



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

UFRPE

DEPARTAMENTO DE ESTATÍSTICA E INFORMÁTICA

CIÊNCIA DA COMPUTAÇÃO

JORGE CANDEIAS DO NASCIMENTO

Projeto integrado de redes ópticas de longa distância e metropolitanas usando algoritmos de inteligência computacional: estudo de caso para o estado de Pernambuco.

TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO

Recife

2017

JORGE CANDEIAS DO NASCIMENTO

Projeto integrado de redes ópticas de longa distância e metropolitanas usando algoritmos de inteligência computacional: estudo de caso para o estado de Pernambuco.

Trabalho de conclusão de curso apresentado ao curso de bacharelado em Ciência das Computação, como parte dos requisitos necessários à obtenção do título de Bacharel.

Orientador: Danilo Ricardo Barbosa de Araújo

Recife
2017

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca Central, Recife-PE, Brasil

N244p Nascimento, Jorge Candeias do.
Projeto integrado de redes ópticas de longa distância e Metropolitanas usando algoritmos de inteligência computacional: estudo de caso para o estado de Pernambuco / Jorge Candeias do Nascimento. – Recife, 2017.
96 f.: il.

Orientador: Danilo Ricardo Barbosa de Araújo.
Trabalho de Conclusão de Curso (Graduação) – Universidade Federal Rural de Pernambuco, Departamento de Computação, Recife, BR-PE, 2018.
Inclui referências.

1. Busca multiobjetivos 2. Kmeans 3. Algoritmos meméticos
4. Clusterização 5. NGAIII 6. PON 7. CWDM 8. Algoritmos evolucionários 9. Topologias de rede I. Araújo, Danilo Ricardo Barbosa, orient. II. Título

CDD 004



MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO (UFRPE)
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

<http://www.bcc.ufrpe.br>

FICHA DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO

Trabalho defendido por Jorge Candeias do Nascimento às 10 horas do dia 06 de fevereiro de 2018, na Sala 23 do CEAGRI-02, como requisito para conclusão do curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, intitulado **Projeto integrado de redes ópticas de longa distância e metropolitanas usando algoritmos de inteligência computacional: estudo de caso para o estado de Pernambuco**, orientado por Danilo Ricardo Barbosa de Araújo e aprovado pela seguinte banca examinadora:

Danilo Ricardo Barbosa de Araújo
DEINFO/UFRPE

Carlos Julian Menezes Araújo
DEINFO/UFRPE

Érica Teixeira Gomes de Sousa
DEINFO/UFRPE

Em memória de minha mãe, quem sempre me apoiou, e de quem sinto saudades todos os dias da minha vida, “ah se houvesse um algoritmo capaz de alcançar o infinito”.

Agradecimentos

Agradeço a Deus por ter me feito chegar até aqui. Agradeço a minha saudosa mãe, Ivonete Candeias de Oliveira, que nunca duvidou dos meus planos e das minhas capacidades em nenhum dia de sua vida. Que me apoiou suprindo e dando as condições necessárias para que eu pudesse lutar as minhas batalhas.

Agradeço aos meus amados irmãos, em ordem alfabética, Albina, Angelina, Sandro e Severino por terem participado da composição de quem sou.

Agradeço a minha amada esposa e companheira, Maria Edilene Pereira Soares, que mudou a sua vida pra se adaptar a minha, sempre me apoiando em minhas decisões e em momentos difíceis, sendo inclusive, a única pessoa que conseguiu administrar a minha mania de trabalhar sem parar.

Agradeço a minha amada filha Maria Izabel Candeias do Nascimento, que, na ausência de minha mãe, surgiu em minha vida como uma luz em um momento de escuridão.

Agradeço ao professor orientador Danilo Araújo, que me mostrou uma “Ponte” perfeita entre uma antiga paixão, as telecomunicações, e uma nova, a computação. Obrigado por todo o conhecimento e norteamento nessa dura jornada.

Agradeço a Universidade Federal Rural de Pernambuco, uma instituição a qual aprendi a chamar de casa. Este agradecimento é em especial ao curso de Ciência da Computação que me fez enxergar mais longe. Também a todos os funcionários que fazem essa universidade funcionar.

A todos os professores que contribuíram para o meu crescimento pessoal e profissional dentro dessa Universidade. Alguns deles, muitas vezes, até me atendendo de madrugada. Obrigado meus queridos, vocês foram um presente de Deus na minha vida.

Agradeço aos meus colegas de curso, que trilharam esta longa estrada comigo. Todos merecem aqui serem lembrados, mas, não posso deixar de falar dos que passaram diversas madrugadas, fins de semanas e feriados ao meu lado, entre uma linha de código e outra, como Jefferson Cabral, Dennis Lucena e Josué Nascimento.

*“ Não se trata de bater forte. Se trata de quanto
você aguenta apanhar e seguir em frente, o
quanto você é capaz de aguentar e continuar
tentando. É assim que se consegue vencer.”
(Trecho do discurso de Rocky Balboa para o
seu filho no filme Rocky Balboa)*

Resumo

Nos dias atuais, várias tecnologias de redes com diferentes preços e adequações vem surgindo no mercado. Um projeto de topologia de rede envolve várias métricas, as métricas são utilizadas para avaliar um projeto. Na avaliação utilizamos métricas como, por exemplo, as métricas de robustez (que auxiliam na capacidade da rede de se recuperar de uma falha), probabilidade de bloqueio e consumo energético. O ideal para otimizar a infraestrutura, em um projeto de redes, seria usar as tecnologias mais recentes, só as mais eficientes, mesmo que tais tecnologias sejam mais caras. Entretanto, das métricas a serem consideradas neste tipo de projeto, uma delas é o custo (capital empregado). Portanto, nem sempre é viável usar o que há de mais caro no mercado. Muitas questões técnicas podem ajudar no controle das métricas destes projetos, dentre elas está a topologia da rede (interconexão do *links*). Algoritmos evolucionários multiobjetivos (algoritmos inspirados na evolução das espécies) vem sendo estudados no estado da arte para a concepção de topologias de rede. Ao mesmo tempo, algoritmos de *clustering* (algoritmos especializados em separar amostras em grupos) vem sendo usados em outros tipos de estudos em redes. Este estudo teve por objetivo fazer o uso de algoritmos de inteligência computacional na construção de um projeto de topologia de redes, utilizando o estado de Pernambuco como um estudo de caso. Em uma primeira etapa do estudo, foi usado um algoritmo de *clustering* na divisão do estado em grupos. A intenção dessa parte do trabalho foi de medir a cobertura da rede em relação a toda a dimensão do estado, e, dessa forma garantir a completude da rede. Além disso a etapa de *clustering* também objetivou propor um modelo de controle de custos através de mescla de diferentes tecnologias para a rede (Passiva ou ativa) dependendo da função do trecho de rede. Em uma segunda etapa, foi usado um algoritmo evolucionário multiobjetivo para compor diversas topologias de redes que atendiam aos *clusters* criados na etapa anterior. Esse algoritmo, evoluiu as diversas topologias de rede em função de melhorar quatro métricas, Probabilidade de Bloqueio, Custo, Consumo energético e Conectividade algébrica. O algoritmo multiobjetivo foi concebido como um algoritmo memético, e, após um conjunto de execuções, foram comparados os desempenhos do algoritmo com e sem a alteração. Os resultados dos testes, na primeira etapa, mostraram que as técnicas de *clustering* são bastante eficientes e adaptáveis ao objetivo proposto tanto no que se diz respeito a completude da rede quanto no controle de custos. Já na segunda etapa, ou etapa de busca multiobjetivos, foi constatado, através do uso de um indicador de qualidade (o *hypervolume*), que houve melhora do algoritmo em relação a convergência e a diversidade à curva de Pareto, com o uso em sua nova forma como algoritmo memético.

Palavras Chave: Busca multiobjetivos, Kmeans, Algoritmos Meméticos, Clusterização, NSGAIII, PON, CWDM, Algoritmos Evolucionários, Topologias de rede.

Abstract

Nowadays, several network technologies with different prices and adaptations are appearing in the market. A network topology project involves several metrics; the metrics are used to evaluate a project. In the evaluation we use metrics such as robustness metrics (which help in the network's ability to recover from a failure), blocking probability and energy consumption. The best way to optimize infrastructure in a network design would be to use the latest technologies, only the most efficient ones, even if such technologies are more expensive. However, of the metrics to be considered in this type of project, one of them is the cost (capital employed). Therefore, it is not always feasible to use the most expensive ones on the market. Many technical issues can help control the metrics of these projects, among which is the network topology (link interconnection). Multiobjective evolutionary algorithms (algorithms inspired by the evolution of the species) have been studied in the state of the art for the conception of network topologies. At the same time, clustering algorithms (algorithms specialized in separating samples into groups) have been used in other types of network studies. This study aimed to make use of computational intelligence algorithms in the construction of a network topology project, using the state of Pernambuco as a case study. In a first stage of the study, a clustering algorithm was used in the division of the state into groups. The intention of this part of the work was to measure the coverage of the network in relation to the entire size of the state, and thus ensure the completeness of the network. In addition, the clustering stage also aimed to propose a cost control model through the merging of different technologies for the network (Passive or active) depending on the function of the network segment. In a second step, an evolutionary multiobjective algorithm was used to compose several network topologies that served the clusters created in the previous step. This algorithm has evolved the various network topologies in order to improve four metrics, Blocking Probability, Cost, Energy Consumption and Algebraic Connectivity. The multiobjective algorithm was designed as a memetic algorithm, and, after a set of executions, the algorithm performances were compared with and without the alteration. The results of the tests, in the first stage, showed that the clustering techniques are quite efficient and adaptable to the proposed goal both in terms of network completeness and cost control. Already in the second stage, or multiobjective search stage, it was verified, through the use of a quality indicator (hypervolume), that there was an improvement of the algorithm in relation to convergence and diversity to the Pareto curve, with the use in its new form as memetic algorithm.

Keywords: Multiobjective Search, Kmeans, Memory Algorithms, Clustering, NSGAIII, PON, CWDM, Evolutionary Algorithms, Network Topologies.

Lista de ilustrações

Figura 1 – Exemplo de topologia de rede para o estado de Pernambuco.	16
Figura 2 – Exemplo de topologia de rede para o estado de Pernambuco com outra topologia.	16
Figura 3 – Exemplo de <i>cluster</i> de região geográfica.	18
Figura 4 – Fluxo da metodologia.	24
Figura 5 – Convergência genética	36
Figura 6 – Gráfico multiobjetivos.	38
Figura 7 – Fronteiras de pareto	39
Figura 8 – Classificação de soluções não dominadas utilizada pelo NSGAII. . .	41
Figura 9 – Avaliação CD das soluções que estão na mesma frente de Pareto. .	42
Figura 10 – Quinze pontos de referência estruturados são mostrados em um hiper plano normatizado.	43
Figura 11 – Solução do problema DTLZ1 obtida pelo NSGAIII.	44
Figura 12 – Área do <i>hypervolume</i>	49
Figura 13 – Fluxo principal do projeto.	51
Figura 14 – Matriz de conexão com seleção de seis interconexões.	54
Figura 15 – Mapa de Pernambuco com busca local	56
Figura 16 – Fluxo do multiobjetivos.	60
Figura 17 – Evolução da rede PON e alcance, retirado de (HERON et al., 2008 p. 163).	67
Figura 18 – Gráfico de índice de <i>silhouette</i> com valor de <i>K cluster</i> de 4 a 30. . .	68
Figura 19 – Mapa de Pernambuco após <i>clustering</i> com <i>K</i> igual a 4.	69
Figura 20 – <i>emphypervolume</i> da média de 11 execuções.	73
Figura 21 – Comparação por <i>hypervolume</i> com e sem busca iteração 100. . . .	74
Figura 22 – Comparação por <i>hypervolume</i> com e sem busca iteração 200. . . .	74
Figura 23 – Comparação por <i>hypervolume</i> com e sem busca iteração 300. . . .	75
Figura 24 – Comparação por <i>hypervolume</i> com e sem busca iteração 400. . . .	75
Figura 25 – Comparação por <i>hypervolume</i> com e sem busca iteração 500. . . .	76
Figura 26 – Pior resultado das execuções sem busca, resultado capturado na iteração 300.	77
Figura 27 – Melhor resultado das execuções sem busca, resultado capturado na iteração 300.	77
Figura 28 – Resultado mais próximo da média das execuções sem busca, resultado capturado na iteração 300.	78
Figura 29 – Pior Resultado das execuções com busca, resultado capturado na iteração 300.	79

Figura 30 – Melhor Resultado das execuções com busca, resultado capturado na iteração 300.	79
Figura 31 – Resultado mais próximo da média das execuções com busca, resultado capturado na iteração 300.	80
Figura 32 – Execução 1 sem busca, resultado final, capturado na iteração 500.	82
Figura 33 – Execução 1 com busca, resultado final, capturado na iteração 500.	82
Figura 34 – Rede mais barata encontrada pelo algoritmo sem a busca local.	83
Figura 35 – Mapa da rede com menor CAPEX encontrada na iteração 500 do método sem busca local.	83
Figura 36 – Rede mais barata encontrada pelo algoritmo com a busca local.	84
Figura 37 – Mapa da rede com menor CAPEX encontrada na iteração 500 do método com busca local.	85
Figura 38 – Troca de cabeça de <i>cluster</i> em benefício de algumas métricas.	88

Lista de tabelas

Tabela 1 – Tabela de comparação de trabalhos relacionados com este trabalho.	29
Tabela 2 – Tabela de soluções não dominadas.	39
Tabela 3 – Tabela de configuração de etapa de clustering.	59
Tabela 4 – Tabela de configuração da etapa de busca multiobjetivos.	61
Tabela 5 – Máxima iteração por K cluster	64
Tabela 6 – Índice de Silhouette por quantidade K de cluster	65
Tabela 7 – Distância entre Centroides de cluster	66
Tabela 8 – Distância Cidade a cidade dentro do cluster.	70
Tabela 9 – Distância máxima, Mínima e Média entre as cidades e o centroide de cada cluster	71
Tabela 10 – Tabela de estatísticas do tempo, valores em horas e minutos.	72
Tabela 11 – Legenda	81

Lista de abreviaturas e siglas

CA	Conectividade Algébrica
CAPEX	Capital Expenditures
CD	Crowding Distance
CE	Consumo Energético
CNS	Complex Network Simulator
DWDM	Dense Wavelength Division Multiplex
GA	Genetic Algorithm
GB	Gigabyte
HP	Hewlett Packard
ID	Identity
IDE	Integrated Development Environment
IP	Internet Protocol
MPLS	Multiprotocol Label Switching
PB	Probabilidade de Bloqueio
PIB	Produto Interno Bruto
POP	Point Of Presence
RAM	Random Access Memory - Memória de Acesso Aleatório
REN	Research Education Network
WDM	Wavelength Division Multiplexing
XML	EXtensible Markup Language

Sumário

1	INTRODUÇÃO	15
1.1	Motivação	19
1.2	Justificativa	20
1.3	Problema da Pesquisa	21
1.4	Objetivos	22
1.5	Resumo da metodologia	22
1.6	Organização do texto	24
2	REVISÃO DA LITERATURA	25
2.1	Trabalhos sobre projetos de redes usando meta-heurísticas	25
2.2	Trabalhos envolvendo técnicas de <i>clustering</i>	27
2.3	Trabalhos que propõem topologias sem uso de MOAEs.	28
3	FUNDAMENTAÇÃO TEÓRICA	31
3.1	Fundamentação teórica sobre algoritmos de <i>clustering</i>	31
3.1.1	<i>Kmeans</i>	31
3.1.2	Índice de <i>Silhouette</i>	32
3.2	Fundamentação teórica para algoritmos multiobjetivos	33
3.2.1	Algoritmos Genéticos	33
3.2.2	Algoritmos Meméticos	34
3.2.3	Problemas multiobjetivos	37
3.2.4	Noção de Dominância e Frente de Pareto	38
3.2.5	O Framework NSGA	40
3.2.6	NSGAI	40
3.2.7	NSGAIII	43
3.3	Métricas para Redes e Qualidade	44
3.3.1	Probabilidade de Bloqueio (PB)	44
3.3.2	Custo de implantação CAPEX	45
3.3.3	Consumo energético em watts	46
3.3.4	Conectividade Algébrica	47
3.3.5	Erlang	48
3.3.6	<i>Hypervolume</i>	48
4	PROPOSTA	50
4.1	Pré-processamento baseado em <i>Clustering</i> por média de distâncias	52

4.1.1	Planejamento da aplicação do <i>Kmeans</i>	52
4.2	Otimização baseada em algoritmos evolucionários multiobjetivos com busca local	53
4.2.1	Planejamento da aplicação do NSGAIII	53
4.3	Arranjo experimental	58
4.3.1	Arranjo experimental da etapa de pré-processamento baseado em <i>clustering</i>	58
4.3.2	Arranjo experimental da etapa de otimização baseada em algoritmos evolucionários multiobjetivos com busca local	59
5	RESULTADOS	63
5.1	Resultados do pré-processamento baseado em <i>clustering</i> por médias de distâncias	63
5.2	Resultados da Otimização baseado em algoritmo evolucionários multiobjetivos com busca local	72
6	DISCUSSÃO	86
6.1	Discussão geral	86
6.2	Discussão da etapa de pré-processamento baseado em <i>clustering</i> por média de distâncias	86
6.3	Discussão da otimização baseada em algoritmos multiobjetivos com busca local	87
7	CONCLUSÃO	90
7.1	Limitações identificadas	91
7.2	Trabalhos Futuros	91
	REFERÊNCIAS	93

1 Introdução

Segundo ARAÚJO, Bastos-Filho e Martins Filho (2014, p. 11), (RAMASWAMI; SIVARAJAN; SASAKI, 2010; KEISER, 2003) a constituição da infraestrutura de redes de transporte e metropolitana, que compõem o provimento de serviços de telecomunicações, os quais, exigem altas taxas de transmissão, é feita basicamente por fibra óptica. Neste contexto pode-se dizer que uma escolha bem recomendada para um projeto de infraestrutura de redes de transporte é um projeto de redes de fibra óptica. Os enlaces ópticos de longa distância utilizam geralmente WDM (*Wavelength-Division Multiplexing*) ou DWDM (*Dense WDM*), tendo um espaçamento de 100Ghz para WDM e de 50Ghz para DWDM, em que o espectro óptico utilizado é dividido em canais espalhados de maneira uniforme (ARAÚJO; BASTOS-FILHO; MARTINS FILHO, 2014, p. 11). Entretanto os equipamentos como filtros e LASERs de alta precisão, geralmente, são mais custosos, portanto, o elevado custos desses equipamentos demandam cuidados com o custo final da rede.

Além da tecnologia de multiplexação usada (WDM/DWDM), a topologia da rede (maneira que os links se interconectam) exerce um importante papel no projeto de uma nova rede ou na expansão de uma rede existente. Com o passar dos anos, pesquisas têm investigado as topologias físicas para obter métricas que ajudem as operadoras a planejar melhor as redes e otimizar o desempenho e os custos (PAVAN et al., 2015, p. 578). Nguyen e Truong (2013, p. 38), afirmam que, em um projeto de topologia de rede óptica, 90% do custo é oriundo do custo dos *links* de fibra óptica e de sua implantação, enquanto que apenas 10% é referente a outros equipamentos. Para melhor ilustrar como a topologia da rede pode influenciar as métricas de um projeto de rede, como probabilidade de bloqueio (PB), o custo e a conectividade algébrica (uma métrica de robustez) a seguir apresentaremos duas topologias. A Figura 1 mostra uma rede (Oriunda de experimentos desta pesquisa) simulada com o tráfego de 100 erlangs, com PB igual a 0,151492, o custo de implantação (CAPEX - *capital expenditure*) igual a 1722,74 u.m. (Unidades monetárias), e a conectividade algébrica igual a 1,38. PB é a probabilidade estatística da rede não atender a uma solicitação de demanda de tráfego, CAPEX é o custo de implantação e, por fim, conectividade algébrica é uma métrica que fornece informações sobre a capacidade da rede de se recuperar após uma falha.

Figura 1 – Exemplo de topologia de rede para o estado de Pernambuco.



Fonte: o autor.

Cada ponto da Figura 1 que interliga as cidades: Recife, Custódia, Ouricuri, Petrolina e Floresta, representa um POP (*Point of presence*) ou nó terminal da rede. Nesta rede, os *links* que interconectam os POPs, são *links* de rede ativa ou *link* de *backbone*, geralmente, constituídos de redes WDM ou DWDM. As cidades circunvizinhas às que possuem POP, devem ser atendidas por redes metropolitanas, ou redes PON (*passive optical network*).

Já na Figura 2 (também oriunda de experimentos desta pesquisa), apresentamos uma rede com os mesmos nós terminais utilizados na Figura 1, no entanto, interligados de maneira diferente, ou seja, com outra topologia, e, simulada com o mesmo tráfego. O motivo da escolha dessas duas topologias foi justamente o fato de ser a mesma demanda de projeto tendo suas métricas sendo variadas, exclusivamente, em função da topologia.

Figura 2 – Exemplo de topologia de rede para o estado de Pernambuco com outra topologia.



Fonte: o autor.

Com esta topologia de rede, da Figura 2, os valores de PB foram igual a 0,215146, CAPEX igual a 1888,16 u.m. e conectividade algébrica igual 1,00. Ou seja, uma mesma demanda de projeto de rede, com a mesma posição de nós terminais,

mas com a topologia diferente, acabou ficando mais cara. Além do que, ainda teve uma probabilidade de bloqueio maior, e também, teve uma robustez reduzida. Este trabalho propõe novas abordagens para encontrar diferentes topologias de rede que apresentem melhores compromisso entre as métricas destes três objetivos, conforme ilustrado pelas Figuras 1 e 2, e mais um, o consumo energético da rede.

Muitos trabalhos da literatura usaram algoritmos de otimização multiobjetivos para propor diversas topologias de rede. Dupleix, Araújo e Bastos-Filho (2016, p. 1), investigaram uma forma otimizada de propor o projeto de uma topologia de rede para o Haiti. O trabalho usou o framework NSGAII proposto por (DEB et al., 2002; SRINIVAS; DEB, 1994; DEB; JAIN, 2014). O NSGAII foi usado como meta heurística adotada para solucionar a busca multiobjetivos. Os autores também destacaram a importância da heurística, apontando as vantagens que obteve em relação às métricas quando houve a inclusão da heurística.

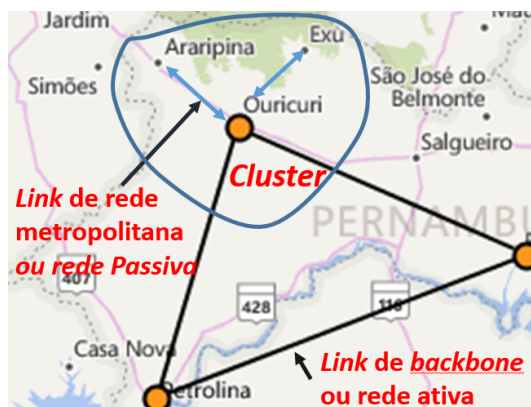
Em outra linha de pesquisa, algoritmos de *clustering* (separação de subgrupos de algum tipo de amostra por características em comum) foram usados em outros tipos de redes. Younis, Krunz e Ramasubramanian (2006, p. 20), investigaram como as técnicas de *clustering* são benéficas para o prolongamento da vida útil das baterias em uma rede de sensores sem fio. Neste artigo foi descrito um rodízio do que, o trabalho, chamou de cabeça de *cluster*, o qual seria um sensor escolhido por um algoritmo. Este sensor assume a função de receber a transmissão dos sensores que estavam dentro do seu *cluster* e faz a retransmissões para sensores mais distantes, como um roteador. Já em outro trabalho, também do seguimento de *clustering* em redes de sensores sem fio, Maheswari, Meenalochani e Sudha (2016, p. 378), concluíram que esta economia de energia nas baterias dos sensores é ainda maior quando o cabeça de *cluster* é localizado no centroide do *cluster*.

Exceto pelo rodízio mencionado no trabalho de Younis, Krunz e Ramasubramanian (2006, p. 20), essa estrutura de grupos, das redes de sensores sem fio, pode ser adaptada para um *cluster* de região geográfica, onde o cabeça de *cluster* será, de forma permanente, a cidade onde ficará o POP e as cidades que estão dentro do *cluster* seriam atendidas por redes passivas ou redes PON. Ou seja, a abordagem usada para redes de sensores sem fios pode servir de inspiração para uma estratégia inédita em projetos de redes ópticas, que seria combinar algoritmos de *clustering* com técnicas de busca multiobjetivos.

A Figura 3 ilustra como este conceito de *clustering* está sendo adaptado para a proposta deste trabalho. Na Figura 3 é possível observar um *cluster* de região geográfica e o esquema de atendimento das cidades por redes de longa distância e redes metropolitanas. O *cluster* contém as cidades de Ouricuri, Araripina e Exu, onde Ouricuri detém o POP sendo, portanto, atendida por rede ativa e servindo, inclusive,

como uma espécie de roteador para as outras cidades. Já as cidades de Araripina e Exu são atendidas por redes metropolitanas ou redes passivas. A Figura 3 também destaca a rede de longa distância ou rede de *backbone* geralmente constituídas por tecnologia WDM ou DWDM.

Figura 3 – Exemplo de *cluster* de região geográfica.



Fonte: o autor.

De fato, as técnicas de *clustering* são muito úteis e adaptáveis a diversos problemas do mundo real. Com a intenção de otimizar a alocação de voluntários em ONGs (organização não governamental) que mais se adequassem às características de cada um Bezerra, B. e Macario (2016, p. 223), propuseram o uso de *clusters* para um sistema de recomendação de voluntários de ONGs no Brasil. Os autores usaram os métodos de K-Means e K-Medoid (MACQUEEN, 1967; KAUFMANN; ROUSSEUW, 1987; JAIN, 2010), mostrando como um sistema de *clustering* é poderoso na separação de membros de um grupo por questões em comum e bastante adaptável a diversos problemas.

(CHAVES; BASTOS-FILHO; MARTINS-FILHO, 2010, p. 1), utilizaram o NSGII para propor topologias de redes ópticas, o trabalho afirma que um projeto de topologia de uma rede é um problema multiobjetivo sobre um espaço de projeto multivariável, já que, o projetista tem que satisfazer simultaneamente várias questões de desempenho.

