Para o problema de redes aqui considerado o espaço de busca é vasto, de forma de que o tamanho da população e a quantidade máxima de geração do NDGA interferem significativamente no resultado. Após vários testes verificou-se o conjunto de parâmetros que produzem bons resultados para a otimização da rede PDH (Bessaou & Siarry, 2001), assim os experimentos foram configurados com valores para que se tenha uma visualização do esforço computacional necessário para a otimização deste problema.

Os valores de aptidão de todos os indivíduos são conhecidos, porém a análise de convergência é feita através das informações do indivíduo mais apto. Como os indivíduos são avaliados de acordo com seu custo, esse dado é utilizado para gerar uma curva de convergência que mostra evolução da melhor solução no decorrer das gerações. O resultado final do NDGA é o indivíduo mais apto da última geração, e o resultado de cada experimento é o melhor teste realizado entre as 100 execuções.

5. RESULTADOS OBTIDOS

Nos relatórios da SNDlib são disponibilizadas seis informações principais: o custo da rede, o número de arcos utilizados, a capacidade total em canais E1 instalada na rede, a porcentagem utilizada dessa capacidade, o comprimento médio dos caminhos utilizados para escoar as demandas e o tempo gasto na otimização. A Tabela 5 mostra os resultados disponíveis na biblioteca e os obtidos para os 11 experimentos realizados para fim de comparação.

Nos resultados disponíveis na SNDlib, dos 34 *links* disponíveis 13 foram instalados sendo 8 de 480E1 e 5 de 1920E1. Para os experimentos realizados o NDGA utilizou 17 *links* com modularidades de 480E1. O maior uso de *links* pode gerar maiores custos, porém as modularidades de 1920E1 que interferem significativamente no custo não foram utilizadas, garantindo uma redução de custo de 15,15% nos experimentos 4, 8, 9, 10 e 11, em comparação com a solução da biblioteca. A capacidade utilizada em valores percentuais mostra a melhor utilização da capacidade instalada.

Tabela 5. Resultados das simulações e da biblioteca SNDlib

Experimento	Custo (\$)	Carga Instalada (Canais E1)	Capacidade Utilizada (%)	Tamanho médio do caminho	Tempo (s)	
pdh ¹	14883708	13440	65,08	2,00	3600*	
1	13353844	8160	74,01	1,38	0,70	
2	13398469	8160	67,11	1,33	1,40	
3	13138609	8160	68,62	1,29	2,86	
4	12925684	8160	72,48	1,38	3,00	
Valores Simulados do algoritmo proposto NDGA	5	13290334	8160	67,76	1,29	4,27
6	13738754	9600	64,66	1,46	0,88	
7	13246744	8160	71,32	1,38	3,52	
8	12966484	8160	70,67	1,38	8,39	
9	12925684	8160	72,48	1,38	17,05	
10	12925684	8160	72,48	1,38	18,97	
11	12925684	8160	72,48	1,38	52,61	

¹ Resultado da biblioteca SNDlib

* Tempo limite

O tempo gasto na otimização é importante tanto quanto o custo final da rede, pois na resolução de problemas ainda mais complexos é necessário que a otimização seja feita em tempo hábil, de acordo com as definições do usuário. Os resultados mostram que o tempo de resolução foi bem inferior ao da biblioteca, segundos ao invés de horas, garantindo que o NDGA possa ser utilizado em redes com quantidades maiores de nós, arcos e demandas. A biblioteca de redes SNDlib utiliza o limite de tempo de 3600s para apresentar a solução conforme Tabela 5.

A melhor solução encontrada foi a apresentada no experimento 4 que chegou ao menor tempo devido ao tamanho da população e número de gerações. O gráfico de convergência (Figura 6) mostra a evolução da solução no decorrer das gerações.

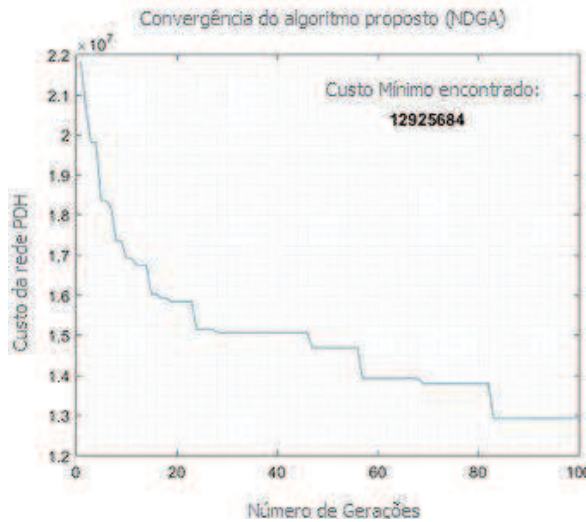


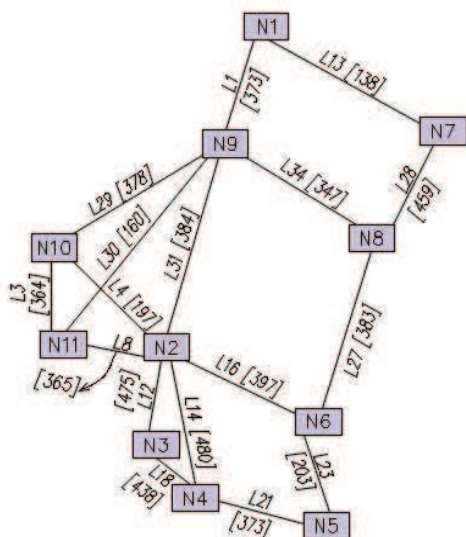
Figura 6. Convergência do algoritmo proposto

A Tabela 6 apresenta os detalhes dos resultados, também para o experimento 4.

Tabela 6. Resultados do experimento 4

Custo:	12925684
Links Utilizados:	17
Capacidade Instalada:	8160 E1
Capacidade Utilizada:	72,48 (%)
Tamanho médio dos caminhos:	1,375
Tempo Gasto:	3 s

Os links utilizados e os respectivos valores de demanda em canais E1 passando por eles são mostrados na Figura 7 da rede otimizada.



Legenda:

Lx - número do link que interliga 02 nós;

[xE1] - Demanda prevista para o nó em canais E1.

Figura 7. Rede otimizada

6. CONCLUSÕES

O planejamento estratégico e a tomada de decisões são fundamentais para solucionar problemas de fluxo e garantir maior viabilidade e confiabilidade no transporte de informações e dados. Os projetos de redes de telecomunicações são altamente influenciados pela quantidade de nós e arcos, pela demanda a ser escoada e pelas modularidades disponíveis para alocação, tornando o planejamento da rede altamente complexo.

Para o design da rede foi proposto um algoritmo genético como alternativa aos métodos determinísticos que trabalha com uma população de possíveis soluções da rede, e através dos operadores genéticos realiza a melhora da solução.

As simulações realizadas permitem uma visualização do esforço computacional necessário para a otimização da rede e também o conjunto de parâmetros ótimos a serem utilizados para o problema específico tratado no artigo. Após ser submetido à otimização de uma rede PDH o algoritmo se mostrou eficiente, pois apresentou uma redução do custo de até 15,15%, quando comparado ao valor disponível na biblioteca SNDlib. Introdução de lógica *fuzzy* e escoamento contínuo de demanda serão realizados em trabalhos futuros.

AGRADECIMENTOS

Os autores agradecem à Fundação de Amparo à Pesquisa do Estado de Goiás (FAPEG) e ao Serviço Nacional de Aprendizagem Industrial do Estado de Goiás (SENAI-GO) pelo apoio financeiro.

REFERÊNCIAS

- Altıparmak, F., Gen, M., Lin, L., & Paksoy, T. (22 de Agosto de 2006). A genetic algorithm approach for multi-objective optimization of supply chain networks. *Computers & Industrial Engineering*, pp. 196–215.
- André, J., Auray, S., Brac, J., De Wolf, D., Maisonnier, G., Ould-Sidi, M.-M., & Simonnet, A. (Março de 2013). Design and dimensioning of hydrogen transmission pipeline networks. *European Journal of Operational Research*, 229, 239-251.
- Babatunde, O., & Mbarouk, S. (Junho de 2014). A review of Plesiochronous Digital Hierarchy (PDH) and Synchronous Digital Hierarchy (SDH). *International Journal of Scientific Research Engineering & Technology (IJSRET)*, 3, pp. 677-681.
- Baumgartner, A., Reddy, V. S., & Bauschert, T. (2015). Combined Virtual Mobile Core Network Function Placement and Topology Optimization with Latency Bounds. *IEEE Communication Networks*, pp. 97-102.
- Baumgartner, A., Reddy, V. S., & Bauschert, T. (2015). Mobile Core Network Virtualization: A Model for combined Virtual Core Network Function Placement and Topology Optimization. *IEEE Communication Networks*.
- Ben-Ameur, W., & Kerivin, H. (2005). Routing of Uncertain Traffic Demands. *Optimization and Engineering*, 6(3), 283-313.
- Bessaou, M., & Siarry, P. (2001). A genetic algorithm with real-value coding to optimize multimodal continuous functions. *Struct Multidisc Optim*, 23(1), 63-74.
- CantoDosSantos, J. V., & Barreto, M. B. (2011). Computacional System Based on Genetic Algorithm for Planning and Restoration of Telecommunication Networks. *International Conference Applied Computing (IADIS), Unico*, pp. 241-248.
- Couëtoux, B., Gourvès, L., Monnot, J., & Telelis, O. A. (11 de Março de 2010). Labeled Traveling Salesman Problems: Complexity and approximation. *Discrete Optimization*, 7(1), pp. 74-85. doi:doi:10.1016/j.disopt.2010.02.003
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Publishing Company, Inc.
- Hillier, F. S., & Lieberman, G. J. (2009). *Introduction to operations research* (9ª ed.). New York: Mc Graw-Hill Education.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems* (1 ed.). MIT Press.
- Liang, Y., & Leung, K.-S. (2011). Genetic Algorithm with adaptive elitist-population strategies for multimodal function optimization. *Applied Soft Computing*, 11, pp. 2017-2034. Fonte: www.elsevier.com/locate/asoc
- Lienig, J., & Thulasiraman, K. (1993). A Genetic Algorithm for Channel Routing in VLSI Circuits. *MIT Press Journals - Evolutionary Computation*, 1(4), pp. 293-311. doi:doi: 10.1162/evco.1993.1.4.293