A questão é que problemas multiobjetivos não retornam apenas uma solução e sim um conjunto delas, ou seja, quando um problema tem múltiplos objetivos a solução para este problema não é única e sim um conjunto de soluções, mais conhecidas como, soluções do Pareto ótimo (DEB et al., 2002, p. 192). Para um projetista, a tarefa de visualizar todas as soluções de forma manual não é trivial, principalmente porque às vezes este conjunto pode ser muito grande. Há várias pesquisas na literatura que estudam o problema dos múltiplos objetivos em uma topologia de rede de fibra óptica. Em muitas delas, como em (DUPLEIX; ARAÚJO; BASTOS-FILHO, 2016, p. 1), (FIGUEIREDO et al., 2016, p. 409), a métrica utilizada para medir a qualidade do

conjunto de soluções ótimas foi o *hypervolume* (ZITZLER et al., 2003, p. 117), que é um indicador que pode avaliar simultaneamente a convergência e a diversidade das soluções oferecidas por um algoritmo multiobjetivo.

Outras linhas de pesquisa mostram que a cognição (aquisição de conhecimento por percepção) está tão presente nos projetos de rede que houveram trabalhos que tentaram sistematizá-la como nos algoritmos apresentados em (DURÁN et al., 2011; N. FERNÁNDEZ et al., 2012). Já em Nascimento et al. (2016), foi realizado um estudo para propor uma topologia de rede óptica para o estado de Pernambuco considerando resiliência e custo. O artigo propôs uma estratégia manual utilizando tabelas para selecionar em torno de 10%, de um total de 185 cidades do estado, em que estes 10% do total de cidades receberiam os POP's. O resultado da técnica foi uma rede com dezoito cidades com POPs que não gerou uma rede com uma boa cobertura no estado. Por isso, por meio de um modelo de decisão baseado na percepção que foi obtida das áreas não cobertas, foi proposto a inclusão de algumas cidades nomeadas de geograficamente estratégicas. O fato é que estas cidades chamadas de geograficamente estratégicas, que foram escolhidas por cognição, melhoraram sensivelmente a topologia da rede proposta, pois, aumentaram a cobertura da rede em relação ao estado de forma mais satisfatória.

Este trabalho propõe uma nova abordagem para encontrar topologias de rede de fibra óptica, em um estudo de caso no estado de Pernambuco, utilizando algoritmos de inteligência computacional. A nova abordagem, usa técnicas de *clustering*, para dividir uma região geográfica, como um estado, em sub partes chamadas *clusters*. Depois, em conjunto com um algoritmo evolucionário memético, propor o atendimento de serviços de dados para cada uma dessas partes, e, decidir quais cidades ficarão com os POP's (idealmente uma por *cluster*).

1.1 Motivação

Algoritmos de *clustering* já foram usados em outros tipos de redes para a definição de topologias, como no caso de redes de sensores sem fios (YOUNIS; KRUNZ; RAMASUBRAMANIAN, 2006, p. 20), (MAHESWARI; MEENALOCHANI; SUDHA, 2016, p. 378). Este trabalho investigou o uso de algoritmos de *clustering* e de busca multiobjetivos para permitir o projeto integrado de redes ópticas de longa distância e redes metropolitanas, considerando as penalidades da camada física (enfraquecimento do sinal devido ao meio) e limitações inerentes à propagação do sinal óptico em longas distâncias. Em meio à integração dos algoritmos de *clustering* e multiobjetivos, também foi implantando um sistema de busca local empregando conhecimento específico do problema ao algoritmo de busca multiobjetivos (a descoberta de cidades vizinhas à detentora do POP). A principal motivação deste estudo é encontrar soluções de projetos

de redes ópticas com bom compromisso entre os objetivos conflitantes, por exemplo PB e CAPEX, e em menos repetições de evoluções do que os métodos já propostos na literatura. Como estudo de caso, a abordagem proposta foi usada para propor uma topologia de rede para o estado de Pernambuco considerando o posicionamento dos nós terminais como uma variável de decisão do algoritmo de *clustering* e do algoritmo multiobjetivos.

1.2 Justificativa

Construir um projeto de topologia de redes é um problema de otimização multiobjetivos, já que, o projetista tem que se preocupar com vários objetivos conflitantes, por exemplo, custo e desempenho. Dentre as práticas para mitigar este problema usa-se, por exemplo, conhecimentos empíricos, como a experiência dos profissionais envolvidos no projeto, ou o conhecimento de detalhes dos locais de implantação. Entretanto, apesar das empresas de telecomunicações realizarem o planejamento da topologia da rede considerando questões técnicas, o custo e diretrizes internas, em algumas situações, aspectos políticos e culturais podem influenciar nesse planejamento. Todavia, mesmo considerando minuciosamente os detalhes técnicos, a equalização de questões conflitantes como desempenho e custo são muito difíceis de mitigar.

Inteligência computacional vem sendo investigada na academia, entre outras coisas, como uma alternativa para o tratamento de problemas complexos (NP-completos e NP-difíceis) e como uma forma de facilitar o trabalho de projetistas (NGUYEN; TRUONG, 2013; N. FERNÁNDEZ et al., 2012; DURÁN et al., 2011). Muitas vezes, um algoritmo especializado em determinada função é capaz de trazer à luz soluções que seriam difíceis de identificar sem um método inteligente de apoio. Neste sentido, este trabalho mostra que para uma busca multiobjetivos, em um projeto de redes, um algoritmo evolucionário multiobjetivo, integrado com um algoritmo de *clustering*, é capaz de auxiliar no projeto. Estes algoritmos podem dar respostas oriundas de questões complexas bem comuns em um projeto integrado de redes ópticas de longa distância e redes metropolitanas.

Das questões importantes em um projeto de redes uma delas é a cobertura da rede, ou seja, se a rede cobre à área que se popõe a atender. Neste sentido o algoritmo de *clustering* foi usado para subdividir o estado em grupos de cidades, como forma de garantir a cobertura do atendimento ao estado todo pela rede. A partir de inspirações do estado da arte, a maneira idealizada para o tratamento deste problema foi separar o estado em grupos de cidades, garantido o atendimento a cada grupo por redes WDM ou DWDM interligando os grupos através das cidades detentoras de POPS. Após isso, as demais cidades do grupo podem ser atendidas por redes PON ou CWDM (Coarse WDM) garantindo então, o atendimento a cada cidade dentro de cada grupo, e

consequentemente, ao estado todo. Desta forma foi possível evitar a necessidade de uso de intervenções externas ao método, praticadas em Nascimento et al. (2016), para garantir a cobertura da rede a todo o estado.

Uma outra questão a ser tratada em um projeto de topologia de redes é a localização dos nós terminais, que de acordo com o levantamento bibliográfico feito, geralmente, é passada para o algoritmo como uma informação estática. Com a intenção de encontrar topologias com melhores resultados em função das métricas, neste estudo, os nós terminais não foram passados para o algoritmo como um informação estática. Os nós foram tratados como variáveis de decisão dos dois algoritmos integrados, o de *clustering* e o multiobjetivos. Primeiramente, como uma variável de decisão do algoritmo de *clustering*, depois, como uma variável de decisão do algoritmo multiobjetivos auxiliado por uma busca local. A busca local foi usada para a variação de cidades dentro de cada *clusters* como a escolhida para ser o nó terminal. Isso de acordo com a configuração de rede que resultasse nas topologias com melhores métricas de CAPEX, probabilidade de bloqueio PB ¹, consumo energético em *watts* e conectividade algébrica ². No Capítulo de fundamentação teórica essas métricas serão melhor detalhadas. Ou seja, para elucidar soluções com bom compromisso entre os objetivos conflitantes de desempenho, custo, consumo e robustez, o algoritmo de busca multiobjetivos foi usado para responder literalmente à pergunta, qual cidade, dentro do *cluster* que atende aquela microrregião dentro do estado, ficará como um tipo de cabeça de cluster (YOUNIS; KRUNZ; RAMASUBRAMANIAN, 2006, p. 20).

1.3 Problema da Pesquisa

A hipótese é que o custo da rede pode diminuir e o desempenho pode melhorar se forem usadas cidades vizinhas de uma cidade que seria a “preferencial” para ser um concentrador de tráfego ou detentora do POP.

Portanto, o problema de pesquisa abordado nessa monografia pode ser formalmente definido da seguinte forma: dada uma matriz de tráfego de *Internet* para um conjunto de $U = \{u_1, u_2, \dots, u_n\}$ cidades, o objetivo é determinar o conjunto de cabeças de *cluster* $N = \{n_1, n_2, \dots, n_n\}$ e um conjuntos de enlaces de fibra óptica $E = \{e_1, e_2, \dots, e_n\}$ que minimize o conjunto de métricas de {CAPEX, PB, consumo energético e conectividade algébrica}, sujeito à restrição de que a topologia da rede seja um grafo conectado.

¹ PB (probabilidade estatística da rede não atender a uma solicitação de demanda de tráfego).

² Conectividade algébrica, uma métrica de robustez: altos valores de conectividade algébrica melhoram a capacidade de uma rede de se recuperar de uma falha.

1.4 Objetivos

O objetivo geral desta pesquisa foi investigar o uso de algoritmos de *clustering* e busca multiobjetivos acrescido de uma busca local, para fazer o projeto integrado de redes ópticas de longa distância e redes metropolitanas, considerando as penalidades da camada física e limitações inerentes à propagação do sinal óptico em longas distâncias.

Já os objetivos específicos foram:

- 1) Construir *clusters* mais coesos o possível subdividindo o estado de Pernambuco em pequenos grupos para que cada um deles possa ser atendido por um POP, garantindo assim uma boa cobertura da rede sobre o estado e o atendimento das demais cidades pela rede PON ou CWDM sem exceder suas características.
- 2) Garantir uma grande diversidade de configurações de topologias de rede com a variação das cidades dentro de cada *cluster* como sendo a portadora do POP, promovendo boas topologias subótimas.
- 3) Garantir avaliações subótimas de métricas de Probabilidade de Bloqueio, CAPEX, consumo energético e conectividade algébrica.
- 4) Incluir conhecimento específico ao algoritmo de busca multiobjetivos de maneira a melhora-lo.

1.5 Resumo da metodologia

Os resultados desta pesquisa foram obtidos a partir de experimentos baseados em simulações computacionais. As implementações foram feitas utilizando a linguagem Java, e a IDE adotada foi o Eclipse Neon. Os experimentos descritos nesta monografia foram todos executados em um notebook HP ProBook 4430s, processador 2.10 Ghz, Intel Core I3 com 4 GB de memória RAM e com sistema operacional Windows 7 de 64 Bits.

As ferramentas e simuladores de redes adotados neste estudo foram usados em estudos anteriores, como em Dupleix, Araújo e Bastos-Filho (2016, p. 1), Bezerra, B. e Macario (2016, p. 223). Para montar a base de dados com informação sobre as cidades como população e PIB foram coletados dados do INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (2010), as informações de localização (latitude e longitude) foram coletada do Google Maps (GOOGLE, 2016). Com isso foi montado um arquivo GML (*Geography Markup Language*) que é um arquivo XML (*Extensible Markup Language*) que serve para modelagem, transporte e armazenamento de informações geográficas (RIBEIRO; FARIAS; ROQUE, 2004; S, 2010).

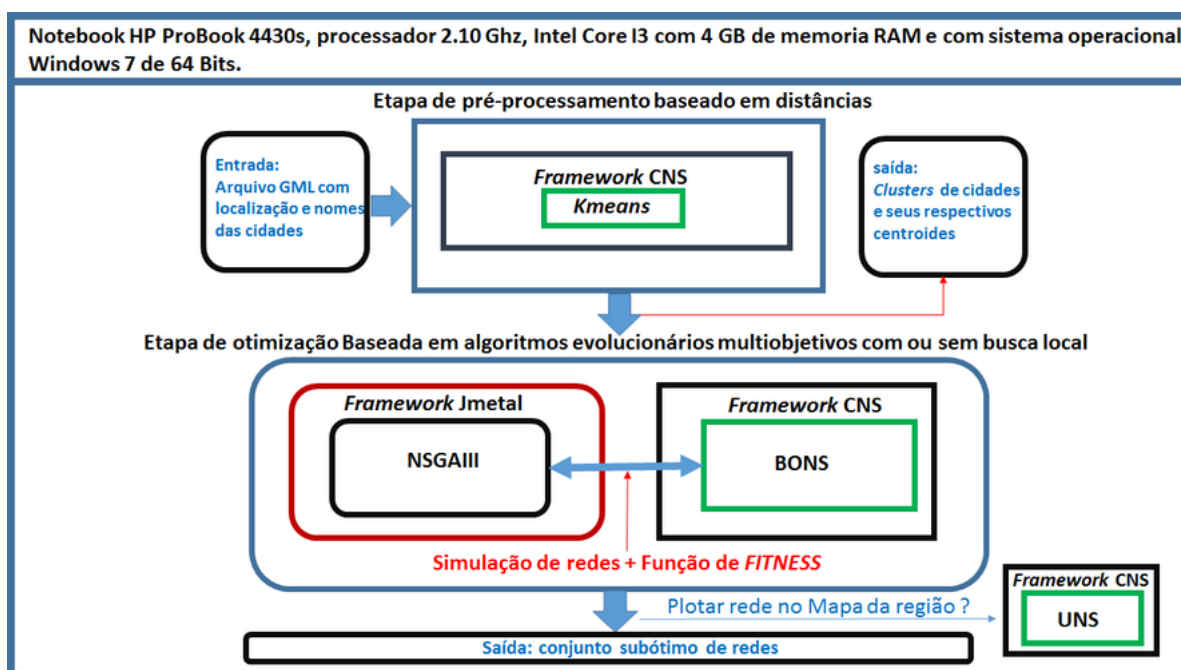
O GML então foi usado como base de dados no CNS (*Complex Network Simulator*), um *framework* produzido em (ARAÚJO, 2015). O CNS já tem em sua implementação os códigos do *Kmeans* e também de duas outras aplicações que foram necessárias para as simulações e cálculos de métricas das topologias geradas neste trabalho. As duas aplicações são o UNS (*Universal Network Sketcher*), um simulador gráfico de redes, e, o BONS (*Brazilian optical network simulator*), que gera valores de métricas como probabilidade de bloqueio e CAPEX a partir de uma simulação.

Para a implementação da busca multiobjetivos foi usado o *framework* Jmetal (DURILLO; NEBRO; ALBA, 2010; NEBRO; DURILLO; VERGNE, 2015) um *framework* já bastante usado em estudos sobre algoritmos multiobjetivos e que implementa de forma geral uma série de algoritmos evolucionários multiobjetivos. Desta forma, foi utilizado uma das práticas aconselháveis das metodologias ágeis, o reuso de software. Dentre os algoritmos existentes está o NSGAIII, o qual, foi usado neste trabalho.

Após estudos iniciais, delimitação do escopo e definição do ambiente como levantamento de dados das cidades, e escolha da linguagem de programação, foi realizado um pré-processamento baseado em *clustering* por média de distâncias. Esse pré-processamento revelou questões interessantes como quantidades de iterações necessárias para *Kmeans* realizar uma etapa de *clustering*, ações de mitigação de controle de distância, e tamanhos de *K cluster* que resultaram em bons indicadores de qualidade de *cluster*. Após essa etapa foi possível escolher que valores de *K cluster* deveriam ser entregues aos algoritmo multiobjetivos.

A etapa de busca multiobjetivos foi realizada com o algoritmo NSGAIII contando com uma adição de um método que o deixou com uma busca local. Esta busca local foi escolhida conforme o problema, e, diante do escopo, foi averiguado a necessidade de uma descoberta de vizinhança. A Figura 4 sintetiza a integração dos recursos usados na metodologia.

Figura 4 – Fluxo da metodologia.



Fonte: o autor.

1.6 Organização do texto

Esta monografia foi escrita de maneira a descrever de forma geral as características de um projeto de rede de dados, até a forma mais específica abordada aqui. Desta forma este texto foi dividido em 7 capítulos. No Capítulo 2 são apresentados os conteúdos da revisão da literatura, onde foram discutidas as obras que mais se relacionam com este trabalho. O Capítulo 3 traz a fundamentação teórica, onde foi detalhado de maneira mais profunda os conhecimentos técnicos usados neste trabalho. O Capítulo 4 traz a proposta do projeto, detalhando de maneira mais profunda como foram elaborados todos os testes da parte de *clustering* e multiobjetivos. No Capítulo 5 são apresentados os resultados. No Capítulo 6 é feita uma discussão geral sobre os detalhes desses resultados. Finalmente o Capítulo 7 fornece considerações finais e sugestões para trabalhos futuros.

2 Revisão da Literatura

Este trabalho adotou as técnicas de *clustering* e de busca multiobjetivo no projeto de topologias de redes de fibra óptica para propor uma nova abordagem para planejamento de redes ópticas. Trabalhos relacionados são bem diversos, visam a construção de redes por meio de algoritmos de busca evolucionária multiobjetivos (MOEAs), buscando soluções subótimas no projeto de topologias de redes. Há outros que fazem comparação de dois ou mais algoritmos no desempenho desta questão, mas também, há outros que propõem topologias sem uso de MOEAs. Ainda há os que propõem algoritmos heurísticos para construção de topologias de redes. Já trabalhos relacionados a algoritmos de *clustering* para segmentação de subgrupos, buscam separação de indivíduos por alguma característica em comum, ou, comparam o desempenho de diferentes técnicas de *clustering* na obtenção de algum tipo de subgrupo.

A pesquisa destes artigos em particular foi realizada com as palavras e frases relacionadas a essa linha de estudo como “*Optical networks*”; “*Multi-objective Optimization*”; “*clustering Wireless Sensor Network*”; “*Node Clustering in Wireless Sensor*”; “*Clustering network*”; “CAPEX”; “Resiliência”; “*Probability of blocking*”; “*physical topology design*”; “*multiobjective evolutionary algorithms*”; “*cluster head*”; “*Virtual topology design*” e “*clustering Pernambuco*”.

2.1 Trabalhos sobre projetos de redes usando meta-heurísticas

Dupleix, Araújo e Bastos-Filho (2016, p. 1), propõem uma topologia de rede de fibra óptica para o Haiti baseado em um estudo de características sócio econômicas daquele país. O artigo usa o NSGAI1 como meta-heurística adotada para tratar a busca dos objetivos conflitantes de baixa probabilidade de bloqueio e custo. Otimizações multiobjetivos, geralmente, buscam soluções que tendem a curva de Pareto, o artigo usa como métrica um indicador que mede, simultaneamente, a convergência e a diversidade em relação a curva, o *hypervolume*. A problematização da pesquisa foi modelada tendo como entrada a posição dos nós, a matriz de tráfego, e o algoritmo de associação de comprimento de onda. Buscou-se determinar a topologia física e a especificação dos dispositivos ópticos que devem ser implantadas. Os resultados do trabalho apontam que o uso de heurística proporciona um valor de hipervolume melhor após 50 000 avaliações de fitness, e que, mesmo após tantas iterações algumas soluções geradas aleatoriamente no início, ainda estavam presentes no final quando era feito o uso da heurística.

Chaves, Bastos-Filho e Martins-Filho (2010, p. 1), adotaram o NSGAI1 em busca

de baixa probabilidade de bloqueio e baixo custo na construção de topologias de rede de modo geral. O método de avaliação do artigo inclui o uso de um *software* de simulação de rede para calcular a probabilidade de bloqueio e usa um modelo de custo baseado em quatro custos, são eles, custo da fibra óptica e da sua implantação, custo com amplificadores ópticos, custo com comutadores ópticos (*optical cross connect - OXC*), o número de comprimentos de onda na rede e o fator de isolamento do *switch*. Os resultados do trabalho apontam que em uma busca multiobjetivos a resposta para as soluções não será única e sim um conjunto delas, chamadas de soluções ótimas de Pareto. O estudo conclui que sem um indicador externo nenhuma dessas soluções podem ser apontadas como melhor que a outra. O artigo apresenta gráficos com diversos pontos onde cada um é uma solução e comenta que, conforme o gráfico, ao se melhorar a probabilidade de bloqueio há uma penalidade em relação ao CAPEX, e, observando este comportamento um projetista pode escolher o que melhor se adapta a sua necessidade. O artigo finaliza destacando que a metodologia foi capaz de encontrar topologias de rede para diferentes cenários.

Durán et al. (2011, p. 1), propõem um algoritmo que tenta sistematizar a cognição em construção de topologias de redes virtuais. Em trabalho anterior Duran et al. (2007, p. 96), propôs um algoritmo genético para fornecer a rede e o design da topologia lógica, o GAPDELTA (*Genetic Algorithm to Provision the network and to Design the Logical Topology*). Este algoritmo na construção de uma topologia de rede segue os seguintes passos. Primeiro, o algoritmo aloca os transmissores e receptores requeridos em cada nó da rede, determina o número de comprimentos de onda por *link*, e, determina o projeto virtual, ou seja, determinar quais nós devem ser conectados por caminho de luz, encontra uma rota e um comprimento de onda para cada caminho de luz e roteia o tráfego através desses caminhos. Já em Durán et al. (2011, p. 1), foi apresentado um aprimoramento do GAPDELTA, o IA-GAPDELTA (*Impairment Aware - GAPDELTA*) que é um outro algoritmo multiobjetivo para projetar topologias virtuais que minimizando o congestionamento e o número de transmissores em operação. Durán et al. (2011, p. 1), também apresenta o cognição simples IA-GAPDELTA (*Simple Cognition - IA-GAPDELTA*), ou, SC-IA-GAPDELTA, que usa cognição ao selecionar os melhores indivíduos gerados na primeira iteração (ou seja, a partir do momento em que passam a ser conhecidos) para fazer parte da geração inicial aleatória das iterações seguintes. Os resultados do artigo mostram que ao comparar os dois algoritmos, com a curva de Pareto usada como referência, o SC-IA-GAPDELTA obteve melhor desempenho.

N. Fernández et al. (2012, p. 1), demonstram as vantagens de usar cognição em construção de topologias virtuais. Motivado pela dificuldade para incluir a qualidade dos requisitos de transmissão, o trabalho propõe o SC^t-IA-GAPDELTA que estende o método do SC-IA-GAPDELTA, adicionando uma lista de tabus ao método, no momento de resolver o RWA (*Routing Wavelength Association*). Os resultados da pesquisa

obtem-se comparando os três algoritmos (IA-GAPDELTA, SC - IA-GAPDELTA e o SC^t-IA-GAPDELTA) em simulações. As conclusões do artigo apontam que, tomando como base uma curva de Pareto de referência, os algoritmos cognitivos (SC - IA-GAPDELTA e SC^t-IA-GAPDELTA) são melhores que o não cognitivo IA-GAPDELTA e que a versão mais nova (SC^t-IA-GAPDELTA) é o melhor algoritmo dos três.

Figueiredo et al. (2016, p. 409), propõem um algoritmo baseado em estratégia de enxame chamado MaOPSO e o compara com o algoritmo, já bem conhecido, NSGAIII. A problemática do trabalho é modelada da seguinte forma. Sendo dada como entrada a posição dos nós, a matriz de trânsito e o algoritmo de RWA. O objetivo foi encontrar a topologia física e a especificação dos dispositivos ópticos que devem ser implantados para minimizar simultaneamente, a despesa de capital (CAPEX), a probabilidade global de bloqueio da rede (PB), o consumo de energia e melhorar a robustez da rede. Os resultados do artigo apontam que o algoritmo proposto (MaOPSO) obteve melhores valores da métrica *hypervolume* comparado aos valores obtidos pelo algoritmo NSGAIII.

(ARAÚJO et al., 2011, p. 76), investigam como os operadores devem ser manuseados no NSGAI para construção de topologias de rede totalmente transparente (opNTD). A abordagem do trabalho consiste em observar o desempenho do algoritmo ao haver variações nos operadores de iniciação, mutação, *crossover*, seleção, além do, critério de parada. Seus resultados da pesquisa mostram que o manuseio desses operadores podem oferecer resultados melhores na construção de topologias de rede, em relação a várias métricas adotadas no artigo.

2.2 Trabalhos envolvendo técnicas de *clustering*

Younis, Krunz e Ramasubramanian (2006, p. 20), apresentam diferentes abordagens de *clustering* que foram propostas para redes de sensores sem fio (*Wireless Sensor Networks* - WSNs), o trabalho descreve os benefícios, inclusive em função da longevidade das baterias dos sensores. O artigo explica que quando adota-se técnicas de *clustering*, para apoiar agregação de dados usando um tipo de organização nas WSNs para separar os sensores em pequenos grupos chamados *cluster*. A *clustering* resulta em duas categorias de um sistema de hierarquia, onde, dentro de um *cluster* um sensor está na posição hierárquica mais alta chamada cabeça de *cluster* e os outros ocupam o nível mais baixo.

O artigo frisa que por causa de características únicas das WSNs que são feitos em *Ad Hoc* e são volumosas em número de nós, os nós, geralmente, são inconscientes de suas localizações. Por isso, uma abordagem com protocolos de *clustering* distribuída são preferidos pois dependem apenas de informações de vizinhança dos nós (no entanto, a maioria dos estudos nesta área ainda assumem que a topologia

de rede é conhecida por um controlador). Como o objetivo do artigo é apresentar diferentes abordagens de *clustering*, o artigo mostra várias técnicas de *clustering* distribuídos. Observando características nessas técnicas, a pesquisa mostra a vantagem que as abordagens de *clustering* distribuídas tem em relação as técnicas de *clustering* centralizadas, citando como exemplo, uma sobrecarga de mensagem constante por nó. Entretanto, é mencionada uma desvantagem em função da sobrecarga de processamento nas abordagens distribuídas, mas, que para questão de economia de energia essa desvantagem pode ser desconsiderada se comparada com o ganho proporcionado pela vantagem anteriormente citada.

Maheswari, Meenalochani e Sudha (2016, p. 378), realizam uma série de testes com *cluster* em WSNs alterando o posicionamento tanto da base de observação quanto dos nós. A pesquisa modela seus testes da seguinte forma. Dando como entrada o posicionamento dos nós, e, variando (em um experimento manual) a distância de um dos nós em relação ao outro, e também, em relação a base de observação, o trabalho tenta responder como fica o consumo de energia total da rede. Os resultados do trabalho apontam que para o consumo total da rede ser mínimo, o sensor cabeça de cluster deve estar no centroide do *cluster*.

Bezerra, B. e Macario (2016, p. 223), propõem um estudo sobre um sistema de recomendações para voluntários de ONGs no Brasil, o trabalho compara técnicas de *clustering* em uma base de dados de aproximadamente 400 voluntários em Pernambuco. O problema do artigo foi modelado da seguinte forma. Em uma primeira etapa, dando como entrada o sexo, a localização e a especialização de um voluntário, foi feito o primeiro procedimento de *clustering* agrupando esses indivíduos. Após a realização desta primeira etapa veio a segunda etapa que visou incluir, agora, as ONGs nestes *clusters*, considerando o baricentro de cada *cluster* para o processo de alocação. O trabalho usa o índice de *silhouette* (altos valores desse índice apontam *clusters* mais compactos e separados) (BRUN et al., 2007, p. 807), e o índice de Dunn (taxa entre a menor distância entre dois clusters e o tamanho do diâmetro do maior *cluster*). Os resultados da pesquisa apontaram que quando é definido o valor de K *cluster* entre dois e dez, o algoritmos *Kmeans* se saiu melhor do que o *KMedoid* e que os melhores resultados aparecem quando o K fica próximo de quatro.

2.3 Trabalhos que propõem topologias sem uso de MOAEs.

Em Nascimento et al. (2016), foi proposta uma topologia de rede de fibra óptica para o estado de Pernambuco considerando resiliência e custo. Entretanto, o método descrito é manual, sem o uso de algoritmos de qualquer tipo. A problematização do artigo foi relacionada a selecionar cidades candidatas aos POPs através de classificação baseada em indicadores sociais além de experiências cognitivas para complementar o

alcance da rede ao longo do estado. Os resultados do trabalho mostram três topologias resultantes, nas quais, a pesquisa demonstra que o formato da topologia afeta a resiliência e o custo da rede.

Como dito na introdução deste capítulo, os trabalhos relacionados trazem inspiração em diversas linhas de conhecimento usados neste estudo. A Tabela 1 mostra o tipo de relação dos 10 trabalhos descritos como relacionados, com este trabalho.

Tabela 1 – Tabela de comparação de trabalhos relacionados com este trabalho.

Trabalho relacionado	Relação com este trabalho
(DUPLEIX; ARAÚJO; BASTOS-FILHO, 2016)	Uso de meta-heurística na construção de topologias de rede ópticas
(CHAVES; BASTOS-FILHO; MARTINS-FILHO, 2010)	Uso de meta-heurística na construção de topologias de rede ópticas
(DURÁN et al., 2011)	Proposta de algoritmo heurístico na construção de topologias de rede ópticas
(N. FERNÁNDEZ et al., 2012)	Proposta de algoritmo heurístico na construção de topologias de rede ópticas.
(FIGUEIREDO et al., 2016)	Proposta de meta-heurística e estudo de caso na construção de topologias de rede ópticas. Além de, comparação entre dois algoritmos, o proposto e um da literatura.
(ARAÚJO et al., 2011)	Uso de meta-heurística na construção de topologias de rede ópticas
(YOUNIS; KRUNZ; RAMASUBRAMANIAN, 2006)	Descrição de técnicas de <i>clustering</i> em outros tipos de redes
(MAHESWARI; MEENALOHANI; SUDHA, 2016)	Estudo de posicionamento de nós em <i>clusters</i> de outros tipos de redes
(BEZERRA; B.; MACARIO, 2016)	Aplicação de <i>Clusters</i> com o <i>Kmeans</i> para divisão e alocação de amostras com características em comum, inclusive, envolvendo localização geográfica em outros tipos de redes
(NASCIMENTO et al., 2016)	Proposta de topologia de rede de fibra óptica tendo como estudo de caso o estado de pernambuco.

Fonte: o autor.

Nos artigos examinados, a localização dos nós sempre são tratadas como informações estáticas de entrada e não uma variável de decisão encontrada pelo

algoritmo. A propósito, em projetos de redes de longa distância é comum assumir o posicionamento dos nós terminais como uma variável de entrada previamente definida por critérios políticos e comerciais. Contudo, em algumas situações, como àquelas relacionadas com o planejamento de novas redes do tipo REN (*research education network*), é interessante que o posicionamento dos nós seja definido por aspectos técnicos e objetivos. Portanto, de acordo com a revisão da literatura efetuada, este é o primeiro trabalho que tratou a otimização simultânea da localização dos nós terminais na vizinhança da demanda do tráfego e otimização da malha de fibra óptica.

3 Fundamentação Teórica

Neste capítulo, será descrito com mais detalhes os fundamentos teóricos das técnicas que foram usadas neste trabalho. A leitura deste capítulo irá facilitar o entendimento da proposta e dos resultados apresentados, visto que detalha os conceitos teóricos usados nos capítulos seguintes.

3.1 Fundamentação teórica sobre algoritmos de *clustering*

Nesta seção será fornecida a fundamentação teórica de um método de *clustering*, incluindo um algoritmo e uma métrica de controle de qualidade de *cluster*. Será explicada a teoria dos métodos que foram usados nesta pesquisa, também será feito um esforço extra para explicar a teoria focando no caso prático que foi tratado aqui, *clustering* de cidades.

3.1.1 *Kmeans*

Nesta pesquisa, no estudo de caso, foi realizado a divisão do estado de Pernambuco em *clusters* utilizado o algoritmo *Kmeans*. Esse algoritmo visa obter um *cluster* que minimize o quadrado da distância, entre o ponto que é o centro empírico do *cluster*, e o resto dos pontos que o integram. Aplicando o algoritmo *Kmeans* ao problema modelado nesta pesquisa, observa-se a capacidade de, por exemplo, separar uma região geográfica em sub partes e calcular os centros das mesmas chamados de centroides. No caso de localização geográfica de cidades, o centro do *cluster* é a média da localização geográfica de cada uma das cidades dentro de cada *cluster*.

Modelando o problema de tal forma que o conjunto $Y = \{\vec{Y}_1 + \vec{Y}_2 + \dots + \vec{Y}_n\}$, seja os pontos que representam as cidades no mapa e $\vec{Y}_i = \{\vec{y}_{i,1} + \vec{y}_{i,2} + \dots + \vec{y}_{i,d}\}$ a representação com d dimensões para cada ponto. O objetivo do *Kmeans* é encontrar o conjunto $C = \{\vec{c}_1 + \vec{c}_2 + \dots + \vec{c}_K\}$ que representa o conjunto de K *clusters*, onde K é o número de *cluster*.

Chamando de μ_k o ponto médio, ou centroide, de cada *cluster*, a função objetivo final do *Kmeans* dar-se por:

$$O(C) = \sum_{k=1}^K \sum_{y_i \in c_k} \|y_i - \mu_k\|^2. \quad (3.1)$$

Em que $\|y_i - \mu_k\|^2$ é a diferença entre os pontos pertencentes ao *cluster* e o centroide em formação ou protótipo de centroide.

O protótipo de centroide, pode ser obtido por:

$$\mu_k = \frac{1}{|C_k|} \sum_{\vec{y}_i \in C_k} d(\vec{y}_i, \mu_k). \quad (3.2)$$

Em que μ_k é protótipo de centroide do *cluster*, $|C_k|$ é o número de pontos do *cluster* e $d(\vec{y}_i, \mu_k)$ é a distância de cada ponto y_i até o centroide.

O *Kmeans* pede como configuração a ser dada pelo usuário, o número K de *clusters*, a métrica de distância que será usada, como por exemplo a distância euclidiana, e a localização dos centroides de *clusters* inicial, que podem ser aleatórias (BEZERRA; B.; MACARIO, 2016, p. 223).

Os passos do *Kmeans* estão reproduzidos a seguir (JAIN, 2010, p.651):

- 1) Selecione uma divisão inicial com K *clusters*.
- 2) Gere um nova divisão associando cada ponto do *cluster* ao centroide mais próximo.
- 3) Calcule os novos centroides.
- 4) Repita os passos 2 e 3 até que os elementos do *cluster* estejam estabilizados.

3.1.2 Índice de *Silhouette*

O índice de *silhouette* é uma métrica para aferição de qualidade de *clusters*, altos valores dessa métrica apontam *clusters* mais compactos e bem separados. O índice de *silhouette* se dá da seguinte forma, sendo \vec{Y}_i um ponto no *cluster* C_k e n_k , o número de pontos no *cluster*, temos que:

$$S(Y) = \frac{b(\vec{Y}_i) - a(\vec{Y}_i)}{\max[b(\vec{Y}_i), Y_i]}. \quad (3.3)$$

Em que $a(\vec{Y}_i)$ é a distância média entre \vec{Y}_i e todos os outros pontos em C_k , e $b(\vec{Y}_i)$ é a menor das distâncias médias entre \vec{Y}_i e os pontos nos outros *clusters*.

O índice de *silhouette* varia entre valores de -1 a 1, onde quanto mais perto de -1 significa que \vec{Y}_i esta mais próximo, em média, de outro *cluster* diferente do qual ele pertence, caso o índice de *silhouette* seja mais próximo de 1, então, a média das

distâncias para o *cluster* ao qual ele pertence é significativamente menor Brun et al. (2007, p. 807). A fórmula global do índice de *silhouette* é dada por:

$$S = \frac{1}{K} \sum_{k=1}^K \left[\frac{1}{n_k} \sum_{y \in C_k} S(Y) \right]. \quad (3.4)$$

Em que K é o número de *clusters*, ck cada um dos *clusters*, n_k o número de pontos no *cluster* e $S(Y)$ o índice de *silhouette*.

Segundo Brun et al. (2007, p. 807), nos experimentos de sua pesquisa, foi visto que esta métrica é bastante adaptável ao problema de *clustering*.

3.2 Fundamentação teórica para algoritmos multiobjetivos

Nesta seção será fornecida a fundamentação teórica para busca multiobjetivos, incluindo noções de Algoritmos genéticos. Também será explicado sobre algoritmos mémeticos, e, aproveitando a sequência do que foi necessário em termos de teoria neste trabalho, também, explicaremos sobre a inspiração para um método de busca local. Por fim, discorreremos sobre o funcionamento do NSGAII e o NSGAIII.

3.2.1 Algoritmos Genéticos

Algoritmos genéticos funcionam como um espécie de evolução biológica, o cromossomo, uma espécie de formação de caracteres que representa o indivíduo, vai sendo modificado através de mutação e também de cruzamento com outros indivíduos. Tudo isso em busca de indivíduos mais aptos a satisfazer uma dada função chamada de função de *fitness*.

Os conceitos dos algoritmos genéricos são, (ARAÚJO et al., 2011; LIDEN, 2008):

- Cromossomo – Estrutura que representa o indivíduo.
- População – Coleção de indivíduos em busca de melhoramento.
- Geração – Uma dada população em determinada iteração.
- Iteração – Repetição dos processos de melhoramento.
- Função de *fitness* – Função que avalia o quão o indivíduo é apto a um dado problema.
- Critério de parada – Regra de quando parar as iterações do algoritmo.
- Operadores:

- Seleção dos pais: Operador que seleciona os indivíduos mais aptos a se tornarem os indivíduos que serão cruzados para gerar os indivíduos da próxima geração.
- Cruzamento: Operador que cruza características de dois indivíduos.
- Mutação: Operador que transforma uma dada posição de um cromossomo de acordo com uma dada medida estatística de probabilidade de modificação.
- Seleção de sobreviventes: É a seleção dos indivíduos que irão para a próxima geração.

Além disso ainda há outros conceitos práticos que facilitam a aplicação de um escopo ao problema como tamanho do cromossomo e os valores limites, superior e inferior (*Upper Bound*, *Lower Bound*), que cada posição cromossomo pode assumir.

O cromossomo 000111, por exemplo, é uma estrutura binária, e portanto, tem como valores de limite, inferior e superior, respectivamente 0 e 1, e o tamanho do cromossomo é seis.

O pseudo código de um GA (*Genetic Algorithm*), está no Código 3.1 (LINDEN, 2008).

Código 3.1 – Pseudo código de um GA

```

T:=0 // Inicialize o contador de tempo.
Inicializa _ População P(0)// Inicialize aleatoriamente.
Enquanto não terminar faça:// Condição de termino.
  Avalie População P(t)// Avalie a população neste instante.
  P = Seleciona pais_Pais P(t)// Seleção dos pais.
  P =Recombinação_e_mutação P // Aplique os operadores gen
    éticos.
  Avalie _ População P // Avalie esta nova população.
  P(t+1) = Selecione_sobreviventes P(t), P // Seleção de
    sobreviventes.
  T:=t+1 // Incremente o gerador de tempo.
Fim enquanto

```

Apesar de uma abordagem geral de um GA ser extremamente poderosa, uma abordagem mais focada nas características de um determinado problema tem sempre mais chances de sucesso (LIDEN, 2008).

3.2.2 Algoritmos Meméticos

Dado a descrição de muitos problemas do mundo real, as particularidades as vezes são muitas, e por isso, entregar a resolução de alguns problemas a um GA

com uma abordagem convencional sem nenhuma estratégia, talvez, não seja a melhor abordagem. Segundo Liden (2008), o teorema da inexistência de almoço grátis (*No free lunch*), deixa claro que sem embutir conhecimento específico do problema, fica difícil conseguir obter bons resultados de maneira consistente. Ou seja, caso a modelagem do problema no algoritmo, não siga alguma estratégia, e assim, fique livre dentro da evolução dos algoritmos genéticos de forma convencional, o custo computacional pode ficar elevado pra encontrar indivíduos bem adaptados. Portanto ao invés de deixar que o GA enverede de forma tradicional entre a evolução tradicional, da literatura, para redes, é preferível, usar uma busca local para otimização de cada população, usando objetivos que foquem nas métricas desejadas.

Ainda segundo Liden (2008), a ideia de embutir o máximo de conhecimento sobre o domínio de um problema em um algoritmo, deu origem ao que chamamos de algoritmos meméticos. Os algoritmos meméticos visam reproduzir a evolução cultural ao invés da biológica. A ideia é casar os GAs com operadores de busca local para que apenas a forma otimizada de cada indivíduo interagisse dentro da população. É uma espécie de evolução cultural de acordo com o modelo de Lamarek, que pregava que as características que um indivíduo adquire, ao longo de sua vida, pode ser transmitida para os seus filhos. Com uma certa analogia, nos algoritmos meméticos, cada solução é melhorada por métodos locais, podendo passar suas características melhoradas para os descendentes. É praticamente um consenso, nos dias de hoje, que a definição de algoritmos meméticos é a somatória de GAs com métodos de busca local, como no Código 3.2:

Código 3.2 – Pseudo código de um algoritmo memético

```
1. T:=0
2. Inicializa _ População P(0)
3. Enquanto não terminar faça:
4.     Aplique o operador de otimização local
5.     Avalie População P(t)
6.     P = Selecciona pais _ Pais P(t)
7.     P =Recombinação _ e _ mutação P
8.     Avalie _ População P
9.     P(t+1) = Selecione _ sobreviventes P(t), P
10.    t:=t+1
11.    Se P(t+1) convergiu então
12.        Escolha k sobreviventes
13.        Reinicialize o resto da população
14.    fim do Se
15. Fim enquanto
```

Se for feito uma comparação como o primeiro, o GA tradicional, será visto que a diferença está na linha 4 e nas linhas de 11 a 14, a mudança da linha 4 é justamente a

introdução do algoritmo de otimização local, se for expandido este algoritmo, ele será como no Código 3.3 (LIDEN, 2008) :

Código 3.3 – Expansão do trecho de Código da linha 4 do código 3.2

```

Enquanto não terminar faça:
  X =Aplique o operador de busca local ao indivíduo x.
  Se f(x')for menor ou igual f(x) então
    Interrompa a aplicação do operador
  Fim se
Fim faça

```

Com isso está se introduzindo uma mudança sobre o indivíduo corrente para melhorar a sua avaliação. Ao ponto que se esse desempenho não melhorar mais, o máximo local foi alcançado, e, portanto, pode-se parar o algoritmo.

Um dos métodos comuns de busca local é o algoritmo de *Hill Climbing* da função 3.3:

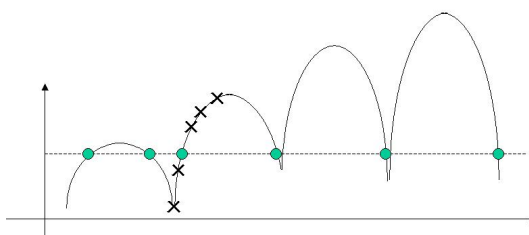
$$X' = x + \alpha \nabla f(x). \quad (3.5)$$

Com $\alpha > 1$ e com $\nabla f(x)$ representando o gradiente da função no ponto representado pelo indivíduo corrente. Tendo como ideia principal que o gradiente aponte para o maior crescimento da função em um ponto.

Entretanto nem sempre é possível calcular o gradiente de uma função, ou ainda, o cálculo pode ser computacionalmente caro. O importante é a percepção de que o método de busca local deve levar a solução para o máximo local da função.

A mudança da linha 11 a 14 dar-se porque a maioria das funções tem poucos máximos locais, o que irá facilmente gerar convergência genética, essa última mudança verifica a convergência genética que deve ser aferida não apenas pela função de *fitness* mas sim pelo genótipo (estrutura do cromossomo). A Figura 5, obtida em (ALGORITMOS GENETICOS, 2017), mostra o porquê não se deve medir a convergência genética apenas pela avaliação de *fitness*:

Figura 5 – Convergência genética.



Fonte: Algoritmos Genéticos (2017).

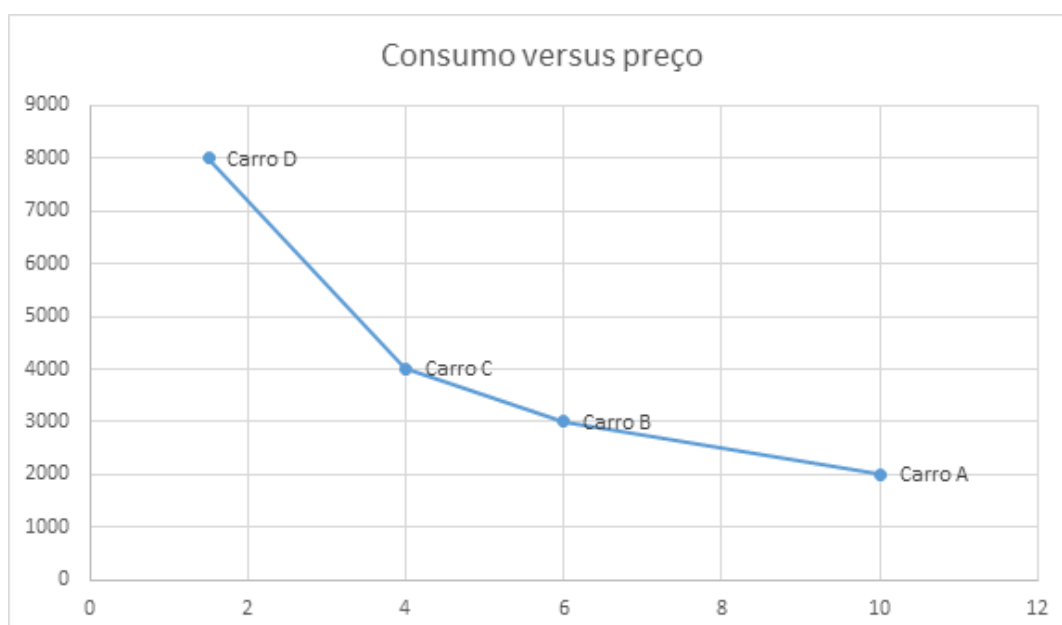
Ao se observar a Figura 5, ver-se que os indivíduos representados por um círculo são bem parecidos em termos de avaliação de *fitness*, entretanto, são bem diversificados geneticamente, já os indivíduos marcados com um X já convergiram geneticamente mas tem avaliação de *fitness* bem diferentes. Em termos de verificação a maneira que primeiro vem à mente quando se trata de comparar os indivíduos é verificar um a um de maneira exaustiva, entretanto isso é computacionalmente muito custoso. Para evitar tal forma de verificação, uma das diversas maneiras de verificar a convergência genética é com o uso do *Kmeans* da seguinte forma, separamos as soluções em grupos e comparamos o centro delas, se a diferença for menor que um certo valor pré-definido, então, houve convergência (LINDEN, 2008).

O método escolhido para busca local foi baseado em descoberta de vizinhança. Uma maneira de descobrir a vizinhança de cidades é calcular a distância da cidade *X*, em questão, para todas as outras cidades do escopo e descobrir as *K* mais próximas. Entretanto isso é computacionalmente custoso e existem algoritmos capazes de reduzir este custo. O *Knn* (*K nearest neighbors*), (FUKUNAGA; NARENDRA, 1975) seria uma opção ideal e a inspiração para a busca local deste trabalho veio dele. Contudo, apesar da implementação completa desse algoritmo ser uma solução adequada para descoberta de vizinhança usada aqui. O CNS já contém algumas funcionalidades que tornam desnecessário a implementação deste recurso por completo neste trabalho, entretanto, em termos de uma reprodução, esta é uma técnica apropriada.

3.2.3 Problemas multiobjetivos

Além das características de um GA e de particularidades da busca local, neste trabalho, também foi preciso se preocupar com a natureza do problema em si. A busca pelas métricas de PB, CAPEX, consumo energético e conectividade algébrica configura uma busca multiobjetivos. Segundo Linden (2008), uma solução para um problema multiobjetivos pode ser descrita como um vetor $\{x_1 \dots x_n\}$ que pertence a um escopo ou espaço de decisão, e vai existir uma função $f(x) \rightarrow y$ que avalie a qualidade do vetor. Nesta seção serão destacadas questões teóricas de multiobjetivos. Diferente da busca por objetivo único, as buscas multiobjetivos não retornam apenas um resultado com o valor máximo ou mínimo (SRINIVAS; DEB, 1994), elas retornam um conjunto de resultados. As buscas por objetivo único tem máximos facilmente identificáveis, quando a decisão tem mais de um objetivo os resultados não são tão simples de indicar qual é o melhor. Principalmente, se os objetivos são conflitantes, o que nos problemas do mundo real acontece com bastante frequência. Se estamos procurando, por exemplo, um carro barato e que consuma pouco combustível a situação pode ficar, como na Figura 6.

Figura 6 – Gráfico multiobjetivos.



Fonte: o autor.

No caso especificado no gráfico da Figura 6 , vemos 4 situações:

- O carro A custa 2000 e consome 10 litros por quilômetros.
- O carro B custa 3000 e consome 6 litros por quilômetros.
- O carro C custa 4000 e consome 4 litros por quilômetros.
- O carro D custa 8000 e consome 1,5 litros por quilômetros.

Veja que uma solução não é melhor e nem pior que a outra, elas são compensatórias, por exemplo, o carro A custa 2000, mas, consome 10 L/km, já o carro D custa bem mais 8000, em compensação, ele consome apenas 1,5 L/km. O que nos leva a necessidade de uma maneira inteligente de selecionar estas soluções de forma eficiente.

3.2.4 Noção de Dominância e Frente de Pareto

Como foi dito antes, tendo em vista que há vários resultados a serem considerados, um algoritmo tem que ser capaz de decidir qual é conjunto desses resultados que são os melhores. Para entender como o algoritmo usado neste trabalho faz isso, é preciso entender a noção de dominância, tratando a função de avaliação como um vetor, em um problema de minimização (LIDEN, 2008):

$$f(x) = \{f_1(x), f_2(x), \dots, f_n(x)\}$$

Uma solução xa domina um solução xb se duas condições forem cumpridas:

Condição 1: $\forall i : f_i(xa) \leq f_i(xb)$

Condição2: $\exists i | f_i(xa) < f_i(xb)$

A Tabela 2 mostra um exemplo onde as soluções com $x = 1$ e com $x = 2$ não são dominadas, entretanto, a solução com $x = 3$ é dominada pela com $x = 2$, pois:

$$f_1(2) = 8 \leq f_1(3) = 8$$

e

$$f_2(2) = 9 < f_2(3) = 13$$

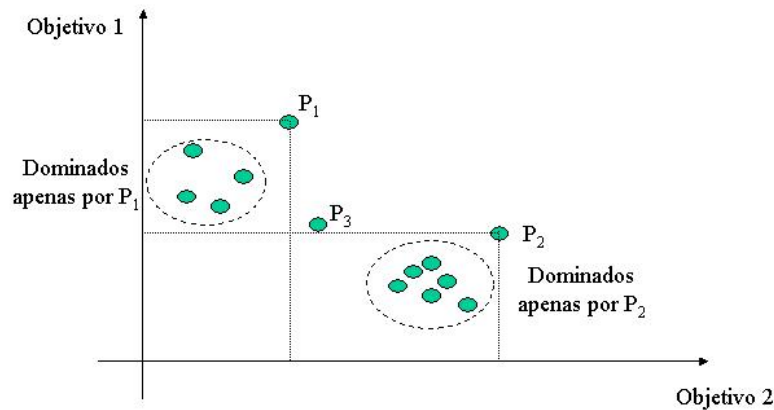
Tabela 2 – Tabela de soluções não dominadas.

x	$f(x)$	$f(x)$
1	9	3
2	8	9
3	8	13

Fonte: o autor.

As soluções que não são dominadas por nenhuma outra são chamadas de não-dominadas. O conjunto de solução que são dominados por um mesmo número de soluções são chamados de conjunto de Pareto (LINDEN, 2008). A Figura 7 mostra um conjunto de Pareto dividido em mais de uma frente, em um problema de maximização, é possível ver dois grupos que são dominados pelo mesma solução, o grupo que é dominada por P1 e o grupo que é dominado por P2, uma vez que um GA trabalho com pontos de população, uma série de soluções de Pareto não-dominadas podem ser capturadas usando GAs (SRINIVAS; DEB, 1994).

Figura 7 – Fronteiras de Pareto.



Fonte: Algoritmos Genéticos (2017).