- Linden, R. (2012). *Algoritmos Genéticos* (3ª ed.). Rio de Janeiro: Ciência Moderna Ltda.
- Miller, B. L., & Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9, pp. 193-212.
- Orlowski, S., Wessälly, R., Pióro, M., & Tomaszewski, A. (10 de Maio de 2010). SNDlib 1.0 - Survivable Network Design Library. *Networks - An International Journal*, 55(3), 276-286.
- Peralta, R. C., Forghani, A., & Fayad, H. (2014). Multiobjective genetic algorithm conjunctive use optimization for production, cost, and energy with dynamic return flow. *Journal of Hydrology*, 511, pp. 776-785. doi:<http://dx.doi.org/10.1016/j.jhydrol.2014.01.044>
- Rodríguez-Martín, I., Salazar-González, J.-J., & Yaman, H. (2016). Hierarchical Survivable Network Design Problems. *Electronic Notes in Discrete Mathematics*, 52, pp. 229-236. doi:<http://dx.doi.org/10.1016/j.endm.2016.03.030>
- Soni, N., & Tapas, K. (2014). Study of Various Mutation Operators in Genetic Algorithms. In: International Journal of Computer Science and Information Technologies. Universidade de Lingaya's. Faridabad: , p. 4514-4521. *IJCSIT - International Journal of Computer Science and Information Technologies*, 5(3), pp. 4519-4521.
- Tripathy, P. K., Dash, R. K., & Tripathy, C. R. (2012). A Genetic Algorithm Based Approach for Topological Optimization of Interconnection Networks. (N. I. Department of Computer Science & Engineering, Ed.) *2nd International Conference on Communication, Computing & Security - ICCCS*, 6, pp. 196-205.
- Zuse Institute Berlin (ZIB). (2005). *SNDlib*. Acesso em 26 de Julho de 2016, disponível em Survivable fixed telecommunication network design: <http://sndlib.zib.de/home.action>

APÊNDICE - B

Artigo apresentado ao XLVIII Simpósio Brasileiro de Pesquisa Operacional (SBPO) em Vitória - ES no período de 27 a 30 de setembro de 2016.

APÊNDICE C – Código-fonte elaborado no software MATLAB para simulação da instância PDH.

```
function
[melhorcusto, linksUtilizados, capTotal, capUsada, quantArcoDemanda, num_de
manda, tempo, G, tam_populacao, num_comp, num_geracoes, prob_crossover, N, pro
b_mutacao, num_trocas]
AG_PDH(cfg, demanda_paths, link_nos, link_custo, demanda_prev);
%% PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS 2016
%MESTRADO EM ENGENHARIA DE PRODUÇÃO E SISTEMAS (MEPROS)
%PROFESSORA ORIENTADORA: Dra MARIA JOSÉ PEREIRA DANTAS
%ALUNO: EMERSON DE SOUSA CAMPOS (BOLSISTA FAPEG) - MEPROS
%ALUNO: VICTOR HONORATO CUNHA (BOLSISTA FAPEG) - ENGENHARIA ELÉTRICA
%ALUNO: LUCAS DE CARVALHO GUIMARÃES (VOLUNTÁRIO) - ENGENHARIA ELÉTRICA

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%ALGORITMO GENÉTICO PARA OTIMIZAÇÃO DA REDE PDH.
%BIBLIOTECA: SNDLib
%INSTÂNCIA: U-U-E-N-C-A-N-N
%PARÂMETROS: x de acordo com trabalho de Bruno Quirino.
%clc;clear;
%% CÓDIGO FONTE
%IMPORTA DADOS E PARÂMETROS DA REDE||||||||||||||||||||||||||||||||
[num_nos, num_links, num_demanda, tam_populacao, cap_link, num_comp, num_ger
acoes, prob_crossover, N, prob_mutacao, num_trocas] = importpdh(cfg);
tic;
%GERA A POPULAÇÃO INICIAL|||||||||||||||||||||||||||||||||||||||||
[modularidade, soma_demandas, atribuicao_mod_link, custo_rede_separado, cu
sto_total_rede, folga]
genpop(tam_populacao, num_demanda, demanda_paths, num_links, demanda_prev,
cap_link, link_custo);

%PROCESSO DE OTIMIZAÇÃO DA SOLUÇÃO|||||||||||||||||||||||||||||
GeracoesSemMelhora = 0;
custo_total_rede_elite=0;
prob_mutacao0 = prob_mutacao;
for ger = 1:num_geracoes
    %ELITISMO
    custo_total_rede_elite1 = custo_total_rede_elite;
    [custo_rede_separado_elite,
custo_total_rede_elite, atribuicao_mod_link_elite, modularidade_elite] =
elitismo(custo_rede_separado,
custo_total_rede, atribuicao_mod_link, modularidade );
    custo_total_rede_elite2 = custo_total_rede_elite;

    if (custo_total_rede_elite1 == custo_total_rede_elite2)
        GeracoesSemMelhora = GeracoesSemMelhora+1;
        if((GeracoesSemMelhora==20) && (prob_mutacao<.6))
            prob_mutacao = prob_mutacao+prob_mutacao0/100*exp(1);
            GeracoesSemMelhora =0;
        end
    end
    monitorarMutacao(ger) = prob_mutacao;
    disp(prob_mutacao);

%    CROSSOVER
```

```

%           [filhos,c_rede_sep,custo_total_rede,atribuicao_mod_link] =
cross(custo_total_rede,tam_populacao,num_comp,num_demanda,num_links,ca
p_link,link_custo,modularidade,prob_crossover);
    [filhos,c_rede_sep,custo_total_rede,atribuicao_mod_link]
=
CrossoverNPontos(custo_total_rede,tam_populacao,num_comp,num_demanda,n
um_links,cap_link,link_custo,modularidade,prob_crossover,N);
    modularidade = filhos; custo_rede_separado = c_rede_sep;

    %MUTAÇÃO

[custo_rede_separado,custo_total_rede,atribuicao_mod_link,modularidade
]
=
mutacao(custo_total_rede,custo_rede_separado,atribuicao_mod_link,tam_p
opulacao,num_demanda,cap_link,link_custo,modularidade,prob_mutacao,
demanda_paths, demanda_prev, num_trocas);
    %SUBSTITUINDO MELHOR VALOR ENCONTRADO NO ELITISMO
    custo_rede_separado(1,:) = custo_rede_separado_elite;
    custo_total_rede(1) = custo_total_rede_elite;
    atribuicao_mod_link(1,:) = atribuicao_mod_link_elite;
    modularidade(:, :, 1) = modularidade_elite;
    disp('Processo Iterativo: ')
    disp(ger)
    %ARMAZENANDO DADOS
    [menorcusto,posmenorcusto]= min(custo_total_rede);
    melhorcusto(ger) = menorcusto;
    melhorcusto_rede_separado(ger,:) =
custo_rede_separado(posmenorcusto,:);
    melhormodularidade(:, :, ger) = modularidade(:, :, posmenorcusto);
    melhoratribuicao_mod_link(ger,:) =
atribuicao_mod_link(posmenorcusto,:);

end
tempo = toc;
disp('Fim do AG_PDH')
%% IMPRIME E EXPORTA RESULTADOS
figure
plot(melhorcusto)
title(num2str(min(melhorcusto)));
grid minor
figure(2)
G
=
DrawGraph(modularidade,atribuicao_mod_link,tam_populacao,num_links,lin
k_nos,num_nos);
G.Edges

%% RELATORIO
fprintf('Custo: %9.2f \n',min(melhorcusto))

linksUtilizados = sum(atribuicao_mod_link_elite>0);
fprintf('Links Utilizados: %3.0f \n',linksUtilizados)

demTotal = sum(sum(modularidade(:, :, 1)));
capTotal = sum(atribuicao_mod_link_elite);
capUsada = demTotal/capTotal*100;
fprintf('Capacidade Total Instalada: %4.2f \n',capTotal)
fprintf('Capacidade Utilizada: %4.2f%% \n',capUsada)

for linha = 1:num_demanda
    quantArcoDemanda(linha) = sum(modularidade(linha, :, 1)>0);
end

```

```
fprintf('Tamanho      medio      de      caminhos:      %4.2f\n', (sum(quantArcoDemanda)/num_demanda))  
  
fprintf('Tempo Gasto: %4.2f segundos \n', tempo)  
  
end
```

```

function
[num_nos,num_links,num_demanda,tam_populacao,cap_link,num_comp,num_ger
acoes,prob_crossover,N,prob_mutacao,num_trocas] = importpdh(cfg)
%
%[num_nos, num_links, num_demanda, link_nos, link_custo, demanda_prev,
demanda_paths] = importpdh()
%
%Essa função importa dados da rede pdh do excel para o MATLAB
%
    %num_nos            numero de nós
    %num_links         numero de links
    %num_deman0da      quantidade de demandas a serem escoadas
    %link_nos          matriz que mostra o link que está entre determinado
par de nós
    %link_custo        matriz com o custo para escoar dados em determinado
link
    %demanda_prev      matriz quantidade de demanda prevista em canais
E1
    %demanda_paths     matriz (5x34x24) contém os caminhos considerados
para escoar a demanda
    %tam_populacao     tamanho da população
%
disp('Importando Dados...')
%% Definições Iniciais
switch cfg
case 1
    num_nos = 11;
    num_links = 34;
    num_demanda = 24;
    tam_populacao = 100;
    num_comp = 2;
    num_geracoes = 100;
    cap_link = [30; 480; 1920];
    prob_crossover = 0.9;
    N = 5; %Numero de cortes no crossover;
    prob_mutacao = .05;
    num_trocas = 2;

case 2
    num_nos = 11;
    num_links = 34;
    num_demanda = 24;
    tam_populacao = 200;
    num_comp = 2;
    num_geracoes = 100;
    cap_link = [30; 480; 1920];
    prob_crossover = 0.9;
    N = 5; %Numero de cortes no crossover;
    prob_mutacao = .05;
    num_trocas = 2;

case 3
    num_nos = 11;
    num_links = 34;
    num_demanda = 24;
    tam_populacao = 300;
    num_comp = 2;
    num_geracoes = 100;
    cap_link = [30; 480; 1920];
    prob_crossover = 0.9;
    N = 5; %Numero de cortes no crossover;