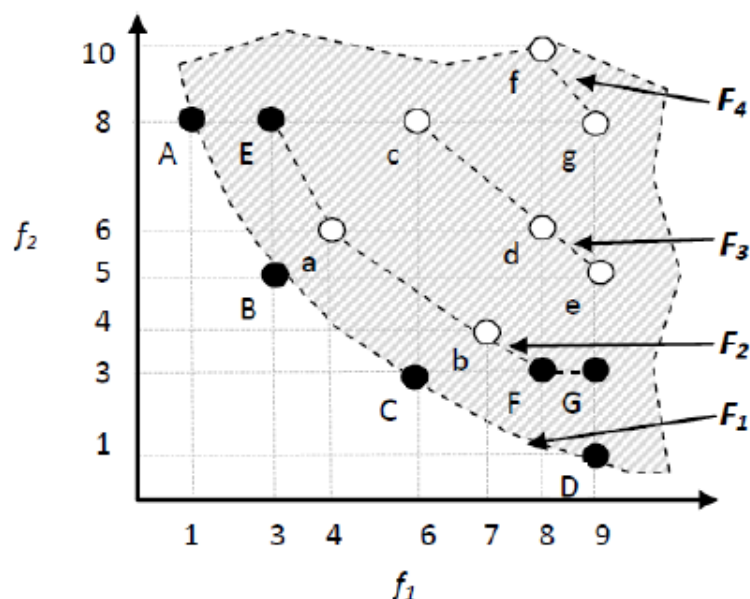
3.2.5 O Framework NSGA

O NSGA (*Non-dominated sorting genetic algorithm*) é um algoritmo genético evolucionário multiobjetivos proposto por (SRINIVAS; DEB, 1994) . As suas evoluções são o NSGAII para dois objetivos e o NSGAIII que foi testado com 3 a 15 objetivos (DEB; JAIN, 2014) . Este *framework* já foi bem testado no estado da arte sendo apontado como uma meta heurística bem adaptada a problemas multiobjetivos (DEB et al., 2002) foi com ele que foi feito a busca multiobjetivos deste trabalho. O funcionamento do NSGA III difere do funcionamento do NSGAII apenas, por uma mudança no operador de seleção e pela manutenção da diversidade que no NSGAIII é auxiliada por um conjunto de pontos de referência. Será mostrado o funcionamento do NSGAII e depois será destacada a diferença.

3.2.6 NSGAII

Supondo que o NSGAII seja programado para formar uma população de indivíduos de tamanho $P=7$, inicialmente a população P é zerada então $P=0$. O NSGAII irá selecionar soluções não dominadas das frentes de Pareto dentro do espaço de busca para adicionar a P ($P = P +$ Soluções não dominadas da frente de Pareto). Em primeiro lugar ele vai classificar as soluções da primeira geração, na primeira frente de Pareto (essa população inicial geralmente é gerada aleatoriamente), após isso, ele adiciona estas soluções a população P . Então ele testa se P atingiu o tamanho $P=7$, se atingiu ele para, se não atingiu ele aplica os operadores de seleção dos pais, mutação e *crossover*, e cria uma nova geração. Desta forma, após criar a nova geração ele a classifica em outra frente de Pareto e repete o procedimento. Ele faz esse procedimento até atingir o tamanho $P=7$. A Figura 8 mostra as populações classificadas em frentes de Pareto, em um problema de minimização, as soluções claras são das gerações anterior, e as soluções pintadas de preto são das novas gerações (ARAÚJO, 2015).

Figura 8 – Classificação de soluções não dominadas utilizada pelo NSGAI.



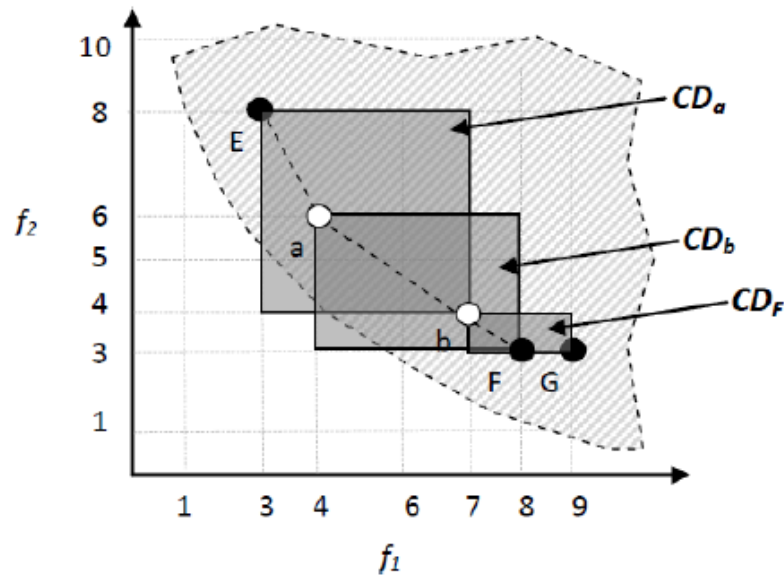
Fonte: Araújo (2015).

Entretanto se na última frente de Pareto o tamanho de P mais o número de soluções não dominadas for maior que o tamanho que foi programada a população, $P=7$:

$$(P + \text{Solues no dominadas da ltima frente de Pareto} > 7)$$

Então ele aplica uma métrica de manutenção de diversidade chamado CD - distância da multidão (*Crowding Distance*). Se houver um raciocínio, em termos de projetos do mundo real, será visto que é mais interessante manter a diversidade das soluções do que entregar a um projetistas soluções muito parecidas e deixá-lo com opções de escolha que quase não farão diferença para ele. O CD é uma métrica que se baseia na medida de um Cubóide formado pelas soluções adjacentes a uma dada solução. Para aplicação do CD, primeiramente, o algoritmo considera as soluções da extremidade da frente de Pareto e depois aplica o CD a partir delas. As soluções que tiverem a maior medida de CD estão em uma região pouco densa de soluções e serão elas a serem selecionada a fim de preservar a diversidade. A Figura 9 mostra a aplicação do CD a última frente de Pareto.

Figura 9 – Avaliação CD das soluções que estão na mesma frente de Pareto.



Fonte: Araújo (2015).

O Código 3.4 mostra o algoritmo do NSGAI.

Código 3.4 – NSGAI - adaptado de Araújo (2015)

```

Crie uma população inicial POP
para i=0; i < Imax; i ++ faça
  para s=0; s < P; s=s + 2 faça
    Selecione dois pais, P1 e P2
    Execute cruzamento entre P1; P2 e gere os descendentes
      D1 e D2
    Faça mutação com D1 e D2
    Avalie D1 e D2 e inclua no conjunto de descendentes
      DESC
  Fim para
Crie um conjunto UNIAO a partir de POP e DESC
Separe as frentes de Pareto usando o conceito de dominância
Crie um conjunto vazio POP
enquanto o tamanho de POP < P faça
  Ordene as solucoes da frente de Pareto por CD
  para cada solucao no conjunto ordenado faça
    se comprimento de POP < P então
      Inclua a solução no conjunto POP
  Fim se
  Fim para
  Analise a próxima frente de Pareto
Fim enquanto
Fim para
Fim procedimento
  
```

3.2.7 NSGAIII

O algoritmo do NSGAIII é semelhante ao do NSGAII com uma significativa diferença no operador de seleção (DEB; JAIN, 2014) . Além disso, a diferença do NSGAIII para o NSGAII está na maneira de preservar a diversidade, o NSGAIII não usa a técnica de *Crowding Distance* (CD), ele usa uma técnica de espalhamento baseada em um conjunto predefinido de pontos de referência em um hiperplano normalizado que é igualmente inclinado para todos os eixos de objetivos(DEB; JAIN, 2014), onde M é o número de objetivos do problema.

Se p divisões são consideradas para cada objetivo, a quantidade de pontos de referência H será dada pelo arranjo simples:

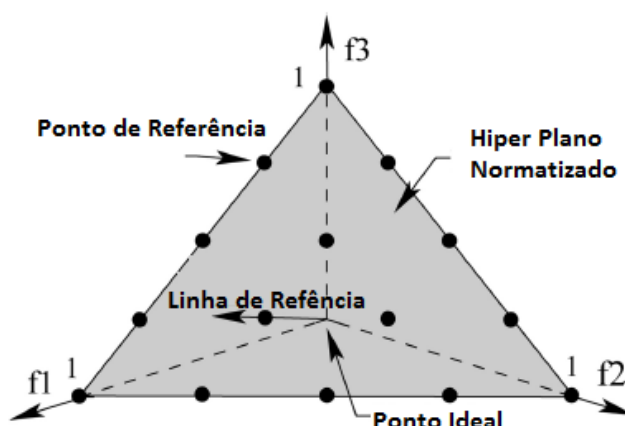
$$H = \binom{M + p - 1}{p}. \tag{3.6}$$

Por exemplo, para três objetivos $M=3$, e com quatro divisões por eixo de objetivos $p=4$, os pontos de referência serão criados em um triângulo com o ápice em (100), (010), (001). Aplicando a fórmula de combinação simples $C_{n,p} = \frac{n!}{p!(n-p)}$ a quantidade de pontos de referência H será:

$$H = \binom{3+4-1}{4} = \frac{6!}{4!(6-4)!} = \frac{5 \cdot 6}{2!} = 15.$$

Com a quantidade de pontos de referência já calculada, o total de pontos é espalhado pelo hiperplano e as soluções são associadas a estes pontos garantindo assim a diversidade já que os pontos são bem espalhados, a Figura 10 mostra o hiperplano normalizado.

Figura 10 – Quinze pontos de referência estruturados são mostrados em um hiper plano normalizado.

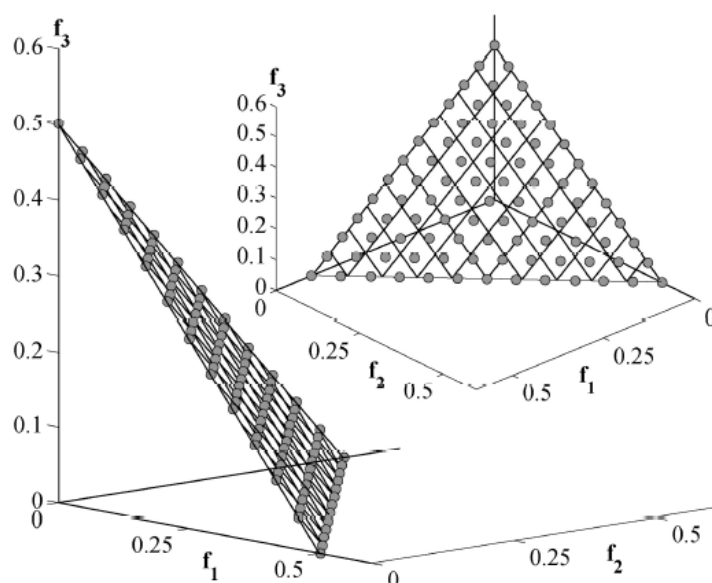


Fonte: adaptado de (DEB; JAIN, 2014).

A Figura 11 mostra como ficou bem espalhada a distribuição do problema DTLZ1

(benchmark conhecido da literatura) usando o NSGAIII.

Figura 11 – Solução do problema DTLZ1 obtida pelo NSGAIII.



Fonte: adaptado de (DEB; JAIN, 2015).

Desta forma, como os indivíduos associados aos pontos de referência ficam bem espalhados por todo o hiperplano, há uma preservação da diversidade das soluções.

3.3 Métricas para Redes e Qualidade

Nesta seção, serão aprofundadas as métricas que foram usadas para avaliação do conjunto de redes encontradas pelo algoritmo de busca multiobjetivo, a saber: a probabilidade de bloqueio (PB), Consumo energético, custo de implantação (CAPEX) e conectividade algébrica.

3.3.1 Probabilidade de Bloqueio (PB)

A probabilidade de bloqueio é a medida estatística da rede não conseguir responder com sucesso a uma solicitação de tráfego por falta de recursos (ARAÚJO, 2015). Sistemas de transmissão tem um limite físico de capacidade de resposta a uma solicitação de tráfego. Como exemplo, pode-se supor a situação em que, em um dado momento, um sistema de transmissão qualquer tenha uma quantidade de canais de transmissão igual a X . Neste mesmo momento a quantidade de solicitação de tráfego seja igual a Y , com $X < Y$ (a quantidade de canais disponíveis seja menor que a solicitação de tráfego). Em uma situação dessas, faltará recursos da rede para atender a solicitação de tráfego e uma situação de bloqueio irá acontecer.

PB é uma medida de desempenho da rede, quanto maior a probabilidade de bloqueio menos usuários conseguirão ser atendidos, sendo portanto essa, uma rede com desempenho baixo. Uma das maneiras de aferir a probabilidade de bloqueio de uma rede é simular um alto número de requisições de tráfego e medir quantas delas tiveram sucesso e quantas não, dessa forma a probabilidade de bloqueio PB é dada por (ARAÚJO, 2015):

$$PB = \frac{sT_{bloqueiadas}}{sT_{total}}. \quad (3.7)$$

Em que $sT_{bloqueadas}$ são as solicitações de tráfego bloqueadas e sT_{total} são as solicitações de tráfego total.

3.3.2 Custo de implantação CAPEX

O modelo de custo neste trabalho foi o mesmo adotado em (ARAÚJO, 2015), usando uma unidade genérica de custo (u.m.), foram considerado quatro custos:

O custo fixo com comprimento de onda para toda a rede, o custo com implantação de fibra óptica, o custo com amplificadores de sinal óptico, e o custo com ROADMs (*Reconfigurable optical add-drop multiplexers*).

O comprimento de onda tem um custo por está associado a OTL (*Optical Line Terminal*) um equipamento de conversão de sinal óptico elétrico que trabalha entre os clientes e a rede translúcida (totalmente óptica). O custo é dado por (ARAÚJO, 2015):

$$COST_{\lambda} = 2 \cdot \eta \cdot w \cdot \sum_{i=1}^n D(i, i). \quad (3.8)$$

Em que:

- η é uma constante em função do custo da OTL.
- w é número de comprimentos de onda em cada enlace.
- n é a quantidade de nós da rede
- $D(i, i)$ é o grado do i -ésimo nó

O custo com a fibra óptica pode ser enxergado como a soma dos custos do cabo em si com os custos de implantação dos mesmos (ARAÚJO, 2015):

$$COST_{cable} = (\beta_{dcf} + \beta_{ssm} + \beta_{dcf} + \beta_{Dep}) \sum_{i=1}^n \sum_{j=i+1}^n d(j, 1). \quad (3.9)$$

Em que:

- β_{dcf} custo com fibras DCF (*Dispersion Compensate Fiber*).
- β_{ssm} custo com fibras SSMF (*Standard Single Mode Fiber*).
- β_{Dep} custo com a implantação das fibras.
- i, j são nós e $d(i, j)$ é a distância entre eles.

O custo com os amplificadores é dado por (ARAÚJO, 2015):

$$Cost_{amplifier} = \delta \cdot \sum_{i=1}^{e_{max}} Camp(i). \quad (3.10)$$

Em que:

- δ é uma constante de custo.
- e_{max} é o número máximo de *links* bidirecionais.
- $Camp(i)$ é o custo unitário de um modelo de amplificador.

O custo com o equipamento de *cross* conexão ou comutador ópticos é dado por (ARAÚJO, 2015):

$$COST_{ROADM} = C_{ws}(v_{k+}) \cdot \sum_{i=1}^n (\gamma_1 \cdot w + \gamma_2 \cdot D(i, i) + \gamma_3). \quad (3.11)$$

Em que:

C_{ws} esta relacionado ao fator de isolamento do comutador e $\gamma_{1,2,3}$ são constantes para adaptar o modelo as condições da mercado.

A soma desses quatro custos é a igual ao custo de implantação da rede.

3.3.3 Consumo energético em watts

Cada equipamento da rede tem um consumo energético associando, comutadores, amplificadores, equipamento de transmissão de dados da camada *ethernet*, entre outros. Geralmente, o consumo é fornecido em quilowatt (kw), o consumo energético em uma dada rede qualquer é a somatória do consumo de todos os seus equipamento. O modelo de consumo usado pelo CNS é o proposto em (HEDDEGHEM et al., 2012, p. 86), e é descrito como a soma dos consumos em equipamentos que lidam com:

- Comutação de protocolos de *Internet* e multi protocolos (*Internet protocol/multi-protocol label switching* - IP/MPLS).
 - Uma camada IP / MPLS com roteadores associados.
- Rede de transporte óptico (*optical transport networking* - OTN).
 - Uma camada OTN, que executa a multiplexação e transmissão de divisão de tempo.
- Multiplexação por divisão de comprimento de onda (*wavelength division multiplexing* - WDM).
 - Uma camada WDM que executa a multiplexação e transmissão de divisão espacial.
- Camada de rede *ethernet* (*ethernet layer*).
 - A camada de transporte.

O consumo total é dado por:

$$P_{core} = P_{ip} + P_{ethernet} + P_{ont} + P_{wdm}. \quad (3.12)$$

Em que:

- P_{core} é o consumo total.
 - P_{ip} é o consumo com protocolos e multi protocolos de *internet*.
 - $P_{ethernet}$ é o consumo com equipamentos da camada de transporte.
 - P_{ont} é o consumo com os equipamentos de multiplexação em divisão de tempo.
 - P_{wdm} é o consumo com os equipamentos de multiplexação em divisão espacial.
- 3.3.4 Conectividade Algébrica

A conectividade algébrica é uma métrica que emite informações sobre a robustez de uma rede, para detalhá-la é preciso adentrar no universo teórico dos grafos. Matrizes são usadas para representar informações sobre um grafo. Uma matriz de adjacências A de um grafo tem seus nós conectados, se $a_{ij} = 1$, e caso seja igual a zero, tem seu nós desconectados.

A matriz D é uma matriz diagonal $n \times n$ que contém o grau dos nós (número de enlaces que liga o nó a seus vizinhos) em sua diagonal principal. Dessas duas matrizes podemos extrair a matriz laplaciana L da seguinte forma (ARAÚJO; BASTOS-FILHO; MARTINS FILHO, 2014):

$$L = D - A. \quad (3.13)$$

O segundo menor autovalor da matriz laplaciana é denominado conectividade algébrica $(\gamma_n - 1)$, altos valores de conectividade algébrica revelam um maior nível de dificuldade de se dividir um grafo em dois componentes se for removida uma aresta. Entretanto se $(\gamma_n - 1) = 0$, então o grafo é desconectado. Resumindo, a conectividade algébrica é uma medida de robustez para rede. Altos valores de conectividade algébrica melhoram a capacidade de uma rede de se recuperar de uma falha.

3.3.5 Erlang

Erlang é uma medida para volume de tráfego, um erlang representa a relação entre o tempo entre chegada de novas demandas à rede óptica e o tempo em que uma chamada permanece ativa. Ou seja, quanto mais tempo uma chamada fica ativa, maior é a carga e quanto menor é o intervalo entre as chamadas, maior é a carga. Por isso um erlang é proporcional à um e inversamente proporcional ao outro. Em um tráfego entre dois nós, a fórmula de erlang é dada por (ARAÚJO, 2015):

$$L = \frac{\text{volume de tráfego}}{\text{período de observação}}. \quad (3.14)$$

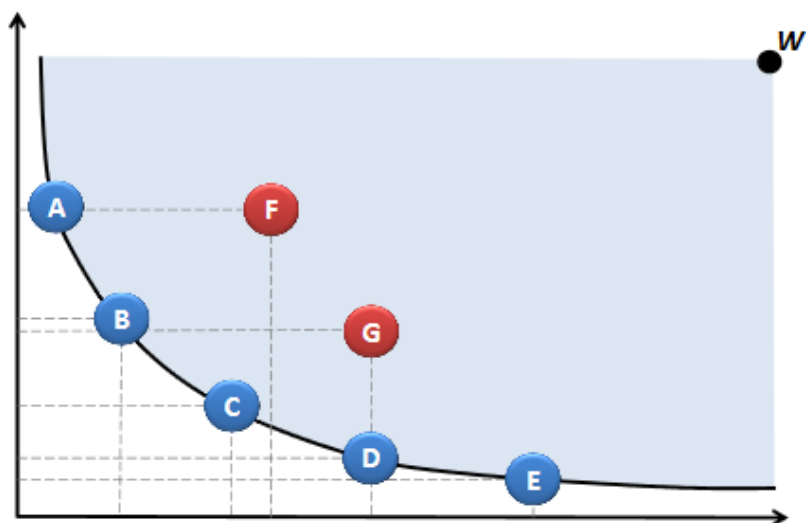
Em que L é a carga do tráfego em erlang.

Neste trabalho, que foi usado o simulador do CNS, o tráfego é modelado no simulador como um processo de Poisson (um tipo de modelo estatístico de aleatoriedade).

3.3.6 Hypervolume

O *hypervolume* proposto por (ZITZLER et al., 2003, p. 117), é uma métrica usada para medir a área destacada do gráfico genérico da Figura 12, note que a área é entre um ponto W e uma curva de Pareto. O ponto W é usado como referência para medir a curva (ARAÚJO, 2015).

Figura 12 – Área do *hypervolume*.



Fonte: adaptado de ARAÚJO(2015).

Altos valores de *hypervolume* indicam uma maior convergência e diversidade à curva de Pareto.

4 Proposta

Como foi descrito nos capítulos anteriores, este projeto usou técnicas de *clustering* e de busca multiobjetivos na construção de topologias de redes ópticas de longa distância e metropolitana. As duas técnicas de inteligência computacional foram usadas de maneira integrada, ou seja, a saída de uma foi a entrada da outra, entretanto, cada uma teve sua atuação em finalidades separadas e bem definidas.

O algoritmo de *clustering* teve dois objetivos de escopos bem definidos. O primeiro foi como uma forma de subdividir uma região geográfica em grupos, neste estudo de caso, o estado de Pernambuco foi dividido em grupos de cidades. Então, com essa divisão feita, propor uma nova abordagem de mitigação de cobertura de rede, ou seja, de forma automática garantir que a rede realmente cubra a região a qual se propõe a atender. O segundo objetivo do algoritmo de *clustering* foi mitigar o custo da rede ao mesclar tecnologias de redes mais caras (WDM, DWDM) e de redes mais baratas (PON, CWDM), dependendo do trecho de rede, inter *cluster* (WDM, DWDM) e intra *cluster* (PON, CWDM).

Se há uma região dividida em partes, a interligação via *link* de fibra óptica entre as partes e a interligação interna, também via *link* de fibra óptica, de cada parte, viram escopos separados e bem definidos onde cada um pode dispor do que mais se adéque em termos de custo e necessidade de demanda de tráfego. A entrada do algoritmo de *clustering* é a localização geográfica dos pontos candidatos a nós terminais. Já a saída são os pontos separados em *clusters* e tendo o centroide de cada *cluster* compondo a lista de cabeças de *cluster* inicial, ou seja, a posição inicial dos nós terminais é uma variável de decisão do algoritmo de *clustering*.

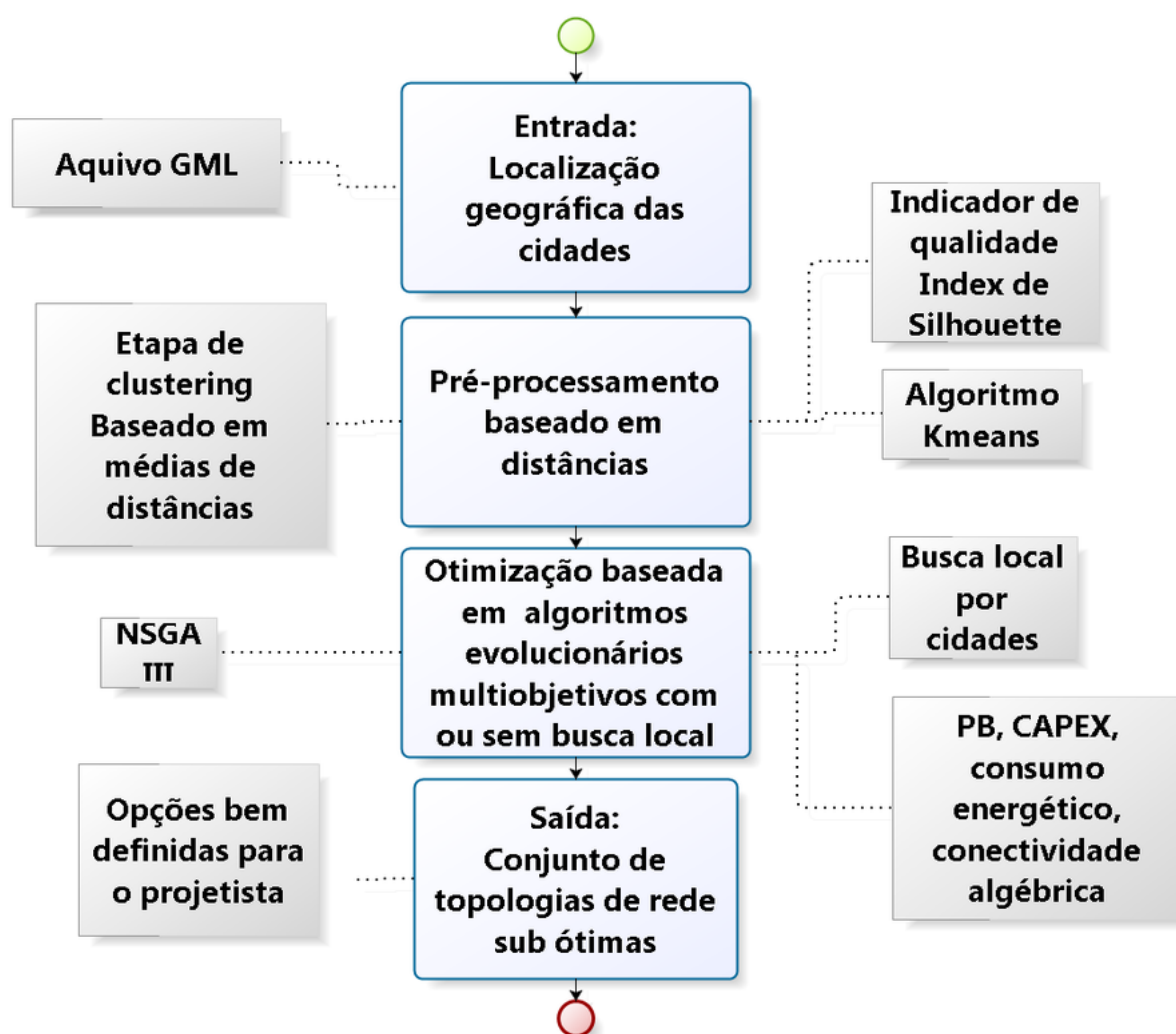
O algoritmo de Busca Multiobjetivos foi utilizado para outras finalidades, também com escopos bem definidos, a saber, foram duas, ambas inspiradas no método, de composição de topologias de redes, da literatura (DUPLÉIX; ARAÚJO; BASTOS-FILHO, 2016, p. 1), (FIGUEIREDO et al., 2016, p. 409), (CHAVES; BASTOS-FILHO; MARTINS-FILHO, 2010, p. 1), entretanto, de forma inovadora. A primeira finalidade foi na composição de topologias de rede da mesma forma que é feito na literatura, como será explicado na seção 4.2.1, mas, com a localização dos nós terminais não determinada de maneira manual, como é feito na literatura, e sim, recebida de um outro algoritmo, o *Kmeans*, que só define essa localização após a etapa de pré-processamento baseado em distâncias. A segunda finalidade, se difere da primeira apenas pela inclusão da busca local, que adiciona mais indefinição ainda a questão da localização dos nós terminais, já que, a busca local nada mais é que, a variação da localização dos nós terminais, definida inicialmente pelo *Kmeans*, com o objetivo de

estimar redes mais adaptadas.