```

```
prob_mutacao = .05;
num_trocas = 2;

case 4
num_nos = 11;
num_links = 34;
num_demanda = 24;
tam_populacao = 400;
num_comp = 2;
num_geracoes = 100;
cap_link = [30; 480; 1920];
prob_crossover = 0.90;
N = 5; %Numero de cortes no crossover;
prob_mutacao = .05;
num_trocas = 2;

case 5
num_nos = 11;
num_links = 34;
num_demanda = 24;
tam_populacao = 500;
num_comp = 2;
num_geracoes = 100;
cap_link = [30; 480; 1920];
prob_crossover = 0.9;
N = 5; %Numero de cortes no crossover;
prob_mutacao = .05;
num_trocas = 2;

case 6
num_nos = 11;
num_links = 34;
num_demanda = 24;
tam_populacao = 100;
num_comp = 2;
num_geracoes = 100;
cap_link = [30; 480; 1920];
prob_crossover = 0.9;
N = 5; %Numero de cortes no crossover;
prob_mutacao = .05;
num_trocas = 2;

case 7
num_nos = 11;
num_links = 34;
num_demanda = 24;
tam_populacao = 200;
num_comp = 2;
num_geracoes = 200;
cap_link = [30; 480; 1920];
prob_crossover = 0.9;
N = 5; %Numero de cortes no crossover;
prob_mutacao = .05;
num_trocas = 2;

case 8
num_nos = 11;
num_links = 34;
num_demanda = 24;
tam_populacao = 300;
```



```
num_comp = 2;
num_geracoes = 300;
cap_link = [30; 480; 1920];
prob_crossover = 0.9;
N = 5; %Numero de cortes no crossover;
prob_mutacao = .05;
num_trocas = 2;

case 9
num_nos = 11;
num_links = 34;
num_demanda = 24;
tam_populacao = 400;
num_comp = 2;
num_geracoes = 400;
cap_link = [30; 480; 1920];
prob_crossover = 0.9;
N = 5; %Numero de cortes no crossover;
prob_mutacao = .05;
num_trocas = 2;

case 10
num_nos = 11;
num_links = 34;
num_demanda = 24;
tam_populacao = 500;
num_comp = 2;
num_geracoes = 500;
cap_link = [30; 480; 1920];
prob_crossover = 0.9;
N = 5; %Numero de cortes no crossover;
prob_mutacao = .05;
num_trocas = 2;

case 11
num_nos = 11;
num_links = 34;
num_demanda = 24;
tam_populacao = 1000;
num_comp = 2;
num_geracoes = 500;
cap_link = [30; 480; 1920];
prob_crossover = 0.95;
N = 10; %Numero de cortes no crossover;
prob_mutacao = .05;
num_trocas = 5;
```

```
end
end
```

```

clear; clc;
%% Importando informação dos links entre cada par de nós
filename = 'pdh_info.xlsx';
sheet = 'PATHS';
xlRange = 'C2:AJ12';
link_nos = xlsread(filename,sheet,xlRange);

%% Custo de cada link
sheet = 'SNDLIB';
xl30Range = 'I2:I35'; xl480Range = 'K2:K35'; xl1920Range = 'M2:M35';
link_custo30 = xlsread(filename,sheet,xl30Range);
link_custo480 = xlsread(filename,sheet,xl480Range);
link_custo1920 = xlsread(filename,sheet,xl1920Range);
link_custo = [link_custo30 link_custo480 link_custo1920];

%% Quantidade de canais El previstos (demandas)
xlRange = 'R2:R25';
demanda_prev = xlsread(filename,sheet,xlRange);

%% Importando os 5 caminhos de cada demanda já com codificação binária
sheet = 'PATH_CODE';
%%
%Demanda 1
xlRange = 'C3:AJ7';
demanda_paths(:, :, 1) = xlsread(filename,sheet,xlRange);

%Demanda 2
xlRange = 'C9:AJ13';
demanda_paths(:, :, 2) = xlsread(filename,sheet,xlRange);

%Demanda 3
xlRange = 'C15:AJ19';
demanda_paths(:, :, 3) = xlsread(filename,sheet,xlRange);

%Demanda 4
xlRange = 'C21:AJ25';
demanda_paths(:, :, 4) = xlsread(filename,sheet,xlRange);

%Demanda 5
xlRange = 'C27:AJ31';
demanda_paths(:, :, 5) = xlsread(filename,sheet,xlRange);

%Demanda 6
xlRange = 'C33:AJ37';
demanda_paths(:, :, 6) = xlsread(filename,sheet,xlRange);

%Demanda 7
xlRange = 'C39:AJ43';
demanda_paths(:, :, 7) = xlsread(filename,sheet,xlRange);

%Demanda 8
xlRange = 'C45:AJ49';
demanda_paths(:, :, 8) = xlsread(filename,sheet,xlRange);

%Demanda 9
xlRange = 'C51:AJ55';
demanda_paths(:, :, 9) = xlsread(filename,sheet,xlRange);

%Demanda 10
xlRange = 'C57:AJ61';
demanda_paths(:, :, 10) = xlsread(filename,sheet,xlRange);

%Demanda 11
xlRange = 'C63:AJ67';
demanda_paths(:, :, 11) = xlsread(filename,sheet,xlRange);

```

```

%Demanda 12
xlRange = 'C69:AJ73';
demanda_paths(:, :, 12) = xlsread(filename, sheet, xlRange);

%Demanda 13
xlRange = 'C75:AJ79';
demanda_paths(:, :, 13) = xlsread(filename, sheet, xlRange);

%Demanda 14
xlRange = 'C81:AJ85';
demanda_paths(:, :, 14) = xlsread(filename, sheet, xlRange);

%Demanda 15
xlRange = 'C87:AJ91';
demanda_paths(:, :, 15) = xlsread(filename, sheet, xlRange);

%Demanda 16
xlRange = 'C93:AJ97';
demanda_paths(:, :, 16) = xlsread(filename, sheet, xlRange);

%Demanda 17
xlRange = 'C99:AJ103';
demanda_paths(:, :, 17) = xlsread(filename, sheet, xlRange);

%Demanda 18
xlRange = 'C105:AJ109';
demanda_paths(:, :, 18) = xlsread(filename, sheet, xlRange);

%Demanda 19
xlRange = 'C111:AJ115';
demanda_paths(:, :, 19) = xlsread(filename, sheet, xlRange);

%Demanda 20
xlRange = 'C117:AJ121';
demanda_paths(:, :, 20) = xlsread(filename, sheet, xlRange);

%Demanda 21
xlRange = 'C123:AJ127';
demanda_paths(:, :, 21) = xlsread(filename, sheet, xlRange);

%Demanda 22
xlRange = 'C129:AJ133';
demanda_paths(:, :, 22) = xlsread(filename, sheet, xlRange);

%Demanda 23
xlRange = 'C135:AJ139';
demanda_paths(:, :, 23) = xlsread(filename, sheet, xlRange);

%Demanda 24
xlRange = 'C141:AJ145';
demanda_paths(:, :, 24) = xlsread(filename, sheet, xlRange);

disp ('Dados importados com sucesso!')

testes = 100;
casos = 11;
for cfig = 9:casos
    mkdir ('EXPERIMENTOS', strcat ('CONFIG', num2str (cfig)))
    for t = 1:testes

[melhorcusto, linksUtilizados, capTotal, capUsada, quantArcoDemanda, num_de
manda, tempo, G, tam_populacao, num_comp, num_geracoes, prob_crossover, N, pro
b_mutacao, num_trocas] =
AG_PDH(cfig, demanda_paths, link_nos, link_custo, demanda_prev);

        %% Relatorio Inicial

```

```

local
strcat('C:\LATINCOM2016\_AG_PDH_2016_07_12\', 'EXPERIMENTOS\', strcat('C
ONFIG', num2str(cfig)), '\', 'Resultados', num2str(cfig), '.xlsx');
    xlswrite(local, {'Custo:'}, t, 'A1');
xlswrite(local, min(melhorcusto), t, 'B1');
    xlswrite(local, {'Links Utilizados:'}, t, 'A2');
xlswrite(local, linksUtilizados, t, 'B2');
    xlswrite(local, {'Capacidade Instalada:'}, t, 'A3');
xlswrite(local, capTotal, t, 'B3');
    xlswrite(local, {'Capacidade Utilizada (%):'}, t, 'A4');
xlswrite(local, capUsada, t, 'B4');
    xlswrite(local, {'Tamanho médio dos caminhos:'}, t, 'A5');
xlswrite(local, (sum(quantArcoDemanda)/num_demanda), t, 'B5');
    xlswrite(local, {'Tempo Gasto'}, t, 'A6');
xlswrite(local, tempo, t, 'B6');

    %% Relatorio da Configuracao
    xlswrite(local, {'População:'}, t, 'A8');
xlswrite(local, tam_populacao, t, 'B8');
    xlswrite(local, {'Competidores:'}, t, 'A9');
xlswrite(local, num_comp, t, 'B9');
    xlswrite(local, {'Geracoes:'}, t, 'A10');
xlswrite(local, num_geracoes, t, 'B10');
    xlswrite(local, {'Crossover:'}, t, 'A11');
xlswrite(local, prob_crossover, t, 'B11');
    xlswrite(local, {'Cortes Crossover:'}, t, 'A12');
xlswrite(local, N, t, 'B12');
    xlswrite(local, {'Mutacao:'}, t, 'A13');
xlswrite(local, prob_mutacao, t, 'B13');
    xlswrite(local, {'Trocas Mutacao:'}, t, 'A14');
xlswrite(local, num_trocas, t, 'B14');

    %% Relatorio da Rede Otimizada
    xlswrite(local, {'Origem'}, t, 'D1');
xlswrite(local, G.Edges.EndNodes(:,1), t, 'D2');
    xlswrite(local, {'Destino'}, t, 'E1');
xlswrite(local, G.Edges.EndNodes(:,2), t, 'E2');
    xlswrite(local, {'Capacidade Instalada'}, t, 'F1');
xlswrite(local, G.Edges.Weight, t, 'F2');
    xlswrite(local, {'Capacidade Utilizada'}, t, 'G1');
xlswrite(local, G.Edges.Code, t, 'G2');
local
strcat('C:\LATINCOM2016\_AG_PDH_2016_07_12\', 'EXPERIMENTOS\', strcat('C
ONFIG', num2str(cfig)), '\', strcat('CUSTO', num2str(t), '.png'));
    saveas (figure(1), local);
local
strcat('C:\LATINCOM2016\_AG_PDH_2016_07_12\', 'EXPERIMENTOS\', strcat('C
ONFIG', num2str(cfig)), '\', strcat('GRAPH', num2str(t), '.png'));
    saveas (figure(2), local);
close all; clc;
end
end

```

```

function
[modularidade,soma_demandas,atribuicao_mod_link,custo_rede_separado,cu
sto_total_rede,folga] =
genpop(tam_populacao,num_demanda,demanda_paths,num_links,demanda_prev,
cap_link,link_custo)
%
%-----
%                               [populacao]                               =                               genpop
(tam_populacao,num_demanda,demanda_paths,num_links)
%
%Esta função cria uma população de indivíduos da rede PDH
%-----
disp('Gerando população e calculando custos...')
%% Escolhendo caminhos
populacao = zeros(num_demanda,num_links,tam_populacao);
for g = 1:tam_populacao
for f = 1:num_demanda
linha = randi(5);
populacao(f,:,g) = demanda_paths(linha,:,f); %Escolhe um dos
5 caminhos disponíveis, faz isso 24x e forma um individuo.
end
end %Faz as atribuições 100x e assim temos uma população de 0 e 1.
%População é uma matriz de 0 e 1 de ordem (24x34x100)
%% Gerando matriz com as modularidades e custo de cada link, e o custo
total da rede
modularidade = zeros(num_demanda,num_links,tam_populacao);
soma_demandas = zeros(tam_populacao,num_links);
atribuicao_mod_link = zeros(tam_populacao,num_links);
custo_rede_separado = zeros(tam_populacao,num_links);
custo_total_rede = zeros(tam_populacao,1);

for g = 1:tam_populacao
for linha = 1:num_demanda
for coluna = 1:num_links
modularidade(linha,coluna,g) =
populacao(linha,coluna,g) * demanda_prev(linha);
end % Matriz modularidade contém os valores de demanda
colocados em cada link escolhido ordem (24x34xg)
end

soma_demandas(g,:) = sum(modularidade(:, :, g)); %Ordem(100x34)
contem a soma de demanda de cada link (carregamento dos links)

%% Correção de Infactibilidade

b1 = soma_demandas(g,:) > cap_link(3); %b1 é um vetor que
armazena o teste lógico
s_b1 = sum(b1);
if s_b1 > 0 %Caso exista alguma soma de demanda maior que
1920
while s_b1 > 0
for s = 1:num_demanda %Esse laço cria uma novo
indivíduo de 0 e 1
linha = randi(5);
populacao(s,:,g) =
demanda_paths(linha,:,s);
end
for linha = 1:num_demanda %Esse laço coloca os
valores de demanda no lugar dos numeros 1
for coluna = 1:num_links

```

```

                                modularidade(linha,coluna,g)           =
populacao(linha,coluna,g) * demanda_prev(linha);
                                end
                                end

                                soma_demandas(g,:) = sum(modularidade(:, :,g));
                                b1 = soma_demandas(g,:) > cap_link(3);
                                s_b1 = sum(b1);
                                end
                                end

%% Alocação de Capacidades de Links

                                %Modularidade = 1920
                                c1 = soma_demandas(g,:) <= cap_link(3); %Teste lógico: quais
links estão carregado com valores menores ou iguais a 1920
                                c2 = soma_demandas(g,:) > cap_link(2); %Teste lógico: quais
links estão carregado com valores maiores a 480
                                c = c1.*c2; %Multiplica elemento
por elemento, para que se satisfaça as duas condições
                                atribuicao_mod_link1 = cap_link(3) * (c); %Aloca o link de
1920 onde se tem valor lógico 1
                                custo_dem_link1 = (c)' .* link_custo(:,3); %Atribui os links

                                %Modularidade = 480
                                d1 = soma_demandas(g,:) <= cap_link(2);
                                d2 = soma_demandas(g,:) > cap_link(1);
                                d = d1.*d2;
                                atribuicao_mod_link2 = cap_link(2) * (d);
                                custo_dem_link2 = (d)' .* link_custo(:,2);

                                %Modularidade = 30
                                e1 = soma_demandas(g,:) <= cap_link(1);
                                e2 = soma_demandas(g,:) > 0;
                                e = e1.*e2;
                                atribuicao_mod_link3 = cap_link(1) * (e);
                                custo_dem_link3 = (e)' .* link_custo(:,1);

                                atribuicao_mod_link(g,:) = atribuicao_mod_link1 +
atribuicao_mod_link2 + atribuicao_mod_link3; % Junta todos os links
alocados
                                custo_rede_separado(g,:) = custo_dem_link1 + custo_dem_link2 +
custo_dem_link3; %Soma os custos de cada link
                                custo_total_rede(g) = sum(custo_rede_separado(g,:));
                                end
                                folga = atribuicao_mod_link-soma_demandas;
                                disp('População e custos gerados!')
                                end
end

```

```

function [custo_rede_separado_elite,
custo_total_rede_elite,atribuicao_mod_link_elite,modularidade_elite] =
elitismo(custo_rede_separado,
custo_total_rede,atribuicao_mod_link,modularidade)
%
%
    [custo_total_rede_elite,posicao_menor_custo] =
min(custo_total_rede);
    custo_rede_separado_elite =
custo_rede_separado(posicao_menor_custo,:);
    atribuicao_mod_link_elite =
atribuicao_mod_link(posicao_menor_custo,:);
    modularidade_elite(:, :, 1) = modularidade(:, :, posicao_menor_custo);

end

```

```

function[pai1,pai2]=selecao(custo_total_rede,tam_populacao,num_comp,mo
dularidade)
%Seleção de pais por torneio
%
%
    a = randi(tam_populacao,num_comp,2);
    valor_cand1 = zeros(num_comp,1);
    valor_cand2 = zeros(num_comp,1);

    for t = 1:num_comp
        valor_cand1(t) = custo_total_rede(a(t,1));
        valor_cand2(t) = custo_total_rede(a(t,2));
    end
    [~,posicao1] = min(valor_cand1);
    [~,posicao2] = min(valor_cand2);

    pai1 = modularidade(:, :, posicao1);
    pai2 = modularidade(:, :, posicao2);

end

```

```

function [filhos,c_rede_sep,custo_total_rede,atribuicao_mod_link] =
cross(custo_total_rede,tam_populacao,num_comp,num_demanda,num_links,ca
p_link,link_custo,modularidade,prob_crossover)
%Crossover de um ponto

    filhos = zeros(num_demanda,num_links,tam_populacao);
    atribuicao_mod_link = zeros(tam_populacao,num_links);
    c_rede_sep = zeros(tam_populacao,num_links);

    for n = 1:tam_populacao
        if rand < prob_crossover
            [pai1,                pai2] =
selecao(custo_total_rede,tam_populacao,num_comp,modularidade);%São pais
com as modularidades
            corte = randi(num_demanda);

            while ((corte == 1)|| (corte == 24))
                corte = randi(num_demanda);
            end

            cortel = zeros(corte,num_links);
            corte2 = zeros(corte,num_links);
            for line = 1:corte
                cortel(line,:) = pai1(line,:);
                corte2(line,:) = pai2(line,:);
            end

            cortel1 = zeros(num_demanda-corte,num_links);
            corte22 = zeros(num_demanda-corte,num_links);
            for c = corte+1:num_demanda
                cortel1(c-corte,:) = pai1(c,:);
                corte22(c-corte,:) = pai2(c,:);
            end
            filho1 = [cortel;corte22];%São filhos com as modularidades
            filho2 = [corte2;cortel1];

            soma_demandas1 = sum(filho1);
            soma_demandas2 = sum(filho2);
            [filho_corrigido,soma_demandas_corrigidas] =
corrigir_infact_cross(soma_demandas1,soma_demandas2,filho1,filho2,cap_
link,custo_total_rede,tam_populacao,num_comp,modularidade);
            filhos(:, :, n) = filho_corrigido;

            %Modularidade = 1920
            c1 = soma_demandas_corrigidas <= cap_link(3);
            c2 = soma_demandas_corrigidas > cap_link(2);
            c = c1.*c2;
            atribuicao_mod_link1 = cap_link(3) * (c);
            custo_dem_link1 = (c)' .* link_custo(:,3);

            %Modularidade = 480
            d1 = soma_demandas_corrigidas <= cap_link(2);
            d2 = soma_demandas_corrigidas > cap_link(1);
            d = d1.*d2;
            atribuicao_mod_link2 = cap_link(2) * (d);
            custo_dem_link2 = (d)' .* link_custo(:,2);

            %Modularidade = 30
            e1 = soma_demandas_corrigidas <= cap_link(1);
            e2 = soma_demandas_corrigidas > 0;

```



```

    e = e1.*e2;
    atribuicao_mod_link3 = cap_link(1) * (e);
    custo_dem_link3 = (e)' .* link_custo(:,1);

    atribuicao_mod_link(n,:) = atribuicao_mod_link1 +
    atribuicao_mod_link2 + atribuicao_mod_link3;
    custo_rede_separado = custo_dem_link1 + custo_dem_link2 +
    custo_dem_link3;
    c_rede_sep(n,:) = custo_rede_separado;
    custo_total_rede(n) = sum(custo_rede_separado);
    clear cortel
    clear corte2
    clear corte11
    clear corte22
else
    filhos(:, :, n) = modularidade(:, :, n);
    soma_demandas_corrigidas = sum(modularidade(:, :, n));

    %Modularidade = 1920
    c1 = soma_demandas_corrigidas <= cap_link(3);
    c2 = soma_demandas_corrigidas > cap_link(2);
    c = c1.*c2;
    atribuicao_mod_link1 = cap_link(3) * (c);
    custo_dem_link1 = (c)' .* link_custo(:,3);

    %Modularidade = 480
    d1 = soma_demandas_corrigidas <= cap_link(2);
    d2 = soma_demandas_corrigidas > cap_link(1);
    d = d1.*d2;
    atribuicao_mod_link2 = cap_link(2) * (d);
    custo_dem_link2 = (d)' .* link_custo(:,2);

    %Modularidade = 30
    e1 = soma_demandas_corrigidas <= cap_link(1);
    e2 = soma_demandas_corrigidas > 0;
    e = e1.*e2;
    atribuicao_mod_link3 = cap_link(1) * (e);
    custo_dem_link3 = (e)' .* link_custo(:,1);

    atribuicao_mod_link(n,:) = atribuicao_mod_link1 +
    atribuicao_mod_link2 + atribuicao_mod_link3;
    custo_rede_separado = custo_dem_link1 + custo_dem_link2 +
    custo_dem_link3;
    c_rede_sep(n,:) = custo_rede_separado;
    custo_total_rede(n) = sum(custo_rede_separado);
end
end
end
end