Observe que o formato de como foi utilizado o algoritmo de busca multiobjetivos, integrado com o algoritmo de *clustering*, torna os dois modelos desenvolvidos neste trabalho, com e sem busca local, modelos inovadores. Vale também salientar que o método é usado com busca ou sem busca de forma excludente por execução, e que, no capítulo de resultados será apresentado várias comparações entre esses dois modelos.

A proposta desta monografia para o planejamento integrado de redes ópticas de longa distância e metropolitanas pode ser resumido pela Figura 13.

Figura 13 – Fluxo principal do projeto.



Fonte: o autor.

Na Figura 13, pode-se perceber que o fluxo de execução recebe como entrada um arquivo GML contendo a localização geográfica das cidades. Em seguida é feito um pré-processamento de *clustering*, baseado em distâncias, utilizando o algoritmo *Kmeans*. Após a etapa de *clustering* começa a etapa de otimização multiobjetivos, com ou sem busca local (a depender de como foi configurado), o NSGAIII busca pela

topologia de rede em si, e quando é configurado pra usar busca local, também busca por cidades para serem cabeças de *cluster*. A resposta do método é o conjunto de topologias de rede subótimas.

4.1 Pré-processamento baseado em *Clustering* por média de distâncias

No estudo de caso dessa pesquisa, foi considerado o estado de Pernambuco. O estado de Pernambuco tem 185 municípios (INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA, 2010), e para atendê-lo com uma rede de fibra óptica é necessário distribuir a rede plenamente para o estado todo, considerando entre outras coisas o CAPEX.

As rede de fibra óptica do tipo PON são para distâncias menores e tem o custo com fibras e equipamentos bem menores em relação aos das redes do tipo WDM ou DWDM (TRAN; CHAE; TUCKER, 2005, p. 1). Em contrapartida as redes do tipo WDM ou DWDM são mais indicadas para longa distância e para transporte de um grande volume de dados (ARAÚJO; BASTOS-FILHO; MARTINS FILHO, 2014, p. 11). Considerando estas características uma maneira eficiente de distribuir uma rede de fibra óptica para uma região qualquer, como um estado, é dividir essa região em sub-regiões como uma espécie de *cluster*. Nessa configuração as redes do tipo *backbone* fariam a ligação entre os *clusters* e as redes do tipo PON ligariam o *cluster* internamente quando possível, ou seja, quando a distância entre as cidades internas ao *cluster* não excedessem as características máximas de distância de uma rede PON. Desta forma, pode-se garantir o atendimento plenamente a toda a completude de uma região, observando-se apenas que todos os *clusters* estejam sendo atendidos. Ao ponto que há, nessa estratégia, uma diminuição nos custos por haver uma mescla entre tecnologias de custos altos DWDM e custos menores CWDM, GPON (Gigabyte PON).

4.1.1 Planejamento da aplicação do *Kmeans*

Para dividir o estado de Pernambuco em sub partes ou grupos de cidades que comporiam os *clusters*, inicialmente se idealizou algo em torno de 10% do total de cidades para o número de *clusters*, ou seja, dezoito clusters. Entretanto, isso é só uma estimativa, a intenção desse número pequeno, como estimativa inicial, era o controle com o CAPEX. Pode-se tomar como pior caso, em termos de CAPEX, a situação onde cada cidade do estado compõe, sozinha, um *cluster*. Isso implicaria que toda cidade teria um POP e que toda a rede seria ativa, o que causaria um elevado custo. Por isso em torno de 10% do total do número de cidades para o número de clusters auxilia a um controle inicial em termos de custo, visto que, haverá apenas em torno de 10% de

idades com POPS. Entretanto, para que as sub-regiões compusessem *clusters* mais compactos e mais bem separados teria que se testar um leque com vários números de sub-regiões e não apenas fixa-las em 10% do total de cidades. Buscando, desta forma, um número de *clusters* que fosse compensatório, em termos de, custo da rede e as características de uma subdivisão da região bem produzida. A métrica escolhida para esta busca de *clusters* mais compactos e bem separados foi o índice de *silhouette* (BRUN et al., 2007, p. 807).

Com as informações de cidades levantadas, em especial, o nome e a localização geográfica (latitude e longitude, em valores decimais), foi montado um arquivo GML com 184 cidades do estado de Pernambuco. Isso porque excluiu-se Fernando de Noronha, a qual por suas características geográficas, que está dentro do mar, por isso tem que ser atendida por cabo marinho ou outra tecnologia que se adéque, as quais não entram no escopo deste trabalho. Esse arquivo GML serviu como base de dados para o *Kmeans*. A resposta do algoritmo foram os *clusters* e os seus devidos centroides. Essas saídas do *Kmeans*, serviram de entrada para o algoritmo multiobjetivos na etapa de busca multiobjetivos.

4.2 Otimização baseada em algoritmos evolucionários multiobjetivos com busca local

Com a etapa de *clustering* do estado concluída, foi possível iniciar a busca multiobjetivos para a rede, a intenção é que cada cluster tenha um POP. A variação da cidade detentora do POP em cada cluster, associada com a forma com que os *links* de fibra óptica as conectam, é que irá gerar as mais variadas configurações de topologias de rede. As quais foram medidas em termos de Probabilidade de Bloqueio PB, CAPEX, consumo energético e Conectividade algébrica. Como já mencionado neste trabalho, a busca por estas métricas em uma rede é uma busca multiobjetivos e pode ser feita por meio de algoritmos genéticos. Foi escolhido do estado da arte o algoritmo NSGAIII (*Nondominated sorting genetic algorithm III*) (DEB; JAIN, 2014, p. 577).

4.2.1 Planejamento da aplicação do NSGAIII

Na etapa de *clustering* foi demonstrado que um número de *K cluster*, para o estado de Pernambuco, está idealmente, entre 15 e 25 *clusters*. Mas para fins de explicação, vamos assumir que sejam quatro *clusters*, os desafios então foram dois, o primeiro foi definir de que forma estes *clusters* iriam se interconectar através dos *links* de fibra óptica. O segundo desafio foi escolher qual cidade de cada *cluster* iria assumir a posição de detentora do POP. Na prática esses desafios são complementares já que a variação na escolha da cidade para ter o POP, também, influência diretamente na forma em que os *clusters* se conectam. E pode-se afirmar que uma dada forma

de topologia pode influenciar na escolha de uma cidade para ter um POP por deixar a rede com melhores valores de métricas. Para se alcançar o primeiro desafio uma abordagem prática, neste caso, foi fazer uma matriz de conexão, sendo cada cidade representada por um número chamado *ID*, as cidades do nosso estudo de caso foram representadas pelos *ID* de 0 a 183. Neste caso supõe-se que dos quatro *clusters* do exemplo, as cidades escolhidas para serem POP foram as de *ID* 18, 67, 123, 160 isso implica que a matriz de conexão se dará como na Figura 14.

Figura 14 – Matriz de conexão com seleção de seis interconexões.

	18	67	123	160
18	X	1	0	1
67	1	X	1	1
123	0	1	X	0
160	1	1	0	X

Fonte: o autor.

A Figura 14 representa uma matriz de conexão, onde as cidades de *ID* 18, 67, 123 e 160 estão escritas, respectivamente, nas linhas e colunas. A matriz representa a conexão ou não entre estas quatro cidades. Observe que a diagonal principal da matriz (preenchida com x) representa a interligação de cada cidade consigo mesma e por isso, não faz sentido usá-la. A matriz é espelhada, visto que a conexão é bidirecional, ou seja, um *link* só pra transmissão e recepção (um par de fibras), então, só precisa-se usar um lado da matriz (FIGUEIREDO et al., 2016, p. 409). As informações contidas nas células, zero ou um, indicam se as cidades são conectadas (um) ou não (zero).

Portanto para representar as interligações entre N cidades de diferentes *clusters* esta pesquisa precisou, apenas de uma parte da matriz de conexão do tamanho de $N \times N$. Onde a célula da matriz trouxe a informação de conexão ou não, e, os cabeçalhos de linhas e colunas (de agora em diante chamados de rótulos) informou quais foram as cidades em questão. Se for observado a lógica do problema, será visto que, qualquer uma das N cidades poderá se interconectar com qualquer uma das outras, o que gera uma possibilidade de $n \times n$ interligações. Mas como não faz sentido uma cidade se ligar com ela mesma, então, existe uma possibilidade de interconexão a menos, ou seja, $n(n - 1)$ interligações. Como a ligação é bidirecional, só a metade da matriz precisará ser usada, então: a quantidade de interligações será igual a $\frac{n(n-1)}{2}$ (FIGUEIREDO et al., 2016, p. 409).

Aplicando a expressão $\frac{n(n-1)}{2}$ ao exemplo de quatro *cluster*, temos 6 interligações possíveis:

$$\frac{4(4-1)}{2} = \frac{4(3)}{2} = \frac{12}{2} = 6 \text{ interligações.}$$

O que de fato é verdade, pois, pode-se verificar isso contando as células que representam as conexões na Figura 14.

A partir da Matriz apresentada na Figura 14, é que também foi retirado o cromossomo para o GA, observe que se for concatenado linha a linha da matriz (linha 1: 101; linha 2: 11; linha 3: 0), a partir da metade superior, se terá o cromossomo 101110. Deste cromossomo considerando 0 como não conectado e 1 como conectado, pode-se tirar a seguinte informação:

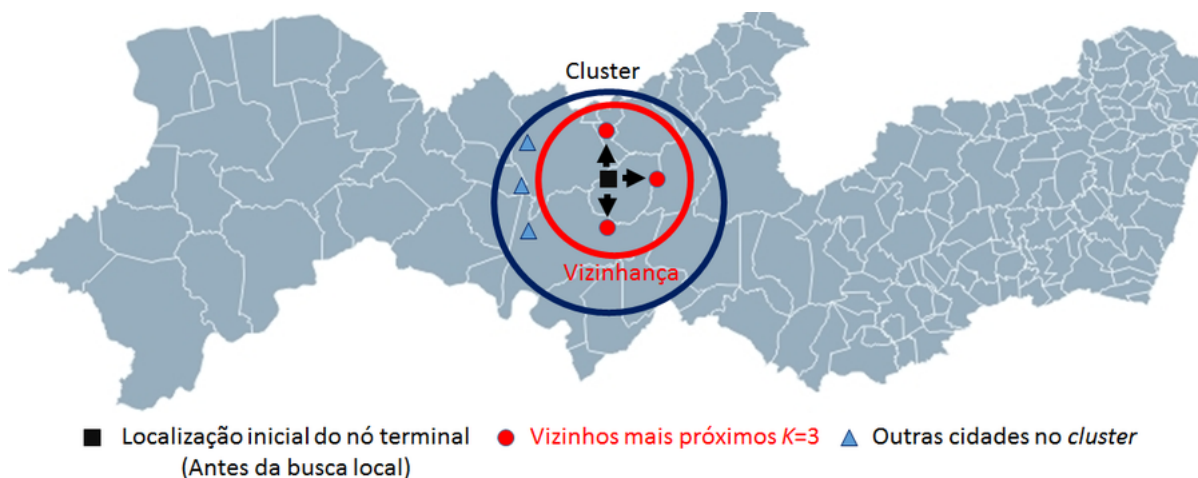
- A primeira cidade é conectada a segunda e com a quarta, mas não é com a terceira (linha 1 da matriz).
- A segunda cidade é conectada com a terceira e a quarta (linha 2 da matriz).
- A terceira cidade não é conectada com a quarta (linha 3 da Matriz).

A concatenação das linhas que gera o cromossomo 101110 é facilmente manipulável por um GA.

O segundo desafio era escolher quais cidades iriam ser selecionadas para receberem um POP, ou seja, trazendo para realidade da abordagem, seria escolher quais cidades iriam ser selecionadas para assumirem as posições das linhas e colunas da matriz de conexão. Resumindo, seriam as cidades que ocupariam os rótulos da matriz, isso partindo da configuração inicial entregue pelo *Kmeans*, onde, os centroides de cada *cluster* é que são os rótulos. Esse aspecto também foi modelado no GA usando métodos de busca local.

Após a decisão de uso de busca local para especialização do GA, este trabalho focou em como seria feito para variar as cidades rótulos das linhas e colunas da matriz de conexão. Seria interessante um operador que ao invés de tentar calcular o gradiente o que, como já foi dito, pode ser muito custoso, agisse com pequenas perturbações (LIDEN, 2008). Observando a natureza do problema foi verificado que as cidades de cada *cluster* devem ser, apenas, substituídas por outras cidades do mesmo *cluster*, dessa forma, seria possível manter a cardinalidade de uma cidade com POP para cada *cluster*. Uma maneira de fazer isso com pequenas perturbações, no método de busca local, é variando as cidades, apenas, entre as cidades vizinhas, o que leva a necessidade de descoberta de vizinhança. Descobrir a vizinhança nada mais é do que achar as K cidade mais próximas à uma cidades X qualquer em questão. A Figura 15, adaptado de (EMBRAPA, 2017), mostra um exemplo deste aspecto:

Figura 15 – Descoberta de vizinhança: variação das cidade com busca local.



Fonte: adaptada de Embrapa (2017).

Como a variação de cidades tem que ser limitada pelas cidades dentro de cada *cluster*, isso já limita a quantidade de cidades para a busca, pois, na etapa de *clustering* o CNS, utilizando o *Kmeans*, já dividiu o estado em grupos de cidades. Além do mais o CNS, em um atributo de uma de suas classes, guarda uma matriz de distâncias. Possibilitando com isso que não seja necessário refazer os cálculo de distâncias entre cidades, bastando apenas uma consulta a esta matriz pelo número dos *IDs* das cidades envolvidas no cálculo.

O NSGAI (DEB; JAIN, 2014), então, foi modificado dentro do Jmetal para conter um busca local. Esta busca local utiliza descoberta de vizinhança para variar as cidades de cada *cluster* como detentora do POP. O mecanismo de variação considera duas restrições a seguir:

- Não é permitido variar uma cidade *X* de um *cluster Y* para uma outra cidade que não pertença ao *cluster Y*.
 - Isso iria acabar causando uma configuração indesejada onde alguns *clusters* poderiam acabar ficando com mais de um POP e outros sem nenhum.
- Não é permitido variar cidades em mais de um *cluster* por vez.
 - Como já foi explanado, calcular o gradiente de uma função em determinado ponto, muitas vezes não é possível ou ainda computacionalmente custoso. Portanto, uma abordagem com pequenas excitações se torna mais interessante.

Fora essas duas restrições a busca é livre, inclusive, utilizando a já citada matriz de distâncias do CNS, a sua execução é bem facilitada, reduzindo (após esse processo) o cálculo de qualquer distância do escopo do problema a uma simples consulta à matriz.

Também não houve nenhuma restrição para que uma cidade que já foi a detentora do POP em uma geração passada e deixou de ser, não volte a ser detentora em uma nova descoberta de configuração dominante. Pois, a simples mudança de uma outra cidade, ou mesmo de um *link*, em outra parte da rede que se relacione com essa cidade em questão, pode tornar uma configuração de posse de POP anterior, mais vantajosa. Isso pode acontecer por causa da natureza do problema, se observamos o objetivo principal desta pesquisa e destacarmos uma parte da frase teremos a resposta no fragmento de frase, “considerando as penalidades da camada física e limitações inerentes à propagação do sinal óptico em longas distâncias”. A questão é que a distância entre as cidades influencia na escolha do tipo de fibra e dos tipos de LASERs para compensação de perdas de sinal obrigando o projeto a ficar mais caro.

Na prática o método de busca local sorteia qual dos *clusters* irá ser beneficiado pela busca local, após isso, o método descobre quais as três cidades mais próximas à cidade a qual se busca a vizinhança. Depois do levantamento destas três cidades existe um sorteio, entre as três, para decidir qual será a cidade escolhida para substituição. Entretanto, esta substituição é controlada pela dominação, ou seja, a nova configuração de rede é avaliada e se não dominar a primeira a substituição simplesmente não acontece. A busca local foi inserida logo antes do operador de seleção, desta forma, toda a população P já teria sido avaliada, sendo necessário apenas avaliar a nova configuração, pós busca, de cada rede e testar a dominância.

O operador de *Crossover* também foi adaptado para que que incluísse o cruzamento das cidades ao cruzar o cromossomo (que representa apenas os *links* entre as cidades). Dessa maneira, sempre que houvesse uma operação de cruzamento entre duas soluções, as cidades também seriam cruzadas respeitando o seguinte mecanismo:

O operador de *Crossover* percorre o cromossomo de duas soluções, índice a índice decidindo se haverá cruzamento baseado em um valor estatístico pré configurada pelo usuário. Para o número de cidades indicadas neste estudo ($15 \leq n \leq 21$ e $n = 25$) o cromossomo, de qualquer solução, sempre será maior que o *array* de cidades rótulo da matriz de conexão desta solução, já que, o tamanho do cromossomo segue a fórmula $\frac{n(n-1)}{2}$, onde n é o número de cidades, ou seja, o tamanho do *array* de cidades. Dessa forma, chamando de *C.size* o tamanho do cromossomo, e de *R.size* o tamanho do *Array* de cidades rótulo da matriz, $C.size > R.size$. Assim, enquanto o operador de cruzamento está percorrendo o índice i de dois cromossomos de soluções quaisquer, se o operador for cruzar o elemento i – *simo* do cromossomo e $i < R.size$,

então, cruzará também, as cidades do *array* rótulo de cidades.

Diferente do operador de *crossover* que foi alterado para incluir as cidades, o operador de mutação foi preservado em sua forma original. Isso porque a busca local, por si só, já faz permutações nas cidades do *Array* de cidades quando encontra uma nova configuração dominante. Ou seja, a busca local, já é um tipo de mutação de cidades, e, feita de forma controlada. Por este motivo o operador de mutação não incluiu as cidades, pois acabaria fazendo a troca de cidades sem testar a dominação, o que poderia deixar o algoritmo sem rumo e terminar atrasando a convergência.

O *framework* CNS possui em suas atribuições uma aplicação chamada BONS. O BONS é um simulador de redes baseado no SIMTON (CHAVES et al., 2010), capaz de simular o tráfego de uma rede e calcular a probabilidade de bloqueio, a conectividade algébrica, a conectividade natural, o consumo em Watts, e o CAPEX. O BONS torna-se então, a ferramenta perfeita pra ser usada como função de *fitness* dos testes.

A única restrição configurada para o NSGAIII de forma geral foi que a conectividade algébrica não fosse zero, pois isso configuraria um *cluster* ilhado, ou seja, sem *links* da rede ativa que o atendesse.

4.3 Arranjo experimental

Nesta seção serão descritas as diversas configurações dos experimentos, em primeiro lugar irá ser descrito o arranjo da etapa de pré-processamento baseado em *clustering*, depois haverá a descrição da etapa de busca multiobjetivos.

4.3.1 Arranjo experimental da etapa de pré-processamento baseado em *clustering*

Além das configurações citadas na fundamentação teórica para o algoritmo de *clustering*, na implementação dos códigos de teste usados nesse projeto, existe uma configuração extra, a ser feita pelo usuário, é o número máximo de iterações do algoritmo. Caso não seja configurado pelo usuário, este número é automaticamente configurado para mil. A Tabela 3 mostra a configuração do experimento.

Tabela 3 – Tabela de configuração de etapa de clustering.

Parâmetros	Valores
Número de K cluster	De 4 a 30
Número máximo de iterações	200
Número de execuções	30
Métrica de distância	Euclidiana
Centroides iniciais	Aleatórios

Fonte: o autor.

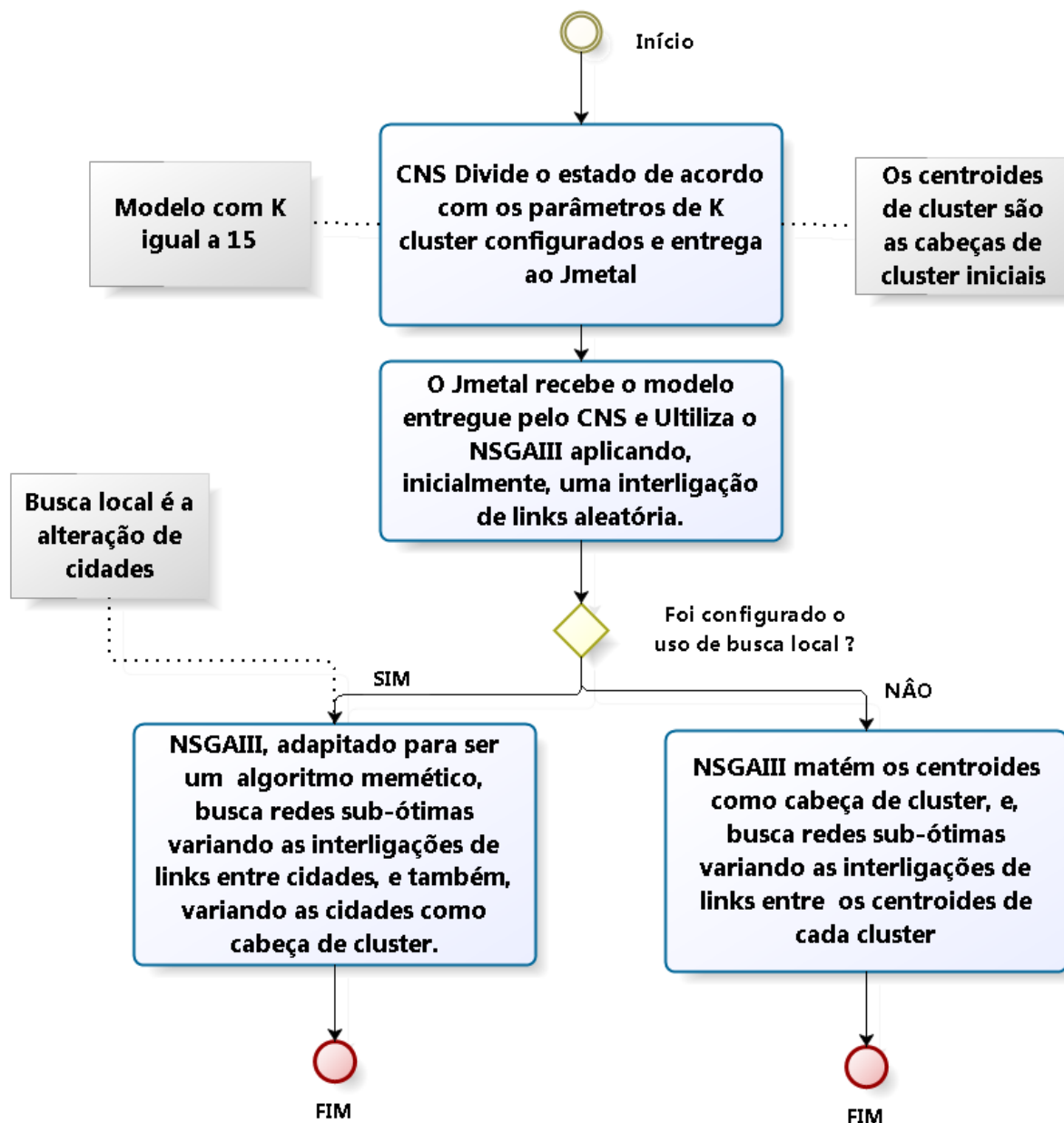
Foi realizado um experimento usando o *Kmeans* variando o número de K clusters entre 4 e 30. A métrica de distância foi a distância euclidiana, os centroides iniciais foram aleatórios e com o número máximo de iterações configurado como 200 (valor escolhido empiricamente). Foram realizadas trinta repetições do algoritmo para cada K em questão. Além disso, a aplicação estava preparada para que o próprio algoritmo interrompesse as iterações, mesmo antes de atingir as 200, se ao longo de suas execuções a posição dos centroides parassem de variar. Após essa etapa decidiu-se usar um tamanho de K igual a 15 para quando o *Kmeans* fosse trabalhar integrado com o NSGAIII, os motivos dessa decisão serão explicitados na próxima seção.

4.3.2 Arranjo experimental da etapa de otimização baseada em algoritmos evolucionários multiobjetivos com busca local

Com a fundamentação teórica para algoritmos multiobjetivos já bem discutida, foi possível, então, começar a descrever como foram os resultados dos testes.

O Jmetal (DURILLO; NEBRO; ALBA, 2010; NEBRO; DURILLO; VERGNE, 2015) é *framework* com excelentes recursos, nele já existe uma implementação geral de vários algoritmos evolucionários, bastando apenas que o desenvolvedor adéque o código do algoritmo, escrito no Jmetal de forma geral, para a realidade do problema a ser tratado. Foi utilizando os recursos do Jmetal que foram preparados os experimentos deste estudo de acordo com a Figura 16:

Figura 16 – Fluxo do multiobjetivos.