```

```

function [filhos,c_rede_sep,custo_total_rede,atribuicao_mod_link] =
CrossoverNPontos...

(custo_total_rede,tam_populacao,num_comp,num_demanda,num_links,cap_lin
k,link_custo,modularidade,prob_crossover,N)
%Crossover de um ponto

    filhos = zeros(num_demanda,num_links,tam_populacao);
    atribuicao_mod_link = zeros(tam_populacao,num_links);
    c_rede_sep = zeros(tam_populacao,num_links);

    for n = 1:tam_populacao
        if rand < prob_crossover
            if N == 1
                [pai1,                pai2]                =
selecao(custo_total_rede,tam_populacao,num_comp,modularidade);%São pais
com as modularidades
                corte = randi(num_demanda);
                while ((corte == 1) || (corte == 24))
                    corte = randi(num_demanda);
                end

                corte1 = zeros(corte,num_links);
                corte2 = zeros(corte,num_links);

                for line = 1:corte
                    corte1(line,:) = pai1(line,:);
                    corte2(line,:) = pai2(line,:);
                end

                corte11 = zeros(num_demanda-corte,num_links);
                corte22 = zeros(num_demanda-corte,num_links);
                for c = corte+1:num_demanda
                    corte11(c-corte,:) = pai1(c,:);
                    corte22(c-corte,:) = pai2(c,:);
                end
                filho1 = [corte1;corte22];%São filhos com as
modularidades
                filho2 = [corte2;corte11];

                soma_demandas1 = sum(filho1);
                soma_demandas2 = sum(filho2);

                [filho_corrigido,soma_demandas_corrigidas] =
corrigir_infact_cross(soma_demandas1,soma_demandas2,filho1,filho2,cap_
link,custo_total_rede,tam_populacao,num_comp,modularidade);
                filhos(:, :, n) = filho_corrigido;

                %Modularidade = 1920
                c1 = soma_demandas_corrigidas <= cap_link(3);
                c2 = soma_demandas_corrigidas > cap_link(2);
                c = c1.*c2;
                atribuicao_mod_link1 = cap_link(3) * (c);
                custo_dem_link1 = (c)' .* link_custo(:,3);

                %Modularidade = 480
                d1 = soma_demandas_corrigidas <= cap_link(2);
                d2 = soma_demandas_corrigidas > cap_link(1);

```

```

d = d1.*d2;
atribuicao_mod_link2 = cap_link(2) * (d);
custo_dem_link2 = (d)' .* link_custo(:,2);

%Modularidade = 30
e1 = soma_demandas_corrigidas <= cap_link(1);
e2 = soma_demandas_corrigidas > 0;
e = e1.*e2;
atribuicao_mod_link3 = cap_link(1) * (e);
custo_dem_link3 = (e)' .* link_custo(:,1);

    atribuicao_mod_link(n,:) = atribuicao_mod_link1 +
atribuicao_mod_link2 + atribuicao_mod_link3;
    custo_rede_separado = custo_dem_link1 +
custo_dem_link2 + custo_dem_link3;
    c_rede_sep(n,:) = custo_rede_separado;
    custo_total_rede(n) = sum(custo_rede_separado);
    clear corte1
    clear corte2
    clear corte11
    clear corte22

else
    [pai1, pai2] =
selecao(custo_total_rede,tam_populacao,num_comp,modularidade);%São pais
com as modularidades
    a = randperm(num_demanda-1);
    cortes = sort(a(1:N));

    %PartesPai1 = zeros(max(cortes),num_links,N+1);
    %PartesPai2 = zeros(max(cortes),num_links,N+1);
    filho1 = zeros(num_demanda,num_links);
    filho2 = zeros(num_demanda,num_links);
    for PosCorte = 1:N
        if PosCorte == 1
            %PartesPai1(1:cortes(PosCorte),:,PosCorte) =
pail(1:cortes(PosCorte),:);
            filho1(1:cortes(PosCorte),:) =
pail(1:cortes(PosCorte),:);
            filho2(1:cortes(PosCorte),:) =
pai2(1:cortes(PosCorte),:);
            %PartesPai2(1:cortes(PosCorte),:,PosCorte) =
pai2(1:cortes(PosCorte),:);
        else
            %PartesPai1((cortes(PosCorte-
1)+1):cortes(PosCorte),:,PosCorte) = pail((cortes(PosCorte-
1)+1):cortes(PosCorte),:);
            %PartesPai2((cortes(PosCorte-
1)+1):cortes(PosCorte),:,PosCorte) = pai2((cortes(PosCorte-
1)+1):cortes(PosCorte),:);
            if mod(PosCorte,2)==0
                filho1((cortes(PosCorte-
1)+1):cortes(PosCorte),:) = pai2((cortes(PosCorte-
1)+1):cortes(PosCorte),:);
                filho2((cortes(PosCorte-
1)+1):cortes(PosCorte),:) = pail((cortes(PosCorte-
1)+1):cortes(PosCorte),:);
                t1 = 1;
            else

```

```

                                filho1((cortes(PosCorte-
1)+1):cortes(PosCorte),:) = pai1((cortes(PosCorte-
1)+1):cortes(PosCorte),:);
                                filho2((cortes(PosCorte-
1)+1):cortes(PosCorte),:) = pai2((cortes(PosCorte-
1)+1):cortes(PosCorte),:);
                                t1 = 0;
                                end
                                end
                                end

%PartesPai1((cortes(PosCorte)+1):num_demanda,:,PosCorte+1) =
pai1((cortes(PosCorte)+1):num_demanda,:);

%PartesPai2((cortes(PosCorte)+1):num_demanda,:,PosCorte+1) =
pai2((cortes(PosCorte)+1):num_demanda,:);
    if t1==1
        filho1((cortes(PosCorte)+1):num_demanda,:) =
pai1((cortes(PosCorte)+1):num_demanda,:);
        filho2((cortes(PosCorte)+1):num_demanda,:) =
pai2((cortes(PosCorte)+1):num_demanda,:);
    else
        filho1((cortes(PosCorte)+1):num_demanda,:) =
pai2((cortes(PosCorte)+1):num_demanda,:);
        filho2((cortes(PosCorte)+1):num_demanda,:) =
pai1((cortes(PosCorte)+1):num_demanda,:);
    end
    end
    soma_demandas1 = sum(filho1);
    soma_demandas2 = sum(filho2);

    [filho_corrigido,soma_demandas_corrigidas] =
corrigir_infact_cross(soma_demandas1,soma_demandas2,filho1,filho2,cap_
link,custo_total_rede,tam_populacao,num_comp,modularidade);
    filhos(:, :, n) = filho_corrigido;

%Modularidade = 1920
c1 = soma_demandas_corrigidas <= cap_link(3);
c2 = soma_demandas_corrigidas > cap_link(2);
c = c1.*c2;
atribuicao_mod_link1 = cap_link(3) * (c);
custo_dem_link1 = (c)' .* link_custo(:,3);

%Modularidade = 480
d1 = soma_demandas_corrigidas <= cap_link(2);
d2 = soma_demandas_corrigidas > cap_link(1);
d = d1.*d2;
atribuicao_mod_link2 = cap_link(2) * (d);
custo_dem_link2 = (d)' .* link_custo(:,2);

%Modularidade = 30
e1 = soma_demandas_corrigidas <= cap_link(1);
e2 = soma_demandas_corrigidas > 0;
e = e1.*e2;
atribuicao_mod_link3 = cap_link(1) * (e);
custo_dem_link3 = (e)' .* link_custo(:,1);

    atribuicao_mod_link(n,:) = atribuicao_mod_link1 +
atribuicao_mod_link2 + atribuicao_mod_link3;

```

```

        custo_rede_separado = custo_dem_link1 + custo_dem_link2 +
custo_dem_link3;
        c_rede_sep(n,:) = custo_rede_separado;
        custo_total_rede(n) = sum(custo_rede_separado);

    else
        filhos(:, :, n) = modularidade(:, :, n);
        soma_demandas_corrigidas = sum(modularidade(:, :, n));

        %Modularidade = 1920
        c1 = soma_demandas_corrigidas <= cap_link(3);
        c2 = soma_demandas_corrigidas > cap_link(2);
        c = c1.*c2;
        atribuicao_mod_link1 = cap_link(3) * (c);
        custo_dem_link1 = (c)' .* link_custo(:,3);

        %Modularidade = 480
        d1 = soma_demandas_corrigidas <= cap_link(2);
        d2 = soma_demandas_corrigidas > cap_link(1);
        d = d1.*d2;
        atribuicao_mod_link2 = cap_link(2) * (d);
        custo_dem_link2 = (d)' .* link_custo(:,2);

        %Modularidade = 30
        e1 = soma_demandas_corrigidas <= cap_link(1);
        e2 = soma_demandas_corrigidas > 0;
        e = e1.*e2;
        atribuicao_mod_link3 = cap_link(1) * (e);
        custo_dem_link3 = (e)' .* link_custo(:,1);

        atribuicao_mod_link(n,:) = atribuicao_mod_link1 +
atribuicao_mod_link2 + atribuicao_mod_link3;
        custo_rede_separado = custo_dem_link1 + custo_dem_link2 +
custo_dem_link3;
        c_rede_sep(n,:) = custo_rede_separado;
        custo_total_rede(n) = sum(custo_rede_separado);
    end
end
end

```

```

function
[custo_rede_separado,custo_total_rede,atribuicao_mod_link,modularidade
]
mutacao(custo_total_rede,custo_rede_separado,atribuicao_mod_link,tam_p
opulacao,num_demanda,cap_link,link_custo,modularidade,prob_mutacao,
demanda_paths, demanda_prev, num_trocas)
%Mutação de bit, escolhe uma (ou mais) demanha(s) (linha da matriz
modularidade) e troca o caminho escolhido.
%
for n = 1:tam_populacao
    if rand < prob_mutacao
        erro = 1;
        while erro > 0

            trocar_caminho = randi(num_demanda,num_trocas,1);
            %Escolhe aleatoriamente quais linhas sofrerão mutação
            novo_caminho = randi(5,num_trocas,1); %Escolhe as
            posições dos caminhos que substituirão os antigos

            for cont1 = 1:num_trocas
                modularidade(trocar_caminho(cont1),:,n) =
                demanda_prev(trocar_caminho(cont1))*demanda_paths(novo_caminho(cont1),
                :,trocar_caminho(cont1));
            end

            erro = (sum(sum(modularidade(:, :, n)) > 1920));
        end

        %Modularidade = 1920
        c1 = sum(modularidade(:, :, n)) <= cap_link(3); %Teste lógico:
        quais links estão carregados com valores menores ou iguais a 1920
        c2 = sum(modularidade(:, :, n)) > cap_link(2); %Teste lógico:
        quais links estão carregados com valores maiores a 480
        c = c1.*c2; %Multiplica elemento
        por elemento, para que se satisfaça as duas condições
        atribuicao_mod_link1 = cap_link(3) * (c); %Aloca o link de
        1920 onde se tem valor lógico 1
        custo_dem_link1 = (c)' .* link_custo(:,3); %Atribui os links

        %Modularidade = 480
        d1 = sum(modularidade(:, :, n)) <= cap_link(2);
        d2 = sum(modularidade(:, :, n)) > cap_link(1);
        d = d1.*d2;
        atribuicao_mod_link2 = cap_link(2) * (d);
        custo_dem_link2 = (d)' .* link_custo(:,2);

        %Modularidade = 30
        e1 = sum(modularidade(:, :, n)) <= cap_link(1);
        e2 = sum(modularidade(:, :, n)) > 0;
        e = e1.*e2;
        atribuicao_mod_link3 = cap_link(1) * (e);
        custo_dem_link3 = (e)' .* link_custo(:,1);

        atribuicao_mod_link(n,:) = atribuicao_mod_link1 +
        atribuicao_mod_link2 + atribuicao_mod_link3; % Junta todos os links
        alocados
        custo_rede_separado(n,:) = custo_dem_link1 + custo_dem_link2 +
        custo_dem_link3; %Soma os custos de cada link
        custo_total_rede(n) = sum(custo_rede_separado(n, :));
    end
end
end
end

```

```

local = strcat('C:\LATINCOM2016\_AG_PDH_2016_07_12\', 'EXPERIMENTOS\');
testes = 100;
casos = 11;
for cfig = 1:casos
    disp(cfig)

    for t =1:testes
        disp(t)

            resultados(:,1,t)
xlsread(strcat(local, 'CONFIG', num2str(cfig), '\Resultados', num2str(cfig)
), '.xlsx'), t, 'B1:B6');

    end

    custo = resultados(1,1,:);
    linksUtilizados = resultados(2,1,:);
    capacidadeInstalada = resultados(3,1,:);
    capacidadeUtilizada = resultados(4,1,:);
    tamanhoCaminho = resultados(5,1,:);
    tempoGasto = resultados(6,1,:);

    bestMIN(1) = min(custo); [~,P] = min(custo);
    bestMIN(2) = linksUtilizados(P);
    bestMIN(3) = capacidadeInstalada(P);
    bestMIN(4) = capacidadeUtilizada(P);
    bestMIN(5) = tamanhoCaminho(P);
    bestMIN(6) = tempoGasto(P);

    xlswrite(strcat(local, 'Avaliacao', '.xlsx'), {strcat('Configuração', num2
str(cfig))}, 1, 'A1');
    xlswrite(strcat(local, 'Avaliacao', '.xlsx'), {'Custo'}, 1, 'A2');
    xlswrite(strcat(local, 'Avaliacao', '.xlsx'), {'Links
Utilizados'}, 1, 'A3');
    xlswrite(strcat(local, 'Avaliacao', '.xlsx'), {'Capacidade
Instalada'}, 1, 'A4');
    xlswrite(strcat(local, 'Avaliacao', '.xlsx'), {'Capacidade
Utilizada'}, 1, 'A5');
    xlswrite(strcat(local, 'Avaliacao', '.xlsx'), {'Tamanho
Caminho'}, 1, 'A6');
    xlswrite(strcat(local, 'Avaliacao', '.xlsx'), {'Tempo'}, 1, 'A7');
    xlswrite(strcat(local, 'Avaliacao', '.xlsx'), bestMIN, cfig, 'B2');

end

```

```

function [filho_corrigido,soma_demandas_corrigidas] =
corrigir_infact_cross(soma_demandas1,soma_demandas2,filho1,filho2,cap_
link,custo_total_rede,tam_populacao,num_comp,modularidade)
%
%
s_b1 = soma_demandas1 > cap_link(3);
s_b2 = soma_demandas2 > cap_link(3);

if (sum(s_b1) > 0 && sum(s_b2) > 0)
    factibilidade = 0;
    while factibilidade == 0

        [pai1,                pai2]
selecao(custo_total_rede,tam_populacao,num_comp,modularidade);

        corte = randi(num_demanda);
        while ((corte == 1)|| (corte == 24))
            corte = randi(num_demanda);
        end
        cortel = zeros(corte,num_links);
        corte2 = zeros(corte,num_links);
        for c = 1:corte
            cortel(c,:) = pai1(c,:);
            corte2(c,:) = pai2(c,:);
        end
        cortel1 = zeros(num_demanda-corte,num_links);
        corte22 = zeros(num_demanda-corte,num_links);
        for c = corte+1:num_demanda
            cortel1(c-corte,:) = pai1(c,:);
            corte22(c-corte,:) = pai2(c,:);
        end
        filho_corrigido1 = [cortel;corte22];
        filho_corrigido2 = [corte2;cortel1];

        soma_demandas_corrigidas1 = sum(filho_corrigido1);
        soma_demandas_corrigidas2 = sum(filho_corrigido2);
        s_b1 = soma_demandas_corrigidas1 > cap_link(3);
        s_b2 = soma_demandas_corrigidas2 > cap_link(3);

        if (sum(s_b1) > 0 && sum(s_b2) > 0)
            factibilidade = 0;
        elseif (sum(s_b1)==0 && sum(s_b2)==0)
            factibilidade = 1;
            filho_corrigido = filho_corrigido1;
            soma_demandas_corrigidas = soma_demandas_corrigidas1;
        elseif sum(s_b1)==0
            factibilidade = 1;
            filho_corrigido = filho_corrigido1;
            soma_demandas_corrigidas = soma_demandas_corrigidas1;
        elseif sum(s_b2)==0
            factibilidade = 1;
            filho_corrigido = filho_corrigido2;
            soma_demandas_corrigidas = soma_demandas_corrigidas2;
        end
    end

elseif (sum(s_b1)==0 && sum(s_b2)==0)
    filho_corrigido = filho1;
    soma_demandas_corrigidas = soma_demandas1;
elseif sum(s_b1)==0
    filho_corrigido = filho1;

```



```
        soma_demandas_corrigidas = soma_demandas1;
elseif sum(s_b2)==0
    filho_corrigido = filho2;
    soma_demandas_corrigidas = soma_demandas2;
end
end
```

APÊNDICE D – Código-fonte do algoritmo de pós-otimização elaborado no software MATLAB para simulação da instância ta1 (SDH).

```
function [ind_modularidade,ind_link,custo_sem_link,ind_custo_separado]
=
folgas(ind_modularidade_org,ind_link_org,custo_sem_link,link_custo,cap
_link)
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
%
%% Identificação do link a ser retirado e demandas a serem
remanejadas
soma_demandas = sum(ind_modularidade_org); %Soma das demandas que
estão sendo escoadas em cada link
folgas = ind_link_org-soma_demandas; %Calcula as folgas da rede em
cada link
[~,posicoes_folgas] = sort(folgas,'descend'); % ~ Está ocultando o
valor da maior folga que representa o primeiro termo da função max
%[~,posicoes_folgas] = sort(custo_sem_link,'descend'); %Funciona para
retirar o link de maior custo

pos_sort = 1;
k = 5;
cond = 0;
while (cond ==0 && pos_sort<=length(posicoes_folgas))
    disp('pos_sort');
    disp(pos_sort);

[kSpaths_demandas,quantidade_demanda,remanejar_demandas,ind_modularida
de,ind_link,valor_demandas,pos_sort] = selecionarLink...

(pos_sort,posicoes_folgas,ind_modularidade_org,ind_link_org,k);
[~,edges,~] = xlsread('pdh_info.xlsx','links','B1:C34'); num_edges
= length(edges); %Importa todos os links do excel (fonte,destino)
if quantidade_demanda == 0
    break
end

%Encontrar quais links foram escolhidos para remanejar as demandas
[~,edges,~] = xlsread('pdh_info.xlsx','links','B1:C34'); num_edges
= length(edges); %Importa todos os links do excel (fonte,destino)
for i = 1:num_edges %Transforma os dados dos links do excel
(string para double)
    numerico_edges(i,1) = str2double(edges{i,1}(1,2:end));
    numerico_edges(i,2) = str2double(edges{i,2}(1,2:end));
end

for d = 1:quantidade_demanda
    for c = 1:length(kSpaths_demandas{d,1}) %Quantidade de
caminhos gerados para cada demanda d
        pos = 0;
        for nos = 1:length(kSpaths_demandas{d}{c})-1 %Quantidade
de link utilizado em cada caminho c da demanda d
            for e = 1:num_edges %Testa qual link equivale aos nós
selecionados no caminho correspondente
                if or(numerico_edges(e,:) ==
kSpaths_demandas{d}{c}(nos:nos+1),fliplr(numerico_edges(e,:)) ==
kSpaths_demandas{d}{c}(nos:nos+1));
```

```

        pos = pos+1;
        kSpaths_demandas_links{d}{c}(nos) = e;
        break
    end
end
end
end
end

% CODIFICAR CAMINHOS EM MATRIZ DE INCIDÊNCIA
demanda_paths = zeros(k,num_edges,quantidade_demanda);
for d = 1:quantidade_demanda
    for c = 1:length(kSpaths_demandas{d,1}) %Quantidade de
        caminhos gerados para a demanda d a ser remanejada
            for s = 1:length(kSpaths_demandas_links{d}{c})
                demanda_paths(c,kSpaths_demandas_links{d}{c}(s),d) =
1;
            end
        end
    end
    ind_modularidade_inter = ind_modularidade; % _inter - caso a
mudança seja infactível recupera-se o individuo inicial
    %% Remanejar as demandas
    for t = 1:quantidade_demanda %Primeiro retira-se o caminho antigo
        das demandas a serem remanejadas
            ind_modularidade(remanejar_demandas(t),:) =
zeros(1,num_edges);
        end
        nao_altera = 0;
        for t = 1:quantidade_demanda
            for c = 1:length(kSpaths_demandas{d,1})
                novo_caminho = demanda_paths(c,:,t)*valor_demandas(t);
                ind_modularidade(remanejar_demandas(t),:) = novo_caminho;
                if sum(sum(ind_modularidade) <= ind_link) == num_edges
                    break %Caso a folga seja suficiente, o link é alocado
e não se testa outro caminho
                elseif c==length(kSpaths_demandas{d,1})
                    disp('Não é possível alterar o caminho para o link:');
disp(posicoes_folgas(pos_sort));
                    nao_altera = 1;
                    pos_sort = pos_sort+1;
                    if pos_sort>length(posicoes_folgas)
                        disp('É impossível retirar mais links, folgas
extrapoladas');
                    end
                    ind_modularidade = ind_modularidade_inter;
                end
            end
        end
    end
    if nao_altera == 1
        cond = 0;
        break
    else
        cond = 1;
    end
end
end
end

for c1 = 1:num_edges