Fonte: o autor.

Como mostra a Figura 16 o CNS (ARAÚJO, 2015) foi configurado para usar o *Kmeans* com valores de *K* igual a 15, e o motivo disso foi simples. De acordo com a etapa de *clustering*, valores de *silhouette* mais elevados foram encontrados em tamanhos de *K cluster* pequenos como quatro. Entretanto, para manter um bom *trade-off* entre *clusters* com boa avaliação de índice de *silhouette* e tamanhos de *cluster* que não inviabilizem a rede PON foram selecionados alguns números de *K* entre 15 e 25. Para efeito de testes, usamos o valor de *K* igual a 15 que do subgrupo selecionado apresenta melhor valor de *silhouette*.

Foram executados 22 experimentos com a seguinte configuração para os algo-

ritmos, a Tabela 4 Mostra a configuração dos experimentos.

Tabela 4 – Tabela de configuração da etapa de busca multiobjetivos.

Parâmetro	Valor
Número máximo de iterações	500
Tamanho da população	40 indivíduos
Número de objetivos	4
Número de restrições	1
Número de cabeças de <i>clusters</i>	15
Ciclos de experimentos sem operador de busca local	11
Ciclos de experimentos com operador de busca local	11

Fonte: o autor.

Além do tamanho de K *cluster* (na Tabela 4, representado pelo número de cabeças de *cluster*), ter sido configurado para 15, ainda houveram as configurações necessárias para o funcionamento do algoritmo de busca multiobjetivos, o NSGAIII. O tamanho da população, escolhido empiricamente, foi configurada em 40 indivíduos, o número de objetivos foram quatro, apenas uma restrição que é a conectividade algébrica igual a zero (significa redes desconectadas), por fim, o número máximo de iterações foi configurado como 500. Além disso, das 22 execuções realizadas, 11 foram para o método sem busca local e 11 foram para o método com busca local.

Os experimentos retornaram uma geração de soluções final com 456 elementos subótimos. Foram realizadas mais de 228 mil avaliações de *fitness* na execução sem busca local e 456 mil avaliações de *fitness* na execução com busca local, visto que, cada indivíduo de cada população precisava ser reavaliado após a reconfiguração de cidades pela busca pra testar a dominância. O que dobrou o número de avaliações de *fitness*. Na execução com busca local uma média de 10379 soluções dominantes foram encontradas pela busca nas onze execuções.

Além disso, o BONS necessita que seja configurado o número de erlangs (medida de tráfego) para que seja realizado a simulação.

O Jmetal foi implementado para trabalhar com problemas de minimização, ou seja, quanto menores valores dos objetivos melhor. Nesse caso, as métricas de PB, CAPEX e consumo energético não precisaram de nenhum artifício para serem representadas, porque naturalmente, pequenos valores dessas métricas são melhores. Já a conectividade algébrica tem valores ótimos com altos valores, para resolver este

conflito os valores de conectividade algébrica foram invertidos após serem retornados pelo BONS, ficando como:

$$\text{conectividade algébrica inversa} = \frac{1}{(1 + \text{Conectividade algébrica})}. \quad (4.1)$$

Dessa forma, a conectividade algébrica inversa, resultaria em 1 quando a conectividade algébrica for igual a zero, e, caso a conectividade algébrica tivesse altos valores a conectividade algébrica inversa ficaria com valores menores que um, passando então, a poder ser avaliada em um problema de minimização.

5 Resultados

Neste capítulo serão apresentados os resultados dos experimentos, primeiramente haverá considerações sobre os resultados da etapa de pré-processamento baseado em *clustering* por médias de distância. Irá ser demonstrado valores de K entre quatro e trinta e como isso influencia o índice de avaliação da qualidade do *cluster*, logo após será identificado uma forma de mitigação do risco das distâncias inter *cluster* ficarem acima das especificações de uma rede PON. Por fim, haverá a parte de otimização baseada algoritmos evolucionários multiobjetivos com busca local onde serão apresentados gráficos de comparação de resultados de 11 execuções independentes das duas modalidades do algoritmo, com e sem busca local.

5.1 Resultados do pré-processamento baseado em *clustering* por médias de distâncias

Foi observado que o cálculo dos centros de *clusters* não excederam a 37 iterações como na Tabela 5 que relaciona o número de *clusters* K e o número mínimo e máximo de iterações atingidas.

Tabela 5 – Máxima Iteração por K cluster: Em negrito o maior alcance de iteração.

K	Iteração Mínima	Iteração Máxima	K	Iteração Mínima	Iteração Máxima
4	4	21	17	5	21
5	3	22	18	5	20
6	4	23	19	4	18
7	6	22	20	5	20
8	4	22	21	4	15
9	4	32	22	4	12
10	4	27	23	4	14
11	3	32	24	4	14
12	6	33	25	4	14
13	7	37	26	3	12
14	5	25	27	4	16
15	5	21	28	4	20
16	4	29	29	4	11
17	5	23	30		

Fonte: o autor.

Também foi observado que pequenos valores de K como quatro produziam valores de índice de *silhouette* maiores, isso significa que com pequenos valores de K , os *clusters* são mais compactos e melhor divididos. Entretanto, escolher uma quantidade de *clusters* tão pequena inviabilizaria o atendimento de uma quantidade excessiva de cidades pela rede PON. O problema é que com poucos *clusters* a distância entre as cidades excederiam o máximo de 20 km suportando pelas redes PON na grande maioria dos casos (HERON et al., 2008). Era esperado que algumas cidades dos diversos *clusters* excedessem a esse limite. Entretanto, é necessário que se identifique alguma ação que amenize estes casos, quando isso for um dos objetivos principais do projetista da rede, essa ação foi identificada e será mostrada mais a frente. Contudo, nos casos em que isso acontecer, as cidades que passarem do limite máximo de distância, poderão ser atendidas por outra tecnologia como CWDM que cobre distâncias de 68 km (MONTHA; MANEEKUT; KAEWPLUNG, 2014), ou até mesmo por WDM ou DWDM.

Todavia, houveram composições de *cluster* para alguns números de K como

15, 16, 17, 18, 19, 20, 21 e 25, que se mantiveram próximo ao número inicialmente idealizado de dezoito *clusters*, e, obtiveram um índice de *silhouette* razoavelmente bom em comparação aos outros.

A Tabela 6 apresenta essas composições de *K cluster* ordenadas pelo índice de *silhouette*.

Tabela 6 – Índice de silhouette por quantidade K de cluster.

<i>Silhouette</i>	nº de <i>Clusters</i>		<i>Silhouette</i>	nº de <i>Clusters</i>
0,464	4		0,350	18
0,422	5		0,348	20
0,410	6		0,347	19
0,396	7		0,345	21
0,381	8		0,341	25
0,370	9		0,338	24
0,370	11		0,337	22
0,367	13		0,336	23
0,366	12		0,335	26
0,365	15		0,333	30
0,365	10		0,330	27
0,358	14		0,324	28
0,356	16		0,323	29
0,355	17			

Fonte: o autor.

Como mostrado na Tabela 6, as configurações de número de *K cluster* como 15, 16, 17, 18, 19, 20, 21 e 25, tem um *trade-off* compensatório entre índice de *silhouette* e uma quantidade razoável de *clusters* para que se possa mitigar a questão da distância intra *cluster*.

Essa questão do alcance é um dos motivos pelo qual não se pode usar uma rede PON pra atender toda a rede de uma grande região como um estado, devendo assim, o *backbone* da rede ser atendido por tecnologias mais eficazes e mais caras como o WDM e DWDM. A Tabela 7 mostra a distância, em km, entre os centroides dos *clusters* com *K* de 4 a 30.

Tabela 7 – Distância entre Centroides de cluster.

K	Distância Máxima	Distância Mínima	Distância Média
4	471,88	134,74	272,55
5	477,23	78,13	251,23
6	479,06	89,69	231,45
7	546,84	86,71	253,04
8	502,15	51,30	212,40
9	576,01	57,95	240,65
10	546,84	48,39	213,28
11	630,97	47,38	235,98
12	550,02	25,63	216,69
13	557,83	47,30	210,97
14	586,00	41,97	230,27
15	557,83	36,96	195,32
16	570,09	23,25	190,13
17	616,55	37,29	244,35
18	648,71	36,31	229,79
19	591,85	27,65	231,58
20	604,32	33,87	220,55
21	648,64	37,42	245,35
22	638,39	32,22	231,44
23	630,25	23,60	261,45
24	652,52	14,53	239,01
25	648,64	23,25	213,83
26	681,98	16,01	260,42
27	630,25	20,27	226,98
28	629,78	21,95	215,37
29	648,71	14,53	213,42
30	681,98	23,98	263,61

Fonte: o autor.

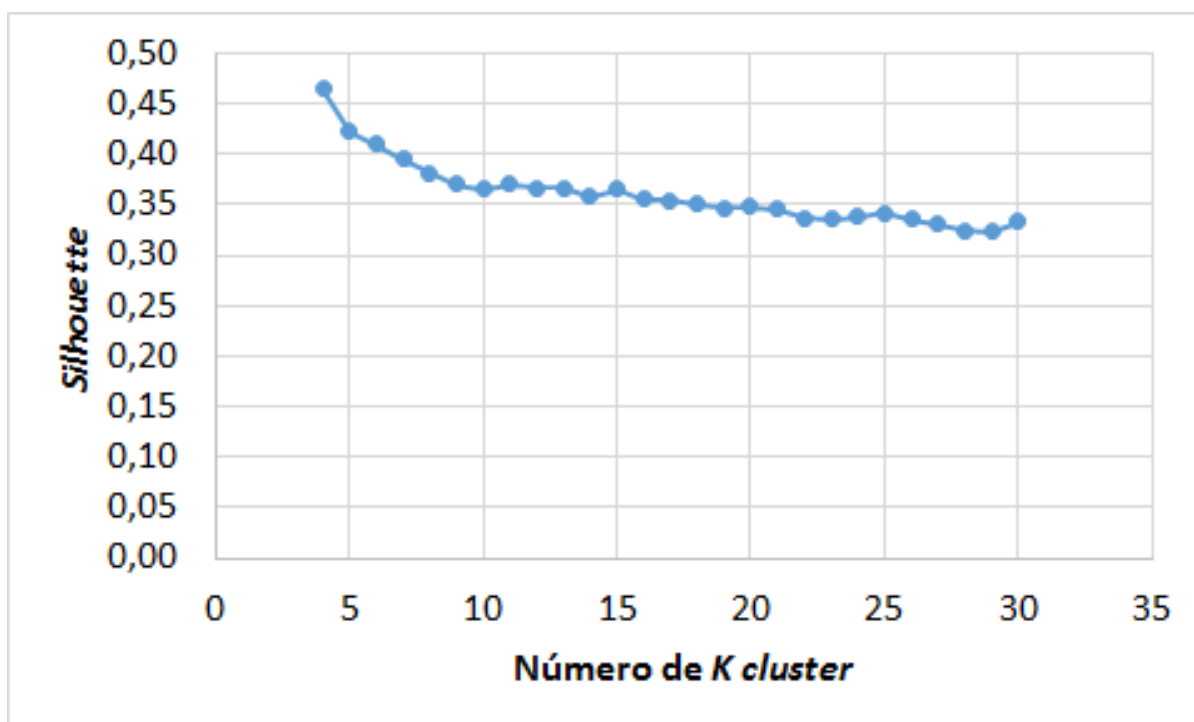
É possível perceber que a média das distâncias entre cabeças de *cluster* (cidades escolhidas para ficarem com o POP e fazer a interligação entre os *clusters*) é bem maior que o alcance máximo de uma rede PON que é de 20 km (HERON et al., 2008, p. 163), a Figura 17 mostra o alcance de diversas tecnologias de redes PON.

Figura 17 – Evolução da rede PON e alcance, retirado de (HERON et al., 2008 p. 163).

Time-frame	Technology	Organization	Bandwidth	Typical characteristics
Standards				
1998	APON	FSAN—ITU	625 Mbps/125 Mbps	1:32 split, 20 km
2002	BPON	FSAN—ITU G.983	625 Mbps/125 Mbps RF overlay	1:32 split, 20 km
2005	GPON	FSAN—ITU G.984	2.5 Gbps/1.25 Gbps	1:32 split, 20 km
2004	EPON	IEEE 802.3ah	1.25 Gbps/1.25 Gbps	1:16, 1:32 split, 20 km
Future	NGA	FSAN—ITU	10 Gbps	TBD
Future	10G EPON	IEEE 802.3 av	10 Gbps	TBD
Research				
Mid-90s	SuperPON	FSAN—ITU		1:2048, collector ring
		Stanford University SUCCESS		
		EU PIEMAN	32 × 10 Gbps	

Fonte: (HERON et al., 2008, p.163).

Como dito anteriormente, grandes valores de índice de *silhouette* resultam de *clusters* mais coesos e mais bem divididos. A Figura 18 mostra a curva do índice de *silhouette* com os valores de *K* configurados.

Figura 18 – Gráfico de índice de *silhouette* com valor de *K cluster* de 4 a 30.

Fonte: o autor.

Como demonstrado na Tabela 6, e pode ser visto visualmente na Figura 18, o melhor valor de *K* para o índice de *silhouette* foi quatro, nos experimentos desse projeto, as cidades para esse número de *K cluster* ficaram divididas da seguinte forma:

- *Cluster 1*: Afrânio, Araripina, Belém do São Francisco, Bodocó, Cabrobó, Cedro, Dormentes, Exu, Granito, Ipubi, Lagoa Grande, Moreilândia, Orocó, Ouricuri, Parnamirim, Petrolina, Salgueiro, Santa Cruz, Santa Filomena, Santa Maria da Boa Vista, Serrita, Terra Nova, Trindade, Verdejante.
- *Cluster 2*: Agrestina, Aguas Belas, Alagoinha, Altinho, Angelim, Arcoverde, Belém de Maria, Belo Jardim, Bom Conselho, Brejão, Brejo da Madre de Deus, Buíque, Cachoeirinha, Caetés, Calçado, Canhotinho, Capoeiras, Caruaru, Catende, Correntes, Cupira, Garanhuns, Iati, Ibirajuba, Jaqueira, Jataúba, Jucati, Jupi, Jurema, Lagoa do Ouro, Lagoa dos Gatos, Lajedo, Maraial, Palmeirina, Panelas, Paranatama, Pedra, Pesqueira, Poção, Quipapá, Saloá, Sanharó, Santa Cruz do Capibaribe, São Benedito do Sul, São Bento do Una, São Caitano, São João, Tacaimbó, Terezinha, Venturosa, Xexéu.
- *Cluster 3*: Afogados da Ingazeira, Betânia, Brejinho, Calumbi, Carnaúba, Carnaubeira da Penha, Custódia, Flores, Floresta, Ibimirim, Iguaracy, Inaja, Ingazeira, Itacuruba, Itaíba, Itapetim, Jatobá, Manari, Mirandiba, Petrolândia, Quixaba, Santa Cruz da Baixa Verde, Santa Terezinha, São José do Belmonte, São José do Egito,

Serra Talhada, Sertânia, Solidão, Tabira, Tacaratu, Triunfo, Tupanatinga, Tuparetama.

- *Cluster 4*: Abreu e Lima, Agua Preta, Aliança, Amaraji, Aracoiaba, Barra de Guabiraba, Barreiros, Bezerros, Bom Jardim, Bonito, Buenos Aires, Cabo de Santo Agostinho, Camaragibe, Camocim de São Félix, Camutanga, Carpina, Casinhas, Chã de Alegria, Chã Grande, Condado, Cortês, Cumarú, Escada, Feira Nova, Ferreiros, Frei Miguelinho, Gameleira, Gloria do Goitá, Goiana, Gravata, Igarassu, Ilha de Itamaracá, Ipojuca, Itambé, Itapissuma, Itaquitinga, Jaboatão dos Guararapes, Joao Alfredo, Joaquim Nabuco, Lagoa do Carro, Lagoa de Itaenga, Limoeiro, Macaparana, Machados, Moreno, Nazaré da Mata, Olinda, Orobó, Palmares, Passira, Paudalho, Paulista, Pombos, Primavera, Recife, Riacho das Almas, Ribeirão, Rio Formoso, Sairé, Salgadinho, Santa Maria do Cambucá, São Joaquim do Monte, São José da Coroa Grande, São Lourenço da Mata, São Vicente Ferrer, Sirinhaém, Surubim, Tamandaré, Taquaritinga do Norte, Timbaúba, Toritama, Tracunhaém, Vertente do Lério, Vertentes, Vicência, Vitória de Santo Antão.

A Figura 19 mostra a plotagem no mapa de como ficou dividido o estado de Pernambuco depois da Etapa de *clustering* com K igual a quatro.

Figura 19 – Mapa de Pernambuco após *clustering* com K igual a 4.



Fonte: (GOOGLE, 2017).

Apesar da curva de *silhouette* mostrar que o aumento do número de *clusters* implica em *clusters* menos compactos e menos bem divididos. Foi observado nos experimentos que o processo de *clustering* com K igual a 30 diminui as distâncias médias entre as cidades de cada *cluster*. Isso pode ser útil, por exemplo, como medida a ser tomada caso o projetista esteja mais interessado em diminuir as cidades dos *cluster* que não poderão ser atendidas por redes do tipo PON por ficar com mais de 20 km de distância da cabeça de *cluster*. Essa questão se torna mais evidente quando se observa que, na segunda fase deste estudo, quando se dará o uso de um algoritmo multiobjetivo, a cabeça do *cluster* não será necessariamente o centroide do *cluster*.

Após o algoritmo multiobjetivo mexer nas configurações da topologia de rede em busca de melhores configurações, a cabeça do *cluster* poderá variar conforme os resultado do algoritmo.

A Tabela 8, ordenada pela distância média, mostra a distância máxima, a distância mínima e a distância média, entre todas as cidades de cada *cluster* entre 1 e 30, ao se elevar o número de *K cluster* até 30, ou seja ao se dividir o estado em 30 grupos.

Tabela 8 – Distância cidade a cidade dentro do *cluster*.

Cluster	Distância Máxima	Distância Mínima	Distância Média	Cluster	Distância Máxima	Distância Mínima	Distância Média
26	12,61	6,32	8,93	21	62,02	7,35	28,07
20	17,98	4,59	12,50	11	58,42	11,39	28,33
27	30,89	5,07	16,59	1	48,67	20,39	30,30
29	36,82	5,92	17,94	23	31,99	31,99	31,99
5	31,09	11,35	19,30	6	33,04	33,04	33,04
17	38,73	6,24	20,39	13	60,89	8,43	33,24
25	36,14	5,13	20,72	24	57,09	18,52	33,25
15	41,88	9,23	20,95	8	63,80	13,88	33,52
18	31,96	9,86	21,15	7	34,53	34,53	34,53
9	40,36	5,57	21,18	14	75,02	11,19	36,49
12	39,96	9,10	21,47	4	55,33	22,59	39,65
19	51,25	6,57	25,51	22	73,63	15,55	42,34
30	45,53	11,03	25,90	28	82,20	22,18	45,92
3	52,77	9,10	26,40	10	88,01	18,63	49,28
16	48,76	5,91	27,23	2	51,06	51,06	51,06

Fonte: o autor.

A Tabela 9, também ordenada pela distância média, mostra as distâncias de todos os nós para o centroide do *cluster*.

Tabela 9 – Distância máxima, Mínima e Média entre as cidades e o centroide de cada cluster.

Cluster	Distância Máxima	Distância Mínima	Distância Média	Cluster	Distância Máxima	Distância Mínima	Distância Média
26	7,86	6,32	7,09	11	31,62	11,39	20,86
20	17,98	5,39	11,19	21	50,30	9,05	20,89
27	18,88	5,07	12,79	8	35,01	13,88	24,66
29	24,68	5,92	13,07	1	28,14	22,37	24,70
5	19,57	12,92	15,49	24	32,67	18,52	25,23
15	22,63	9,23	15,55	14	54,01	16,23	26,75
25	19,16	8,14	16,32	13	43,60	8,78	27,60
12	20,02	13,58	16,41	23	31,99	31,99	31,99
17	30,53	7,30	16,76	6	33,04	33,04	33,04
9	28,84	10,65	17,33	4	47,60	22,59	33,06
18	23,30	12,13	17,45	22	57,80	16,02	33,29
19	32,00	7,81	18,98	7	34,53	34,53	34,53
3	27,78	9,37	19,60	28	56,39	22,18	35,49
16	36,29	12,68	20,12	10	58,32	25,94	40,55
30	31,75	14,26	20,24	2	51,06	51,06	51,06

Fonte: o autor.

As Tabelas 8 e 9 mostram que as médias de distâncias entre as cidades de um *cluster* diminuem quando aumentamos o número de *clusters* para dividir uma região. Isso se mostra uma ação de controle de distâncias intra *cluster* para ser usada pelo projetista. Entretanto, para este estudo de caso, os experimentos mostraram que com o aumento de K acima de trinta começa a haver uma frequência maior de retorno do algoritmo de *clusters* com uma cidade só, o que é um caso indesejado. Este aspecto pode ser melhor explorado usando outros algoritmos de *clustering* em trabalhos futuros, por exemplo o *Kmedoid*.

5.2 Resultados da Otimização baseado em algoritmo evolucionários multiobjetivos com busca local

Dos resultados observados, pode-se perceber que um fator importante a ser apresentado é o tempo de execução, essa seção será iniciada apresentando isso. A avaliação de *fitness* depende de simulação, o que é computacionalmente custoso, cada avaliação de *fitness* demora tempos diferentes dependendo da configuração. O principal custo computacional está associado às estimativas de probabilidade de bloqueio, quando há um número alto de carga da rede (200 erlangs). Em uma visão geral, redes ruins costumam ser rapidamente avaliadas, pois, com uma requisição de tráfego tão grande, uma situação de bloqueio é facilmente identificada em uma rede ruim. Entretanto, uma rede boa, teria um custo de cálculo maior por causa do número de simulações que teriam que ser testadas antes que se encontre uma situação de bloqueio.

Quando se diminui o valor da carga da rede, mesmo as redes ruins, tendem a demorar mais para serem avaliadas, outra configuração que aumenta o tempo de execução é o tamanho da população. Quanto maior a população, mais indivíduos terão que ser evoluídos, retardando o tempo total da execução.

Os experimentos foram configurados com o número de erlangs em 120, e a população em 40, com essa configuração, as redes ruins, geralmente encontradas nas iterações iniciais, eram avaliadas numa média de 10 avaliações de rede por segundo, entretanto, com a evolução das iterações, redes melhores eram encontradas e o número de avaliações por segundo ficavam entre uma e duas avaliações de rede. Em algumas execuções o tempo de avaliação chegou a quatro segundos por rede. No caso da execução com busca local esta questão do tempo se agravava mais ainda, já que, redes bem melhores eram alcançadas com muito menos iterações, além do mais, a avaliação da rede com o uso da busca, não se trata apenas da avaliação, e sim, da avaliação associada a uma descoberta de vizinhança. O tempo total de execução sem busca local foi em média 12 h 06 min e com busca local 28 h 08 min, nas 11 execuções, a Tabela 10 mostra os dados estatísticos do tempo de execução.

Tabela 10 – Tabela de estatísticas do tempo, valores em horas e minutos.

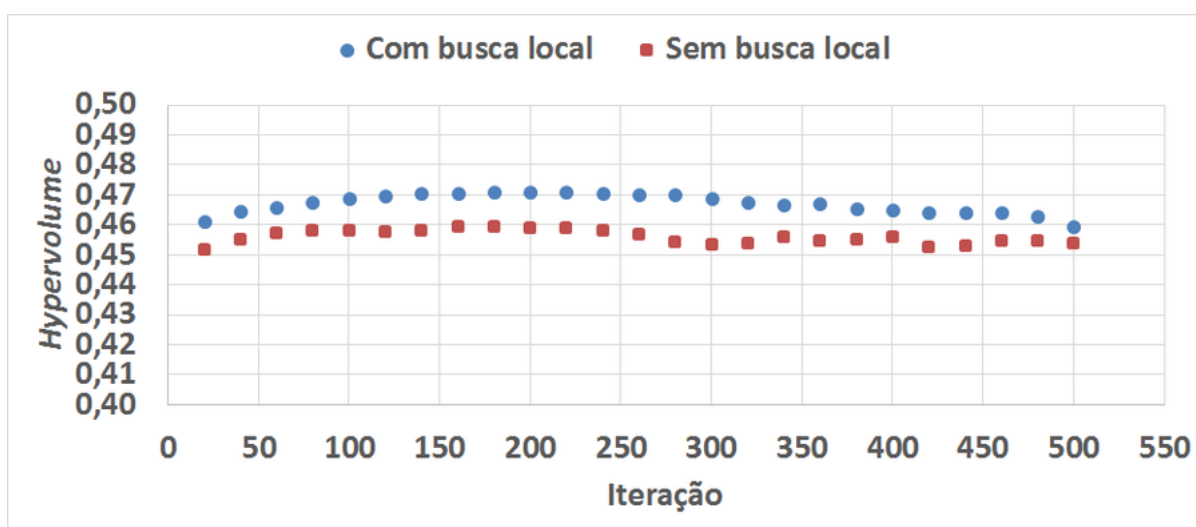
Método	Média em horas	Desvio padrão em horas	Erro em horas	Número de execuções
Sem busca	12:08	10:15	3:05	11
Com busca	28:06	12:37	3:45	11

Fonte: o autor.

Segundo o indicativo de erro estatístico, a diferença entre os dois métodos nem é tão grande, trata-se apenas, de 40 minutos. Entretanto este aumento de tempo é compensado pela vantagem de se obterem melhores soluções em uma quantidade menor de iterações (gerações). Nos resultados coletados, o método com busca local conseguiu resultados que indicavam maior compromisso entre as métricas de objetivos (aferido pelo indicador externo *hypervolume*) já em torno da metade das iterações programadas (500 iterações). Já o método sem busca local ainda precisou chegar ao número de iteração final (500 iterações) pra chegar a resultados que se assemelharam ao método com busca, ainda assim ficando abaixo. Além disso, a busca local é um procedimento que é de fácil paralelização, podendo ter este efeito de aumento de tempo mitigado em plataformas paralelas.