```

```
for c2 = 1:length(cap_link)
    if ind_link(c1) == cap_link(c2)
        ind_custo_separado(c1) = link_custo(c1,c2);
        break
    elseif c2 == length(cap_link)
        ind_custo_separado(c1) = 0;
    end
end
end
custo_sem_link = sum(ind_custo_separado);
end
```

```

function
[kSpaths_demandas,quantidade_demanda,remanejar_demandas,ind_modularida
de,ind_link,valor_demandas,pos_sort] = selecionarLink...
    (pos_sort,posicoes_folgas,ind_modularidade,ind_link_org,k)
%UNTITLED2 Summary of this function goes here
% Detailed explanation goes here

impossivel = 0;
while (impossivel == 0 && pos_sort<=length(posicoes_folgas))
    disp('pos_sort');
    disp(pos_sort);
    pos_maior_folga = posicoes_folgas(pos_sort); %Armazena a
identificação do link a ser retirado

    remanejar_demandas =
find(ind_modularidade(:,pos_maior_folga)); % Encontra quais demandas
passam no link a ser retirado
    quantidade_demanda = length(remanejar_demandas); %Quantidade
de demandas a serem remanejadas
    if quantidade_demanda == 0
        kSpaths_demandas = 0;
        remanejar_demandas = 0;
        valor_demandas = 0;
        ind_link = ind_link_org;
        ind_link(1,pos_maior_folga) = 0;
        break
    end

    [~,st_demandas,~] = xlsread('info.xlsx','AG2:AH397');
num_demandas = length(st_demandas); %Importa todas as demandas do
excel (fonte,destino)
    numerico_st_demandas = zeros(num_demandas,2); %Transforma os
dados do excel string para double
    for i = 1:num_demandas
        numerico_st_demandas(i,1) =
str2double(st_demandas{i,1}(1,2:end));
        numerico_st_demandas(i,2) =
str2double(st_demandas{i,2}(1,2:end));
    end

    %Busca os valores e nos de saída e chegada das demandas a
serem remanejadas
    valor_demandas = zeros(1,quantidade_demanda); %Irá armazenar
os valores das demandas a serem remanejadas
    io_demandas = zeros(quantidade_demanda,2); % Nós
(fonte,destino) das demandas a serem remanejadas (input,output)
    for t = 1:quantidade_demanda
        valor_demandas(t) =
max(ind_modularidade(remanejar_demandas(t),:));
        io_demandas(t,:) =
numerico_st_demandas(remanejar_demandas(t),:); %Entrada e saída das
demandas a serem remanejadas i=in o=out
    end
    ind_link = ind_link_org;
    ind_link(1,pos_maior_folga) = 0; % Retira o link de maior
folga

    %% ENCONTRAR K CAMINHOS MINIMOS PARA CADA DEMANDA A SER
RETIRADA
    load('adjacencia.mat');

```

```

load('link_nos.mat');
pos_sem_link = find(~ind_link); % Posições que não possuem
link em relação a matriz de adjacencia inicial
for t = 1:length(pos_sem_link)
    io_link = find(link_nos(:,pos_sem_link(t))); %Entrada e
saída do link retirado i=in o=out
    adjacencia(io_link(1),io_link(2)) = Inf;
adjacencia(io_link(2),io_link(1)) = Inf; %Retira o link na matriz de
adjacencia original
    link_nos(io_link(1),pos_sem_link(t)) = 0;
link_nos(io_link(2),pos_sem_link(t)) = 0; %Retira o link na matriz de
incidencia link_nos
end
netCostMatrix = adjacencia;
%% Algoritmo de k caminhos mínimos para cada demandas a ser
remanejada
for t = 1:quantidade_demanda
    source = io_demandas(t,1);
    destination = io_demandas(t,2);

    %-----Show case selected:-----:
    fprintf('Voce selecionou o caso #%d, uma rede %dx%d
:\n',t,size(netCostMatrix,1),size(netCostMatrix,1));

    %    fprintf('Demanda #%d: Caminhos da FONTE %d para o
DESTINO %d, com K = %d \n',t,source,destination, k);

    %-----Call kShortestPath-----:
    [shortestPaths, totalCosts] = kShortestPath(netCostMatrix,
source, destination, k);

    %-----Display results-----:
    fprintf('\nResult of Function call:
kShortestPath(netCostMatrix, source, destination, k) = \n\n');

    if not isempty(shortestPaths)
        impossivel = 1;
        kSpaths_demandas{t,:} = shortestPaths;
    else
        fprintf('No path available between these nodes\n\n');
        pos_sort = pos_sort+1;
        impossivel = 0;
        if pos_sort>length(posicoes_folgas)
            disp('É impossivel retirar mais links, sem
caminhos disponiveis');
        end
        break
    end
end
end
end
end
end

```

APÊNDICE E

Tabela 27 – Links utilizados para escoamento das demandas para a rede tal (SDH)

Demandas	Links						
D1	29	30					
D2	19	21	33	47	48		
D3	1	2	17	18	46	47	
D4	1	5	16	19	46	47	50
D5	11	12	15	21	46	47	
D6	1	18	19	21	45	46	47
D7	8	28					
D8	8	19	21	28			
D9	19	21	29	47	48		
D10	22						
D11	2	18	45	47	48		
D12	1	17	18	45	46	47	
D13	15						
D14	9	29	31	33	46	48	
D15	9	11	12	15	16	18	
D16	21						
D17	9	11	12	15	16	22	
D18	22						
D19	21						
D20	1	16	19	21	33	46	47
D21	2	31	33	45	46		
D22	12	15	16	17	45	46	
D23	12	15	16				
D24	18	19	21				
D25	1	16	31	48			
D26	1	16	31				
D27	9	11					
D28	9	11					
D29	17	22	45	46	47	48	
D30	46	47	48				
D31	16	31					
D32	16	19	21	31	48		
D33	11	12	15	16	17	22	31
D34	11	12	15	16	18	31	
D35	16						

D36	1	5	11	15	50				
D37	12								
D38	5	11	12	50					
D39	1	15							
D40	17	22	45	46	47	48			
D41	22								
D42	5	21	29	30	33	50			
D43	5	19	33	46	47	50			
D44	2	5	8	16	18	31	45	47	50
D45	1	5	18	29	46	47	48	50	
D46	5	9	17	22	45	46	47	50	
D47	2	17	22						
D48	1	2	17	22					
D49	19	21	28	29	47	48			
D50	8	21	27	28	30	31			
D51	9	19	21	30	47	48			
D52	17	19	21	27					
D53	18								
D54	1	2	17	22					
D55	18								
D56	17	22	29	30	31	45	46	47	
D57	29	30							
D58	19	21	31	47	48				
D59	19	21	22	29	47	48			
D60	2	12	15	17					
D61	18	47	48						
D62	18	47	48						
D63	1	9	11	12	15	18			
D64	9	11	12	21	22	27	31		
D65	9	11	12	18	29	47			
D66	30								
D67	1	2	17	22					
D68	11	16	18	19	21	31			
D69	8	21	28	29	30	31			
D70	12	15	18						
D71	8	28	29	30	31				
D72	18	19							
D73	18	19	29	48					
D74	1	16	27	31					

D75	15	16	31							
D76	15	18	19	21	31					
D77	1	16	21	27	31	48				
D78	1	33								
D79	1									
D80	1	8	19	33	47	48				
D81	1	18	47	48						
D82	19	21	31	46	47					
D83	19	21	31	46	47					
D84	1	18	47	48						
D85	8	16	28	29	31	47	48			
D86	8	16	28	31	48					
D87	1	8	16	28	31					
D88	16	21	27	31						
D89	1	8	16	28	31					
D90	1									
D91	16	27	31							
D92	1	5	12	15	50					
D93	11									
D94	8	27	28							
D95	8	9	17	27	28	45				
D96	8	11	12	15	16	28	31			
D97	8	11	12	15	16	28	31			
D98	9	11	12	15	17	45	46			
D99	1	5	9	16	18	31	50			
D100	8	19	21	31	47	48				
D101	1	5	8	16	28	29	30	31	48	50
D102	15	19	21	31	47	48				
D103	1	16	21							
D104	1	18	19							
D105	2	17	22							
D106	15	16								
D107	15	16								
D108	18	47	48							
D109	18	47	48							
D110	11	12	15	17	18	19	22	45	46	47
D111	1	5	8	11	18	19	28	31	50	
D112	25	39								
D113	11	12	15	16	19	47				

D114	1	11	12	15	16	27	31	33	
D115	11	12	15	18	47	48			
D116	9	11	12	15	16	17	22	31	
D117	1	5	9	16	27	31	50		
D118	9	11	12	15	16	17	22	31	45
D119	1	5	16	18	27	31	47	48	50
D120	8	27	28	29	30	31			
D121	16	18	27	31	47	48			
D122	8	19	21	28	31	46	47		
D123	8	19	21	28	31	46	47		
D124	19	21	27	31					
D125	17	22	45						
D126	1	2	17	22					
D127	15	18	46	47					
D128	17	22	45						
D129	15	18	46	47					
D130	1	2	17	22					
D131	8	18	21	27	31	47	48		
D132	16	18	19	27	31				
D133	1	16	31	33					
D134	1	16	31	33					
D135	25	28	39						
D136	1	2	16	17	22	31			
D137	1	2	16	17	22	31			
D138	8	15	16	27	28	29	31	48	
D139	17	45							
D140	15	16	27	31	46	47	48		
D141	16	17	19	28	31	45			
D142	5	9	19	33	47	48	50		
D143	5	8	9	28	31	33	50		
D144	8	21	28	31					
D145	8	17	21	22	28	31	48		
D146	1	16							
D147	2	5	17	22	50				
D148	2	5	17	22	50				
D149	8								
D150	12	19	46	47					
D151	8	19	21	29	47	48			
D152	19	46	47						