O Jmetal foi configurado para gerar saídas com os resultados a cada 20 iterações (20, 40, 60... 500), ou seja, foram 25 saídas de resultados por execução. Chamando a iteração 20 de *i20*, a iteração 40 de *i40*, e assim por diante, tivemos a impressão de (*i20*, *i40*, *i60*... *i500*). A média das 11 saídas de todos os *i20* até todos todos os *i500*, foram plotadas no gráfico de *hypervolume* da Figura 20.

Figura 20 – *emphhypervolume* da média de 11 execuções.



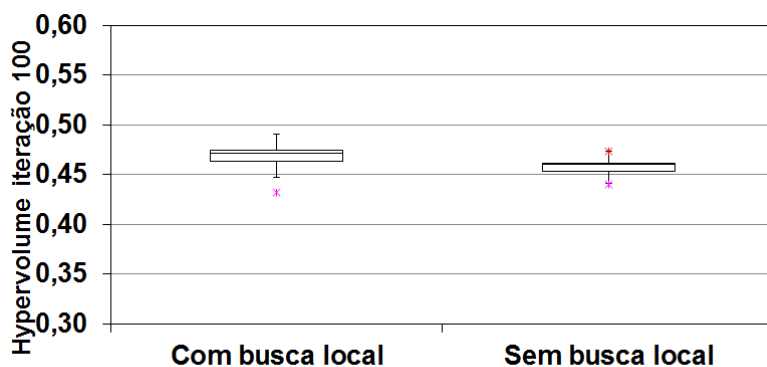
Fonte: o autor.

Os círculos, da Figura 20, são as médias de *hypervolume* de 20 em 20 iterações das execuções com busca local. Já os quadrados são das sem busca local. Veja que na média, o método com busca foi superior ao método sem busca em todos os pares de aferição.

Um gráfico de box-plot é uma forma de analisar os resultados de várias execuções de métodos diferentes, o gráfico contém em cada coluna uma figura que representa o pior, o melhor e o resultado médio de todas as execuções de cada método analisado, no caso deste projeto, 11 execuções do método com busca e 11 sem busca.

Um gráfico de box-plots foi gerado para as iterações de 100, 200, 300, 400 e 500. A Figura 21 mostra a estatística descritiva para 11 execuções na iteração 100, para as abordagens com e sem busca local. Observa-se que a média das execuções com a busca local foi melhor que o método sem busca local já na iteração 100.

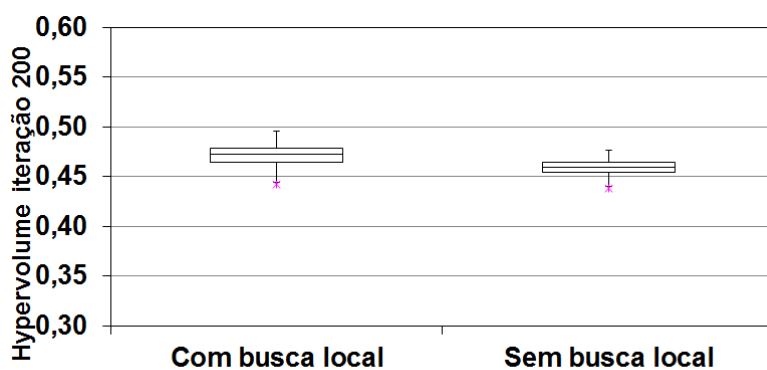
Figura 21 – Comparação por *hypervolume* com e sem busca iteração 100.



Fonte: o autor.

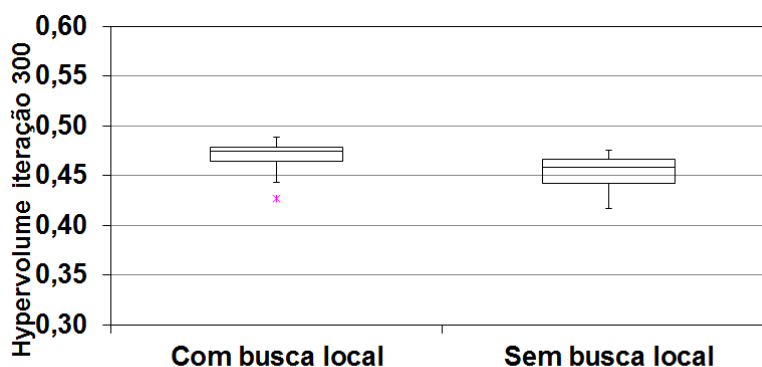
A Figura 22 mostra o box-plots com a estatística descritiva para 11 execuções na iteração 200, para as abordagens com e sem busca local. O método com busca local se manteve com a uma média melhor desde a iteração 100.

Figura 22 – Comparação por *hypervolume* com e sem busca iteração 200.



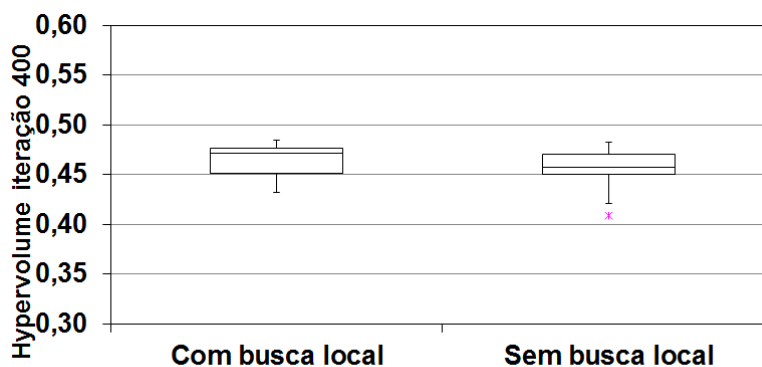
Fonte: o autor.

A Figura 23 mostra o box-plots com a estatística descritiva para 11 execuções na iteração 300, para as abordagens com e sem busca local. O método com busca local por volta da metade das iterações já atingia resultados satisfatórios enquanto o método sem busca ainda precisaria do dobro de iterações.

Figura 23 – Comparação por *hypervolume* com e sem busca iteração 300.

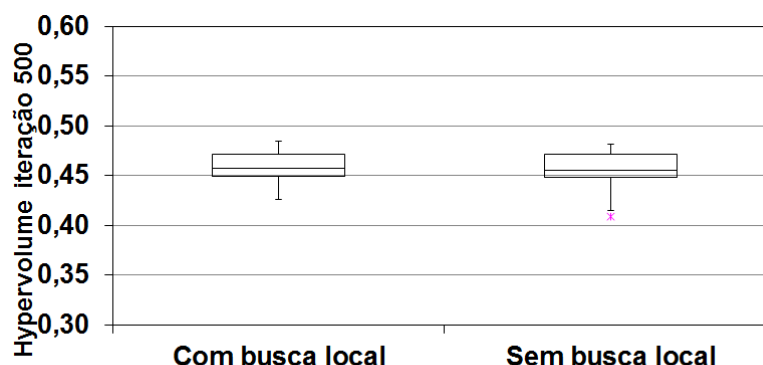
Fonte: o autor.

A Figura 24 mostra o mostra o box-plots com a estatística descritiva para 11 execuções na iteração 400, para as abordagens com e sem busca local. O método com busca local na média ainda se manteve melhor.

Figura 24 – Comparação por *hypervolume* com e sem busca iteração 400.

Fonte: o autor.

Por fim, a Figura 25 mostra o mostra o box-plots com a estatística descritiva para 11 execuções na iteração 500, para as abordagens com e sem busca local. só por volta da iteração 500 que o método sem busca local conseguiu resultado que se aproximavam do método com busca local. Ainda assim, na média ele se manteve inferior.

Figura 25 – Comparação por hypervolume com e sem busca iteração 500.

Fonte: o autor.

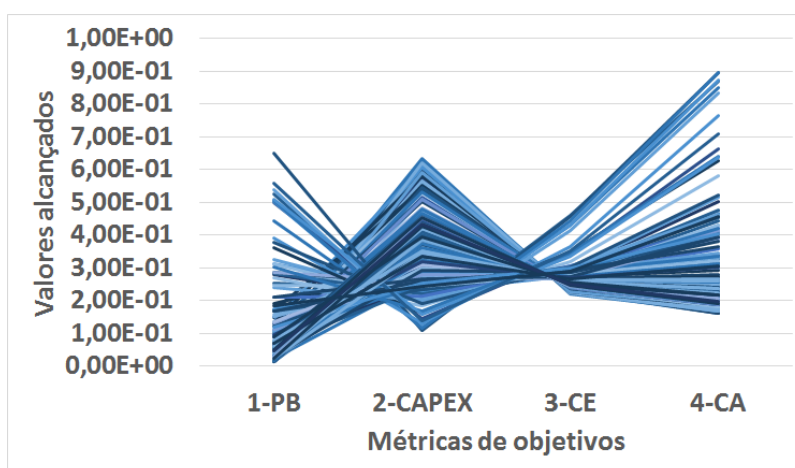
Foram feitos testes até 500 iterações, como mostram as Figuras de box-plots 21, 22, 23, 24, e 25. Foi selecionada, dessas iterações destacadas, a que vem logo após a metade 250. Ou seja, foi selecionado a de 300 iterações para plotar o gráfico de coordenadas paralelas normalizado com as médias dos valores dos quatro objetivos encontrado nas 11 execuções. A intenção disso foi demonstrar que desde a metade das iterações as execuções com busca local já encontravam resultados bem melhores, ou seja, com maior compromisso entre as métricas aferidos por um indicador externo (o *hypervolume*). Será mostrado o gráfico de coordenadas paralelas do método sem a busca local, e do método com a busca local. Neste tipo de gráfico, cada linha plotada corresponde à uma configuração de rede óptica diferente (diferentes topologias). O eixo vertical deste gráfico possui os valores das 4 métricas e o eixo horizontal apresenta um número que rotula cada métrica, inclusive com uma abreviação a título de facilitação de leitura, seguindo a seguinte convenção:

- 1) Probabilidade de bloqueio ou PB.
- 2) CAPEX.
- 3) Consumo energético em watts ou CE.
- 4) conectividade algébrica ou CA.

O ideal pra se estimar as redes mais adaptadas em função das métricas, seria testar por exaustão encontrando todas as possibilidades, entretanto, problemas complexos (NP-completos e NP-difíceis), não se resolvem em tempo polinomial. Sendo assim seria custoso, em termos de custo computacional, testar por exaustão. Por isso a meta-heurística é uma maneira mais adequada a resolver problemas desse tipo, mas, como se pode ver, deve ser feita uma análise estatística dos resultados, pois os algoritmos dessa natureza são probabilísticos, e execuções individuais podem apresentar

pequenas variações nos ganhos obtidos. Esse é o motivo de uma execução do mesmo algoritmo poder ser pior ou melhor que as outras execuções com as mesmas condições de configuração. A Figura 26 mostra o gráfico de coordenadas paralelas normalizado com o resultado da pior execução, sem busca local, na iteração 300. No caso do gráfico de coordenadas paralelas, o *hypervolume* é a área a partir do ponto W, demonstrado na Figura 11, formada na parte superior do gráfico. O valor calculado de *hypervolume* deste momento foi de 0,4172, em outras execuções houveram melhoras.

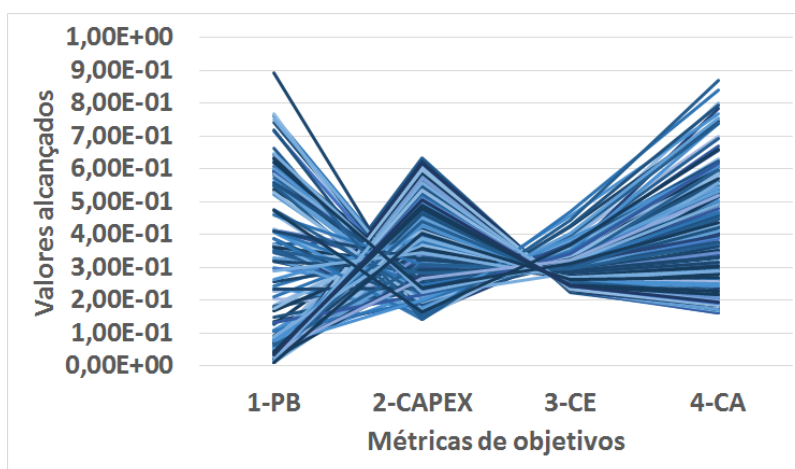
Figura 26 – Pior resultado das execuções sem busca, resultado capturado na iteração 300.



Fonte: o autor.

A Figura 27 mostra o gráfico de coordenadas paralelas normalizado com o resultado da melhor execução, sem busca local, na iteração 300. O valor calculado de *hypervolume* deste momento foi de 0,4761, bem acima da pior execução.

Figura 27 – Melhor resultado das execuções sem busca, resultado capturado na iteração 300.

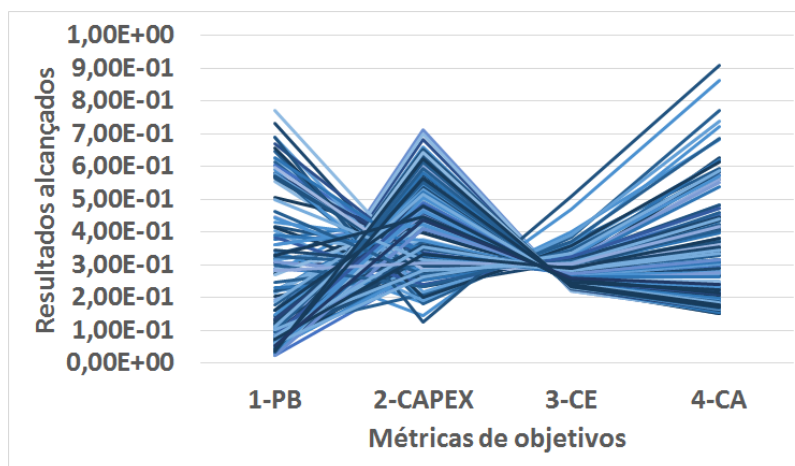


Fonte: o autor.

A Figura 28 mostra o gráfico de coordenadas paralelas normalizado com o resultado da execução que mais se aproximou da média global, sem busca local, na

iteração 300. O valor calculado de *hypervolume* deste momento foi de 0,4526, a média global de *hypervolume*, para esta iteração foi de 0,4536, os resultados estão bem próximos.

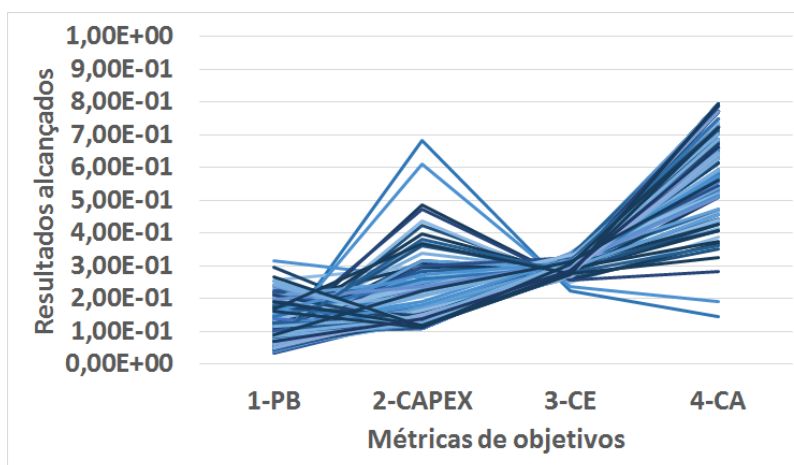
Figura 28 – Resultado mais próximo da média das execuções sem busca, resultado capturado na iteração 300.



Fonte: o autor.

A Figura 29 mostra o gráfico de coordenadas paralelas normatizado com o resultado da pior execução, com busca local, na iteração 300. O valor calculado de *hypervolume* neste momento foi de 0,4272, é possível ver que o pior resultado com busca local foi melhor que o pior sem a busca que teve o *hypervolume* de 0,4172. Entretanto, o pior resultado com busca local, que lembrando, obteve *hypervolume* de 0,4272 não é superior ao melhor resultado sem busca local que foi de 0,4761. Isso quer dizer que a técnica com busca foi melhor em todos os pares de testes, também foi melhor na média, mas estatisticamente falando, pode ter execuções que não se saia melhor que o método sem busca. Isso pode-se dever a possibilidade do método com busca ter ficado preso em ótimos locais, estudos futuros podem testar novas possibilidades de ampliar a busca.

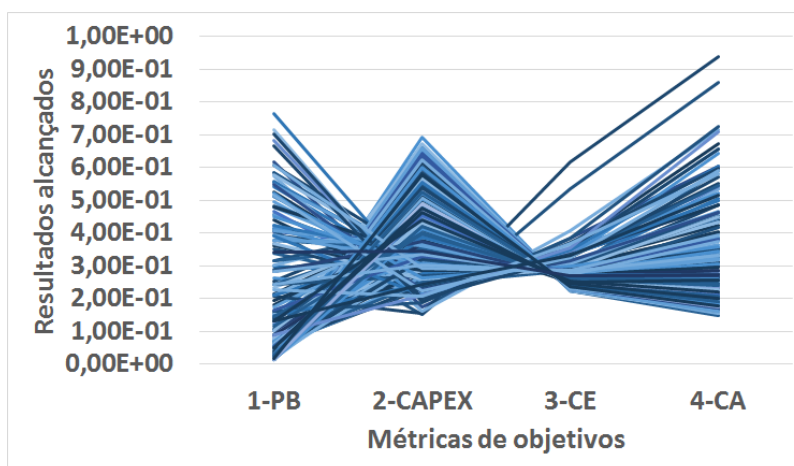
Figura 29 – Pior Resultado das execuções com busca, resultado capturado na iteração 300.



Fonte: o autor.

A Figura 30 mostra o gráfico de coordenadas paralelas normalizado com o resultado da melhor execução, com busca local, na iteração 300. O valor de *hypervolume* verificado neste momento foi de 0,4885, este valor supera todos os valores de *hypervolume* observados ao longo de todas as execuções do método sem busca local.

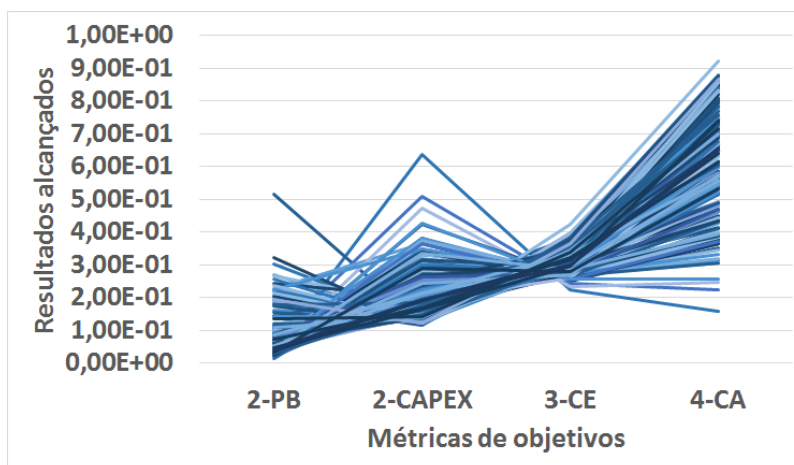
Figura 30 – Melhor Resultado das execuções com busca, resultado capturado na iteração 300.



Fonte: o autor.

A Figura 31 mostra o gráfico de coordenadas paralelas normalizado com o resultado da execução que mais se aproximou da média, com busca local, na iteração 300. O valor de *hypervolume* calculado neste momento foi de 0,4739, e, a média global de *hypervolume*, para esta iteração foi de 0,4689. Se for observado o valor da média do método sem busca local que foi de 0,4536, é possível ver que o método com busca local também se saiu melhor na média.

Figura 31 – Resultado mais próximo da média das execuções com busca, resultado capturado na iteração 300.



Fonte: o autor.

A Tabela 11 mostra os ganhos percentuais estatísticos dos resultados dos gráficos das figuras 26, 27, 28, 29, 30 e 31.

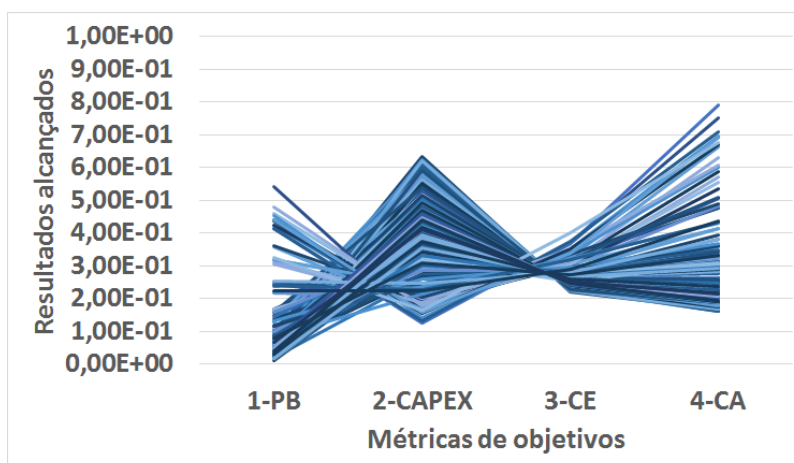
Tabela 11 – Legenda

Método / situação	Iteração	Valores alcançados	Ganho percentual em relação a mesma situação	média	Desvio padrão	Erro
Sem busca / Pior execução	300	0,4172	-	3,12%	0,97%	0,56%
Com busca / Pior execução	300	0,4272	2,34%			
Sem busca / Melhor execução	300	0,4761	-			
Com busca / Melhor execução	300	0,4885	2,54%			
Sem busca / Execução mais próxima da média	300	0,4526	-			
Com busca / Execução mais próxima da média	300	0,4739	4,49%			

Fonte: o autor.

Como exemplo final dos resultados escolhemos a rede mais barata de cada modelo (com e sem busca) de uma das execuções dos experimentos. Isso na iteração 500, ou seja, do resultado final. A Figura 32 mostra a curva de coordenadas paralelas da iteração 500 sem busca local. o valor final de *hypervolume* desta execução foi de 0,4093.

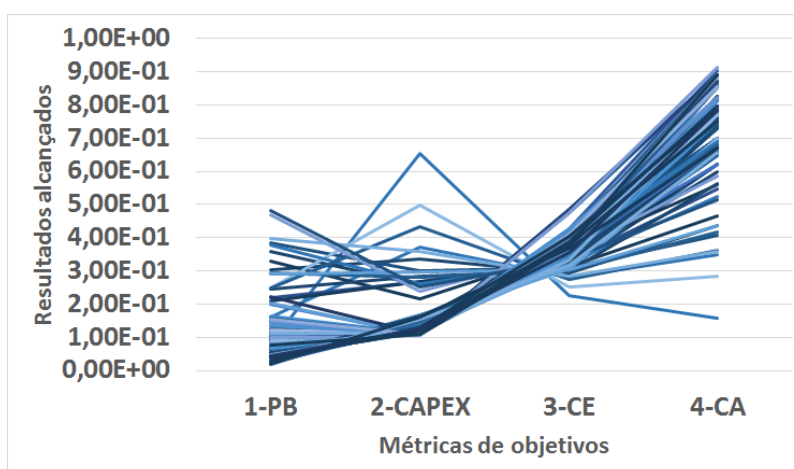
Figura 32 – Execução 1 sem busca, resultado final, capturado na iteração 500.



Fonte: o autor.

Já a Figura 33 mostra a curva de coordenadas paralelas da iteração 500 com busca local. O *hypervolume* calculado foi de 0,4851, ou seja superior ao método sem busca local, que como falado, foi de 0,4093.

Figura 33 – Execução 1 com busca, resultado final, capturado na iteração 500.



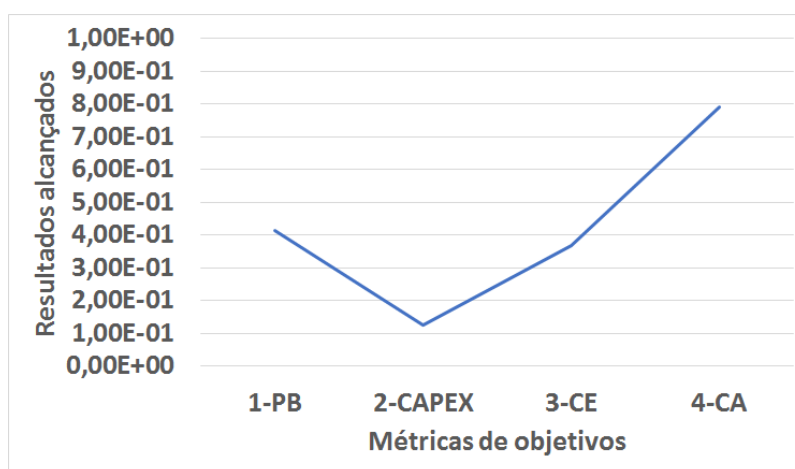
Fonte: o autor.

A Figura 34 mostra o gráfico de coordenadas paralelas para a rede mais barata que foi encontrada pelo método sem busca local na iteração 500. As métricas dessa rede são:

- 1) PB igual a 0,414250207 (ou seja acima de 40%).
- 2) CAPEX igual a 3677,044255 u. m.
- 3) Consumo energético igual a 1403188,477 watts.
- 4) Conectividade algébrica igual a 0,791296077 (com conversão para minimização).

É possível observar que se trata-se de uma rede que conseguiu um resultado que teve uma tendência maior para o zero na métrica de CAPEX (em relação as outras métricas destacadas no gráfico da Figura 34), mas em compensação, obteve um resultado de PB (acima de 40% para o tráfego de 120 erlangs), o qual, na Figura 35 veremos que ficou bem acima do método com busca (abaixo de 10%). Veja que uma rede que tem a métrica de PB alta (quanto mais alta pior) é uma rede que tem boa chance de não atender a demanda de tráfego a que é submetida e isso é um grande problema.

Figura 34 – Rede mais barata encontrada pelo algoritmo sem a busca local.



Fonte: o autor.

A Figura 35 mostra a rede que gerou a curva no gráfico da Figura 34, é uma rede que tem uma boa cobertura do estado, mas que poderia melhorar em outras questões técnicas como na probabilidade de bloqueio. Estudos futuros podem testar um número maior de iterações para ver se melhoram os resultados.

Figura 35 – Mapa da rede com menor CAPEX encontrada na iteração 500 do método sem busca local.



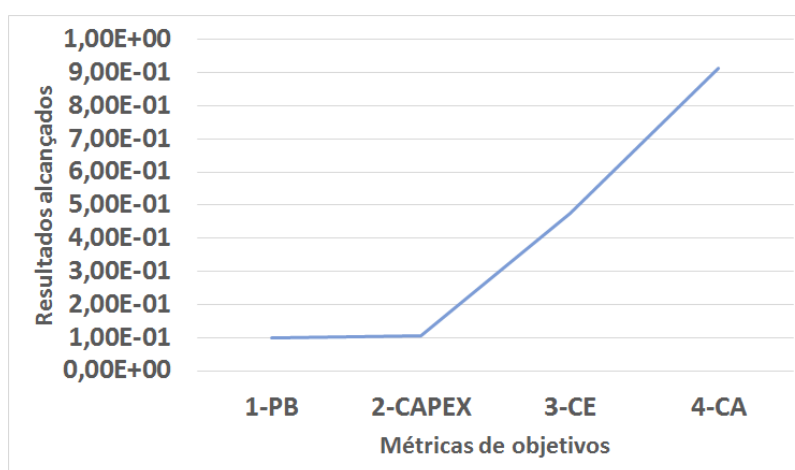
Fonte: o autor.

A Figura 36 mostra o gráfico de coordenadas paralelas para a rede mais barata que foi encontrada pelo método com busca local na iteração 500. As métricas dessa rede são:

- 1) PB igual a 0,099206349 (ou seja abaixo de 10%).
- 2) CAPEX igual a 3089,902702 u.m.
- 3) Consumo energético igual a 1802003,309 watts
- 4) Conectividade algébrica igual a 0,912929888 (com conversão para minimização).

É possível observar que foi uma rede que se saiu bem melhor que a rede do método sem busca local nas métricas de CAPEX e PB (menos de 10%). Essa rede representa bem a ideia que foi passada na hipótese de que o custo da rede pode diminuir e o desempenho pode melhorar se forem usadas cidades vizinhas de uma cidade que seria a “preferencial” para ser um concentrador de tráfego ou detentora do POP. No final das contas são duas redes não dominadas pois a rede do método sem busca local tem a conectividade algébrica melhor. Mas pelo menos no que se diz respeito a hipótese aqui levantada a rede do método com busca local foi bem mais fiel. Estudos posteriores podem combinar o método de busca aqui descrito com um método de busca que foque na conectividade algébrica.

Figura 36 – Rede mais barata encontrada pelo algoritmo com a busca local.



Fonte: o autor.

A Figura 37 mostra a rede que gerou a curva no gráfico da Figura 35. É possível notar que assim como na rede obtida com o método sem busca local, esta rede cobre

bem o estado. Isso se deve a estratégia proposta na etapa de *clustering*, que mitigou com total sucesso esta questão. Estudos posteriores também podem aumentar o número de iterações para testar se são encontradas novas populações que mantenham as boas métricas de PB e CAPEX e que tenham melhor conectividade algébrica.

Figura 37 – Mapa da rede com menor CAPEX encontrada na iteração 500 do método com busca local.



Fonte: o autor.

6 Discussão

Neste capítulo será feita uma discussão geral sobre os resultados obtidos comparando o que foi alcançado com a técnica proposta e os objetivos principais e específicos. Será feita considerações de forma geral e depois será discorrido sobre os resultados obtidos na etapa de pré-processamento baseado em *clustering* por média de distâncias. Finalmente, será considerado os resultados da etapa final de otimização baseada em algoritmos multiobjetivos com busca local e suas conquistas.

6.1 Discussão geral

Foi destacado no problema da pesquisa que seria proposto um método baseado em *clustering* e busca multiobjetivos acrescida de uma busca local, para um projeto de topologia de rede óptica de forma geral. Esse proposta foi alcançada, o método pode ser aplicado em qualquer situação, bastando apenas que, os nós terminais, possam ser atribuídos a localização geográfica. Também foi afirmado que seria usado o estado de pernambuco como um estudo de casos, e assim foi feito. Vale destacar também que isso, por si só, já representa alguma contribuição inovadora, pois, nas pesquisas realizadas no estado da arte, não foi identificado nenhum projeto que propôs isso de maneira automática.

O objetivo geral também destacou que seria investigado o uso de algoritmos de *clustering* e busca multiobjetivos acrescido de uma busca local, para permitir o projeto integrado de redes ópticas de longa distância e redes metropolitanas, considerando as penalidades da camada física e limitações inerentes à propagação do sinal óptico em longas distâncias. Pode-se considerar esta missão cumprida da mesma forma, a mescla da rede ativa (WDM, DWDM) e das redes passivas (PON), cumprem com os requisitos do proposto. Por fim, a busca local considera as penalidades da camada física ao modificar as diversas topologias de rede e observar a influência nas métricas que tais modificações causam.

6.2 Discussão da etapa de pré-processamento baseado em *clustering* por média de distâncias

O processo de *clustering* foi bem sucedido em vários aspectos desde a identificação do que seriam situações ruins (*clusters* com uma cidade só), o que influenciaria negativamente a métrica de CAPEX. Até a identificação do que possibilitou o alcance de um dos objetivos específicos proposto aqui: “Construir *clusters* mais coesos o possível“. Ao ser identificado que *clusters* com tamanho de K igual a quatro ofereciam

índice de *silhouette* de melhor qualidade foi identificado o valor ótimo para a o processo de *clustering*. Isso foi importante porque a partir dessa informação foi possível fazer o *trade-off* entre o número de K *clusters* e as distâncias intra *cluster* para mitigar a questão da possibilidade de inviabilização da implantação da rede PON por causa de suas limitações técnicas. Partindo desse conhecimento, então, foi identificado tamanhos de K *clusters* que minimizavam a métrica de CAPEX, porém sem perder muito a qualidade do *cluster*, continuando com características de *clusters* mais coesos e mais bem divididos.

Uma das principais preocupações era que a técnica de *clustering* garantisse a cobertura do atendimento a toda a região do estado. Sem precisar que depois do processo terminado, houvesse intervenção externa ao método para corrigir o alcance da rede. Conclui-se que esse problema foi tratado com sucesso já que a técnica garantiu o atendimento de todo o estado da forma planejada, sem maiores problemas, ao garantir o atendimento a cada *clusters* projetado de forma automática.

Detalhes técnicos de custo computacional também puderam ser encontrados neste estudo, como o número máximo de iterações para realização da etapa de *clustering*. Nos experimentos realizados foi identificado um máximo de 37 iterações para a conclusão total do processo chegando então a apenas 18,5% do total de iteração máxima programa que era de duzentas. Esse detalhe técnico mostra que, pelo menos nos experimentos realizados nesse estudo de caso, a técnica se quer é computacionalmente custosa na etapa de pré-processamento baseado em *clustering*.

6.3 Discussão da otimização baseada em algoritmos multiobjetivos com busca local

Apesar da importância de tudo o que foi encontrado neste estudo de caso, talvez, o que deva ser colocado em evidência primária seja a busca local. Nas pesquisas realizadas, foi averiguado que, estudos anteriores já haviam propostos topologias de redes de forma automática para diversas regiões do mundo e com os mais diversos algoritmos. Entretanto, nenhum dos estudo que foram levantados, continham a estratégia de usar a localização dos nós terminais como uma variável de decisão do algoritmo e não simplesmente uma informação estática.

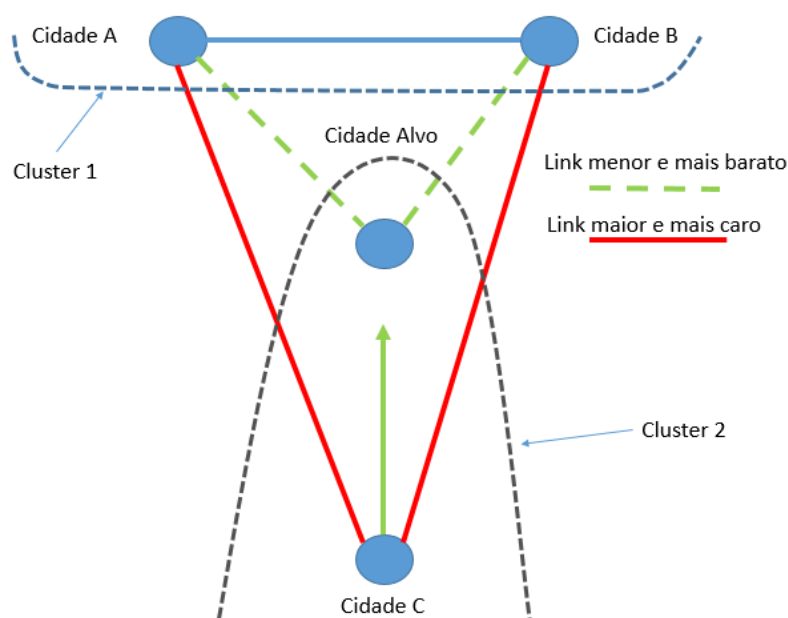
Como era esperado, o NSGAIII conseguiu suprir bem todas as expectativas do que seria um algoritmo genético de alto padrão em uma busca multiobjetivos. A convergência para curva de Pareto pela averiguação do *hypervolume*, foi observada aos poucos em todo a evolução das soluções mais adaptadas. Entretanto, o conhecimento de detalhes técnicos do problema incluídos ao funcionamento do algoritmo fez a convergência para a curva ser alcançada em muito menos iterações.

Fazendo uma análise, dos gráficos de coordenadas paralelas, é possível ver

que muito desse impacto está no objetivo de CAPEX e de PB, e, como foi explicitado na introdução quando foi levantado a hipótese, isso já era esperado: “a hipótese é que o custo da rede pode diminuir e o desempenho pode melhorar se forem usadas cidades vizinhas de uma cidade que seria a “preferencial” para ser um concentrador de tráfego ou detentora do POP”.

O algoritmo de *Clustering* gerou a lista de cidades a serem consideradas como preferenciais por serem centroides de *cluster* (note que já deste ponto a posição dos nós terminais já era uma variável de decisão do algoritmo). Isso, mesmo antes da busca local alterar as cidades dos nós terminais para tentar encontrar redes mais adaptadas. A Figura 38 ilustra a lógica do funcionamento da dominação de uma rede sobre a outra, neste caso, em função da métrica de CAPEX:

Figura 38 – Troca de cabeça de *cluster* em benefício de algumas métricas.



Fonte: o autor.

Uma modificação desta natureza também altera outras métricas além do CAPEX e da PB. Mas, não é possível deixar de observar que especificamente em relação ao CAPEX, esse benefício é mais claro aos olhos humanos, já que como foi dito anteriormente, o custo com a topologia da rede vem boa parte dos *links* de fibra e dos tipos de LASERs utilizados. Então, quanto mais longe os pontos a serem interligados, maior o tamanho da fibra (aumentando a necessidade de fibras com tecnologias mais caras) e maior a necessidade de LASERs mais potentes. Observe na ilustração da Figura 38 que o simples deslocamento entre duas cidades vizinhas pode afetar diretamente o tamanho do link e, conseqüentemente, afetar algumas métricas, dentre elas, provavelmente a métrica de CAPEX.

Se for recordado as regras de dominação explicadas na seção 3.2.4, veremos que se uma solução A tem avaliações de *fitness* W , X , Y e Z e uma mudança qualquer faz com que ela se transforme em uma solução B que obtenha uma avaliação W , X , Y e $(Z-N)$, pelas regras da dominação (todas as avaliações menores ou iguais e pelo menos uma menor), a solução B dominará a solução A . Portanto ao se mexer em uma rede e se obter uma outra rede que, no mínimo, mantenha todos os valores de avaliação iguais e diminua pelo menos um, estará se obtendo uma rede dominante (em um problema de minimização).

Vale destacar que a inclusão da busca não impediu o desenvolvimento geral do algoritmo, o que houve foi um impulso positivo. Esse foi um dos motivos que, com certeza, ajudaram a alcançar mais dois dos objetivos específicos propostos, o de garantir uma boa variedade de redes subótimas e o de garantir avaliações subótimas de métricas de Probabilidade de Bloqueio, CAPEX, consumo energético e conectividade algébrica. O último dos objetivos específicos (incluir conhecimento específico ao algoritmo de busca multiobjetivos de maneira a melhorá-lo) foi alcançado com a busca local. Pois, como demonstra os gráficos de *hypervolume*, houve boa convergência para curva de Pareto com muito menos iterações.

7 Conclusão

O uso de algoritmos de *clustering* e multiobjetivos na construção de redes ópticas de longa distancia integradas com redes metropolitanas foi uma integração de técnicas de inteligência computacional que funcionou para este problema em específico. Entretanto, o uso de conhecimento específico, especializando o algoritmo de busca multiobjetivos NSGAIII e transformando-o em um algoritmo memético contribuiu muito para o ganho de desempenho da técnica. Na fase de *clustering* foi possível identificar e mitigar todos os pontos críticos levantados em tempo de pesquisa. Além disso, dados importantes sobre custo computacional foram coletados. Como muitas vezes, em aplicações práticas, os objetivos são conflitantes, a identificação de que o uso de *clusters* mais coesos e mais bem divididos é conflitante com a necessidade de diminuir as distâncias intra *clusters* não chegou a ser algo totalmente inesperado. Situações de conflitos sempre irão aparecer, mas foi possível identificar os pontos ótimos para trabalhar um *trade-off*.

Um aspecto curioso na pesquisa é que a busca local não interferiu na formação do cromossomo. Ela interferiu em uma outra estrutura que o auxiliava na avaliação do cromossomo, entregando dados inerentes a composição da rede para levantamento das características, e assim, poder executar a avaliação. O que faz surgir a pergunta se uma busca local pode ter seu entendimento de uma maneira geral ou se cabe uma classificação futura na definição do que ela é através de seu mecanismo de funcionamento.

O custo computacional da etapa de multiobjetivos foi identificado como extremamente oneroso, justificando dessa forma, qualquer que seja a intervenção para que adicione alguma amenização nestes sentido. A execução de cada experimento do algoritmo nesta etapa durou horas e até dias com configurações que nem foram tão agressivas. Isso causa um problema difícil de mitigar, mesmo com a identificação de que o CNS substitui o cálculo das distâncias, entre cidades, por uma simples consulta a uma matriz pré processada. Tudo isso não foi capaz de não interferir no tempo de execução do algoritmo com a implementação da busca. Em contra partida na metade das iterações com a busca local, já foi possível identificar uma convergência para a curva de Pareto mais robusta, ganhando um parte do tempo perdido em função desta melhora. Não se deve esquecer de observar que ao obter melhores resultados com a metade das iterações a busca local evitou metade das etapas de mutação e cruzamento. Ou seja a busca funciona e apesar do custo computacional para o seu pleno funcionamento ser alto ela tem outros aspectos compensatórios. Além do mais, o tempo adicional da busca local é compensado pela melhor qualidade das soluções

obtidas e é um passo que pode ser facilmente paralelizado em trabalhos futuros.

7.1 Limitações identificadas

- 1) Na etapa de *clustering* o aumento do número de divisões de subgrupos acima de 30, ou seja, o aumento do número de *K cluster* acima de 30, pelo menos com o uso do algoritmo de *clustering* adotado aqui (o *Kmeans*), causou uma incidência maior, de retorno do algoritmo, de *clusters* com uma única cidade, o que influencia a métrica de CAPEX negativamente.
- 2) A inclusão da busca local causou um aumento no tempo de execução do método.
- 3) A técnica aqui proposta não prevê trechos com necessidade de uso de fibra marinha.
- 4) O método não tem uma busca especializada para cada um dos objetivos tratados.
- 5) O método não teve seus experimentos realizados em plataformas paralelas.

7.2 Trabalhos Futuros

Como desdobramentos da pesquisa realizada nessa monografia, trabalhos futuros podem ser conduzidos nas seguintes linhas:

- 1) Realização de experimentos com um número maior de *K*, dividindo o estado em outro número de subgrupos para entregar a etapa de busca, por exemplo 25.
- 2) Investigação de outras técnicas de busca local, talvez alguma que trabalhe em focada no cromossomo ao invés de em uma estrutura auxiliar como foi feito nessa pesquisa.
- 3) Inclusão, neste método, de uma pequena probabilidade de mudança de cabeças de *cluster*, sem que haja dominação da nova configuração de rede, para experimentar se há alguma melhora na técnica por evitar ficar preso em ótimos locais.
- 4) Execução de mais iterações ou aumento do tamanho da população, e ou, diminuição do número da carga da rede, para 80 erlangs, por exemplo.
- 5) Elaboração de uma matriz de tráfego com as informações de PIB e População das cidades aqui tratadas, disponíveis em (INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA, 2010).

- 6) Investigação de outras métricas como conectividade natural que é uma métrica de robustez, por exemplo.
- 7) Classificação para os métodos de busca local, talvez pelo seu mecanismo de trabalho.
- 8) Investigação do método com o uso de outro algoritmo de *clustering*, diferente do *Kmeans*.
- 9) Investigação da paralelização da busca local fazendo a paralelização de algoritmos usando arquiteturas multi-core e processamento gráfico.
- 10) Execução da metodologia proposta em outros estudos de casos, como redes de *benchmark* americanas e europeias, que são bem conhecidas na literatura.

Referências

- ALGORITMOS GENETICOS. *Algoritmos Geneticos*. 2017. Disponível em: <<http://www.algoritmosgeneticos.com.br/>>. Acesso em: 10/10/2017.
- ARAÚJO, D. *Design of Optical Networks by Using Computational Intelligence and Network Science*. 2015. Tese (Doutorado).
- ARAÚJO, D.; BASTOS-FILHO, C.; MARTINS FILHO, J. Métricas de Redes Complexas para Análise de Redes Ópticas. *Revista de Tecnologia da Informação e Comunicação*, v. 2, n. 4, p. 11 – 18, Jan 2014.
- ARAÚJO, D. R. B. et al. An efficient multi-objective evolutionary optimizer to design all-optical networks considering physical impairments and CAPEX. In: *2011 11th International Conference on Intelligent Systems Design and Applications IS -* [S.l.: s.n.], 2011. p. 76 – 81.
- BEZERRA, C. M. M.; B., A. D. R.; MACARIO, V. Allocation of Volunteers in Non-governmental Organizations Aided by Non-supervised Learning. In: *2016 5th Brazilian Conference on Intelligent Systems (BRACIS) IS - SN - VO - VL -* [S.l.: s.n.], 2016. p. 223 – 228.
- BRUN, M. et al. *Model-based evaluation of clustering validation measures*. [S.l.: s.n.], 2007. v. 40. (Pattern Recognition, v. 40).
- CHAVES, D. et al. *SIMTON: A simulator for transparent optical networks*. [S.l.: s.n.], 2010. v. 25.
- CHAVES, D. A. R.; BASTOS-FILHO, C. J. A.; MARTINS-FILHO, J. F. Multiobjective physical topology design of all-optical networks considering QoS and Capex. In: *2010 Conference on Optical Fiber Communication (OFC/NFOEC), collocated National Fiber Optic Engineers Conference IS - SN - VO - VL -* [S.l.: s.n.], 2010. p. 1 – 3.
- DEB, K.; JAIN, H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*, v. 18, n. 4, p. 577 – 601, 2014.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182 – 197, 2002.
- DUPLEIX, V.; ARAÚJO, D. R. B.; BASTOS-FILHO, C. A. Designing the optical network of Haiti using a multi-objective evolutionary approach. In: *2016 IEEE Latin American Conference on Computational Intelligence (LA-CCI) IS - SN - VO - VL -* [S.l.: s.n.], 2016. p. 1 – 6.
- DURAN, R. J. et al. Multiobjective Genetic Algorithm to Design Cost-Efficient Wavelength-Routed Optical Networks. In: *2007 9th International Conference on Transparent Optical Networks IS -* [S.l.: s.n.], 2007. v. 4, p. 96 – 99.

- DURÁN, R. J. et al. Advantages of using cognition when solving impairment-aware virtual topology design problems. p. 1 – 4, 2011.
- DURILLO, J. J.; NEBRO, A. J.; ALBA, E. The jMetal framework for multi-objective optimization: Design and architecture. In: *IEEE Congress on Evolutionary Computation IS -* [S.l.: s.n.], 2010. p. 1 – 8.
- EMBRAPA. *Embrapa*. 2017. Disponível em: <<http://www.uep.cnps.embrapa.br/solos/index.php?link=pe>>. Acesso em: 02/11/17.
- FIGUEIREDO, E. M. N. et al. Physical Topology Design of Optical Networks Aided by Many-Objective Optimization Algorithms. In: *2016 5th Brazilian Conference on Intelligent Systems (BRACIS) IS - SN - VO - VL -* [S.l.: s.n.], 2016. p. 409 – 414.
- FUKUNAGA, K.; NARENDRA, P. M. A Branch and Bound Algorithm for Computing k-Nearest Neighbors. *IEEE Transactions on Computers*, C-24, n. 7, p. 750 – 753, 1975.
- GOOGLE. *google maps*. 2016. Disponível em: <<https://www.google.com.br/maps>>. Acesso em: 16/08/16.
- HEDDEGHEM, W. V. et al. Power consumption modeling in optical multilayer networks. *Photonic Network Communications*, v. 24, n. 2, p. 86 – 102, 2012. ISSN 1572-8188.
- HERON, R. W. et al. Technology innovations and architecture solutions for the next-generation optical access network. *Bell Labs Technical Journal*, v. 13, n. 1, p. 163 – 181, 2008.
- INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. *IBGE*. 2010. Disponível em: <<https://cidades.ibge.gov.br/brasil/pe/panorama>>. Acesso em: 15/8/16.
- JAIN, A. K. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, v. 31, n. 8, p. 651 – 666, 2010. ISSN 0167-8655.
- KAUFMANN, L.; ROUSSEEUW, P. *Clustering by Means of Medoids VL -* [S.l.: s.n.], 1987. (Data Analysis based on the L1-Norm and Related Methods).
- KEISER, G. *Wiley Encyclopedia of Telecommunications*. John Wiley & Sons, Inc., 2003. ISBN 9780471219286. Disponível em: <<http://dx.doi.org/10.1002/0471219282.eot158>>.
- LIDEN, R. *Algoritmos Genéticos Uma Importante Ferramenta da Inteligência Computacional*. 2. ed. Rio de Janeiro: Brasport, 2008.
- MACQUEEN, J. Some methods for classification and analysis of multivariate observations. In: . Berkeley, Calif.: University of California Press, 1967. (Fifth Berkeley Symposium on Mathematical Statistics and Probability), p. 281 – 297. ISSN 0097-0433.
- MAHESWARI, D. U.; MEENALOCHANI, M.; SUDHA, S. Influence of the cluster head position on the lifetime of wireless sensor network A case study. In: *2016 2nd International Conference on Next Generation Computing Technologies (NGCT) IS - SN - VO - VL -* [S.l.: s.n.], 2016. p. 378 – 381.

- MONTHA, P.; MANEEKUT, R.; KAEWPLUNG, P. The performance limitation of 10-Gbps-per-channel-based coarse wavelength division multiplexed passive optical network. In: *16th International Conference on Advanced Communication Technology IS - [S.l.: s.n.]*, 2014. p. 1089 – 1092.
- N. FERNÁNDEZ et al. Cognitive genetic algorithms to design impairment-aware virtual topologies in optical networks. In: *OFC/NFOEC IS - [S.l.: s.n.]*, 2012. p. 1 – 3.
- NASCIMENTO, J. C. et al. Estudo de Expansão da Malha de Fibra Óptica de Pernambuco considerando Resiliência e Custo. In: *Jornada de Ensino Pesquisa e Extensão (JEPEX), XVI*. Recife: [s.n.], 2016. Resumos... Universidade Federal Rural de Pernambuco, p. Disponível em: <http://www.eventosufrpe.com.br/2016/anais/lista_area_01.htm>. Acesso em: 17/06/2017.
- NEBRO, A. J.; DURILLO, J. J.; VERGNE, M. Redesigning the jMetal Multi-Objective Optimization Framework. In: *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2015. (GECCO Companion '15), p. 1093 – 1100. ISBN 978-1-4503-3488-4. Disponível em: <<http://doi.acm.org/10.1145/2739482.2768462>>.
- NGUYEN, H. V.; TRUONG, D. L. Survivable physical topology design for all-optical metro core networks. In: *2013 International Conference on Computing, Management and Telecommunications (ComManTel) IS - SN - VO - VL - [S.l.: s.n.]*, 2013. p. 38 – 42.
- PAVAN, C. et al. How reliable are the real-world optical transport networks? *IEEE/OSA Journal of Optical Communications and Networking*, v. 7, n. 6, p. 578 – 585, 2015.
- RAMASWAMI, R.; SIVARAJAN, K. N.; SASAKI, G. H. *Optical Networks (Third Edition)*. Boston: Morgan Kaufmann, 2010. ISBN 978-0-12-374092-2.
- RIBEIRO, J. A.; FARIAS, O. L. M. de; ROQUE, L. A. O. L. A syntactic and lexicon analyzer for the geography markup language (GML). In: *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium IS - SN - [S.l.: s.n.]*, 2004. v. 5, p. 2896–2899 – vol.5.
- S, N. V. A Query Language for Information Retrieval from GML. In: *2010 International Conference on Recent Trends in Information, Telecommunication and Computing IS - SN - VO - VL - [S.l.: s.n.]*, 2010. p. 372 – 374.
- SRINIVAS, N.; DEB, K. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, v. 2, n. 3, p. 221 – 248, 1994.
- TRAN, A. V.; CHAE, C. j.; TUCKER, R. S. Ethernet PON or WDM PON: A Comparison of Cost and Reliability. In: *TENCON 2005 - 2005 IEEE Region 10 Conference IS - [S.l.: s.n.]*, 2005. p. 1 – 6.
- YOUNIS, O.; KRUNZ, M.; RAMASUBRAMANIAN, S. Node clustering in wireless sensor networks: recent developments and deployment challenges. *IEEE Network*, v. 20, n. 3, p. 20 – 25, 2006.
- ZITZLER, E. et al. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, v. 7, n. 2, p. 117 – 132, 2003.