D153	19	21	27	31	46	47			
D154	19	21	27	31	46	47			
D155	19	47	48						
D156	19	47	48						
D157	1	16							
D158	2	15	16	19	21	33	45	46	47
D159	1	18	33	47	48				
D160	1	8	15	16	33	46	47		
D161	1	18	33	47	48				
D162	1	12	15						
D163	8	15	16	28	31				
D164	15	18	47	48					
D165	8	15	16	28	31				
D166	15	18	47	48					
D167	29								
D168	8	25	28	39					
D169	29								
D170	1	16	27	28	29	31	33		
D171	1	2	17						
D172	1	2	17						
D173	1	16	27	28	31	33			
D174	2	5	17	50					
D175	1	15	18	19	46	47			
D176	9	11	12	15	17	18	19	22	45
D177	2	5	17	50					
D178	1	15	18	19					
D179	18	19	21						
D180	9	11	12	15	17	18	19	45	
D181	9	11	12	15	18	27	47	48	
D182	9	11	12	15	18	47	48]		
D183	17	22	45	46	48				
D184	8	9	11	12	15	16	28	31	
D185	8	9	11	12	15	16	28	31	
D186	18								
D187	18								
D188	17	22	45	46	48				
D189	9	11	12	15	18	22	46	47	
D190	16								
D191	1	8	16	28	31	33			

D192	9	11	12	15	18	46	47		
D193	1	8	16	28	31	33			
D194	8	19	21	27	28	47	48		
D195	1	2	16	17	31				
D196	1	5	8	27	28	50			
D197	1	2	16	17	31				
D198	1	2	8	16	17	22	28	31	
D199	1	2	8	16	17	22	28	31	
D200	1	18	28	29	30	31	33		
D201	11	12	15	16	17	45	46		
D202	16	28	29	30	31				
D203	16	28	31						
D204	9	11	12	15	16	27	31		
D205	9	11	12	15	16	27	31		
D206	1	18	19	33					
D207	16	28	31						
D208	1	16	21	33					
D209	11	12	15	16	17	28	31	45	
D210	1	5	8	9	16	28	31	50	
D211	2	8	17	22	28	30	31		
D212	11	12	15	16	17	19	45	46	47
D213	8	28	30	31	50				
D214	1	2	17	18	19	22			
D215	2	17	22	33					
D216	1	2	17	18	19	22			
D217	2	17	22	33					
D218	1	16	28	31					
D219	1	16	28	31					
D220	46	47							
D221	16	29	31	46	47	48			
D222	12	15	16	31					
D223	1	11	12	15					
D224	8	28	29	31					
D225	1	5	8	11	16	18	28	31	50
D226	2	5	9	17	22	50			
D227	19	21	28	31	46	47			
D228	12	15	18	19	21	31			
D229	19	21	28	31	46	47			
D230	2	5	9	17	22	50			

D231	12	15							
D232	12	15	18	46	47	48			
D233	1	2	16	17	22	27	31		
D234	12	15	18	46	47				
D235	2	17	22						
D236	1	2	16	17	22	27	31		
D237	1	2	8	16	17	27	28	31	
D238	1	2	16	17	21	27	28	31	
D239	1	2	17	18	19				
D240	1	2	17	18	19				
D241	19								
D242	19								
D243	12	15	16	27	31				
D244	1	2	15	17	22				
D245	2	15	16	17	22	28	31		
D246	8	16	28	31	33				
D247	1	2	15	17	22				
D248	15	16	21	28	31				
D249	12	15	16	27	31				
D250	2	17	33						
D251	2	17	33						
D252	1	12	15	17	22	33	45	46	47
D253	1	16	28	31	33				
D254	1	18							
D255	1	2	17	22	30	31			
D256	30	31							
D257	19	21	46	47					
D258	2	5	9	17	50				
D259	2	5	9	17	50				
D260	1	18	47	48					
D261	1	12	15	17	22	33	48		
D262	1	16	28	31	33				
D263	12	15	18	47	48				
D264	9	11	12	15	16	22	28	31	
D265	21	28	31	48	50				
D266	8	9	11	12	15	16	28	31	
D267	5	9	21	28	31	50			
D268	19	21	46	47	50				
D269	12	15	18	47	48				

D270	8	12	15	16	28	31			
D271	1	16	19	21	31				
D272	8	12	15	16	28	31			
D273	1	2	8	16	17	28	31		
D274	8	19	21	28	31				
D275	1	2	8	16	17	28	31		
D276	19	21	31	33	48				
D277	8	19	21	28	31				
D278	12	15	18	19					
D279	15	16	33	45	46	47	48		
D280	8	9	11	12	15	17	18	45	
D281	18	29	47	48					
D282	5	12	15	18	19	50			
D283	9	11	12	15	18]				
D284	16	30]							
D285	1	2	16	17	22	28	31		
D286	1	2	16	17	22	28	31		
D287	1	18	29	47	48				
D288	1	18	29	47	48				
D289	15	16	30						
D290	15	16							
D291	15	18	29	47	48				
D292	17	45	46	48					
D293	1	18	27	30	33				
D294	17	45	46	48					
D295	1	18	33						
D296	1	2	12	15	17	22			
D297	12	15	16	28	31				
D298	1	2	12	15	17				
D299	1	2	12	15	17	22			
D300	1	2	17	18	22				
D301	8	11	12	15	18	28	46	47	
D302	12	15	16	28	31				
D303	8	9	11	12	15	18	28	29	47 48
D304	1	2	17	18	19	22	29	47	48
D305	1	2	15	17	22				
D306	1	5	11	12	18	30	50		
D307	9	11	12	15	18	29	47	48	
D308	9	11	12	15	18	27	31		

D309	1	2	12	15	17				
D310	1	2	15	17					
D311	19	29	47	48					
D312	2	5	9	15	18	45	46	47	50
D313	1	2	17	18	22				
D314	15	18							
D315	9	11	12	15	18	27	29	31	
D316	47	48							
D317	47	48							
D318	12	15	18						
D319	11	12	15	18	29	47	48		
D320	5	28	30	31	50				
D321	11	12	15	18	29	47	48		
D322	28	30	31						
D323	17	25	27	39	45	46	47		
D324	19	27	29	30	31	47	48		
D325	1	18	29	33	47	48			
D326	5	9	27	30	31	50			
D327	19	21	27	31					
D328	12	15	18						
D329	19	21	27	31	46	48			
D330	1	18	29	33	47	48			
D331	8	19	21	28	31	33			
D332	5	9	19	21	31	50			
D333	2	5	11	12	15	45	47	48	50
D334	18	46	47						
D335	29	30							
D336	9	11	12	15	18	29	30	47	48
D337	1	5	9	18	29	30	47	48	50
D338	31								
D339	5	9	33	50					
D340	1	16							
D341	1	16							
D342	25	27	39						
D343	1	5	50						
D344	27	29	31	45	48				
D345	11	12	15	28	29	30			
D346	28	29	31	45	48				
D347	1	16	33						

D348	1	18	19	21	33				
D349	33								
D350	2	17							
D351	9								
D352	9								
D353	21								
D354	1	2	16	17	22				
D355	1	2	9	16	17	22			
D356	11	12	15	18	47	48			
D357	11	12	15	18	47	48			
D358	1	5	18	33	50				
D359	1	9	29	30	33				
D360	1	33							
D361	5	11	12	15	18	27	46	47	50
D362	5	9	11	12	27	31	50		
D363	11								
D364	2	45							
D365	11								
D366	25	28	39						
D367	18	46	47						
D368	18	46	47						
D369	5	50							
D370	9								
D371	9								
D372	19	21	46	47					
D373	19	21	46	47					
D374	1								
D375	25	39							
D376	1								
D377	1	5	12	15	17	33	50		
D378	1	5	11	12	15	30	50		
D379	5	17	22	27	30	33	50		
D380	28	31	45	47	48				
D381	2	16	27	28	31	45			
D382	46	48							
D383	9	19	21	29	30	46	47	48	
D384	1	11	12	15	28	31	33		
D385	1	2	18	28	47	48			
D386	29	30							

D387	1	2	18	28	45	47		
D388	1	2	16	17				
D389	1	2	16	17				
D390	1	16	29	30	33	47	48	
D391	17	19	21	27	31	45	46	47
D392	19	21	27	29	30	47		
D393	28							
D394	1	2	5	17	18	22	50	
D395	5	28	29	48	50			
D396	15							

ANEXO A - Carta de apresentação para visita técnica à Embratel



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA

Av. Universitária, 1059 • Sator Universitário
Caixa Postal 86 • CEP 74605-010
Goiânia • Goiás • Brasil
Fone: (62) 3946.1070 • Fax: (62) 3946.1070
www.pucgoias.edu.br • prope@pucgoias.edu.br

Goiânia, 09 de julho de 2015

À
EMBRATEL – Empresa Brasileira de Telecomunicações
Ilmo Senhor Waldemar Jahnecke
Diretor de Operações – Gerencia Técnica e Assistência Técnica de Goiás.

Ref.: realização de pesquisa de natureza acadêmica sobre Redes de Telecomunicações.

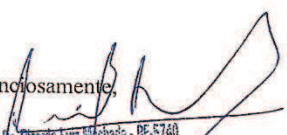
Apresentamos, com finalidade exclusiva de solicitar acesso para realização de uma pesquisa de natureza científica, o acadêmico o Senhor Emerson de Souza Campos.

A investigação a ser realizada pelo pesquisador visa analisar a estrutura de redes SDH no âmbito da Empresa.

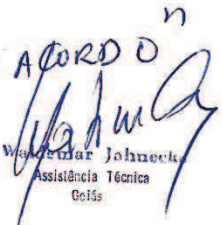
A investigação científica será realizada utilizando uma estrutura de dados e informações de Redes SDH existentes.

O referido pesquisador assegurará o sigilo dos dados coletados, que serão processados e transformados em resultados, que não identificarão as identidades das organizações participantes sem autorização prévia.

Atenciosamente,


Prof. Dr. Ricardo Luiz Machado - RE-5740
Coord. do Mestrado em Eng. de Prod. e Sistemas
PUC Goiás - PROPEC - PÓS-GRADUAÇÃO

Prof. Dr. Ricardo Luiz Machado
Programa de Pós-Graduação Stricto Sensu em Engenharia de Produção e Sistemas
Pontifícia Universidade Católica de Goiás
Fone: (062) 3946-1119
E-mail: mepros@pucgoias.edu.br

"DE ACORDO"

Waldemar Jahnecke
Assistência Técnica
Goiás