



UNIVERSIDADE FEDERAL DO TOCANTINS
CAMPUS UNIVERSITÁRIO DE PALMAS
CURSO DE CIÊNCIA DA COMPUTAÇÃO

UMA HEURÍSTICA BASEADA EM OTIMIZAÇÃO DE COLÔNIA DE
ABELHAS ARTIFICIAL PARA O PROBLEMA DA ÁRVORE DE
STEINER EUCLIDIANO

THIAGO GONÇALVES FEITOSA

PALMAS (TO)

2018

THIAGO GONÇALVES FEITOSA

UMA HEURÍSTICA BASEADA EM OTIMIZAÇÃO DE COLÔNIA DE ABELHAS
ARTIFICIAL PARA O PROBLEMA DA ÁRVORE DE STEINER EUCLIDIANO

Trabalho de Conclusão de Curso II apresentado
à Universidade Federal do Tocantins para
obtenção do título de Bacharel em Ciência da
Computação, sob a orientação do(a) Prof.(a)
Dr. Marcelo Lisboa Rocha.

Orientador: Dr. Marcelo Lisboa Rocha

PALMAS (TO)

2018

THIAGO GONÇALVES FEITOSA

UMA HEURÍSTICA BASEADA EM OTIMIZAÇÃO DE COLÔNIA DE ABELHAS
ARTIFICIAL PARA O PROBLEMA DA ÁRVORE DE STEINER EUCLIDIANO

Trabalho de Conclusão de Curso II apresentado à UFT – Universidade Federal do Tocantins – Campus Universitário de Palmas, Curso de Ciência da Computação foi avaliado para a obtenção do título de Bacharel e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Data de aprovação: 10 / 12 / 2018

Banca Examinadora:

Prof. Nome do Primeiro Examinador Sobrenome

Prof. Nome do Segundo Examinador Sobrenome

Prof. Nome do Terceiro Examinador Sobrenome

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da Universidade Federal do Tocantins

- F311h Feitosa, Thiago Gonçalves.
Uma Heurística Baseada em Otimização de Colônia de Abelhas Artificial para o Problema da Árvore de Steiner Euclidiano. / Thiago Gonçalves Feitosa. – Palmas, TO, 2018.
40 f.
- Monografia Graduação - Universidade Federal do Tocantins – Câmpus Universitário de Palmas - Curso de Ciências da Computação, 2018.
Orientador: Marcelo Lisboa Rocha
1. Árvore de Steiner. 2. Otimização Combinatória. 3. Colônia de Abelhas Artificial. 4. Árvore Mínima Geradora. I. Título

CDD 004

TODOS OS DIREITOS RESERVADOS – A reprodução total ou parcial, de qualquer forma ou por qualquer meio deste documento é autorizado desde que citada a fonte. A violação dos direitos do autor (Lei nº 9.610/98) é crime estabelecido pelo artigo 184 do Código Penal.

Elaborado pelo sistema de geração automática de ficha catalográfica da UFT com os dados fornecidos pelo(a) autor(a).

*A todos aqueles que se dedicaram
a descobrir e modificar o mundo
de forma benéfica aos seres que
aqui habitam.*

AGRADECIMENTOS

Gostaria de agradecer ao Criador e toda a espiritualidade pelo presente que é a vida e pela oportunidade de uma evolução moral e espiritual. Agradeço aos meus pais Elza Maria das Chagas Feitosa e Luiz Gonçalves Feitosa e a minha irmã Amanda Gonçalves Feitosa pelo apoio que se estende a várias áreas da minha vida, ao companheirismo e fidelidade desde antes de eu compreender a minha existência. Agradeço também minha leal namorada Larissa Cristina Oliveira de Souza que me acompanhou nessa batalha árdua que é a criação de um trabalho científico, pela companhia nas noites em claro, nas leituras de artigos e escrita de código, também pelos maravilhosos momentos que passei ao seu lado dentro e fora da universidade. Minha companheira de aventuras. Não poderia de forma alguma deixar de citar meu orientador Professor Doutor Marcelo Lisboa Rocha pela dedicação e ajuda que só um pai teria com seu filho. Sua empolgação e determinação me motivaram muito durante o nosso trabalho e continuam motivando e me inspirando a conquistar mais, assim como conquistou em sua vida e segue conquistando. Por fim agradeço a todos os meus amigos dentro e fora da universidade, os quais sempre me apoiaram em boas escolhas, estiveram comigo nos momentos felizes e me ajudaram a carregar os fardos que surgiram em minha vida.

Todos vocês mencionados aqui direta ou indiretamente, sintam-se abraçados. Sem vocês eu não seria nada.

RESUMO

Este trabalho consiste em propor uma abordagem que propicie um melhor desempenho comparado com técnicas já existentes na literatura sobre a resolução de problemas de Otimização Combinatória, mais precisamente para o Problema da Árvore de Steiner Euclidiano no R^n , fazendo uso de Programação Linear Inteira, com resolução através da ferramenta GLPK e do algoritmo de Colônia de Abelhas Artificial somado com algoritmo de geração de coordenadas quasi-random de baixa discrepância.

Palavra-chave: L^AT_EX. U^FT_EX. Árvore de Steiner. Árvore Mínima Geradora. Otimização Combinatória. Colônia de Abelhas Artificial.

ABSTRACT

This work consists in proposing an approach that provides a better performance compared to existing techniques in the scientific literature on the problem solving of Combinatorial Optimization, more precisely to the Euclidean Steiner Tree Problem in R^n , making use of integer linear programming, with resolution through the GLPK tool and the Artificial Bees Colony algorithm added with algorithm of generation of quasi-random coordinates of low discrepancy.

Keywords: L^AT_EX. U^FT_EX. Steiner Tree. Minimum Spanning Tree. Combinatorial optimization. Artificial Bee Colony.

LISTA DE FIGURAS

Figura 2.1 – Árvore Steiner no R^2 com 4 pontos fixos (1, 2, 3 e 4) e 2 pontos de Steiner (5 e 6).	13
Figura 2.2 – Exemplo de AGMs e ASMs com 3 e 4 pontos fixos (em preto) com cálculo de custo (Z) em que as ASMs contam respectivamente com 1 e 2 Pontos de Steiner (em vermelho).	16
Figura 2.3 – Árvore evolucionária para 19 espécies.	20
Figura 2.4 – Hélice tripla do DNA com uma Árvore de Steiner Mínima para $P=20$	21
Figura 2.5 – Rede de dutos de ventilação	21
Figura 3.1 – Fluxograma do algoritmo de Colônia de Abelhas Artificial	25
Figura 3.2 – Fluxograma dos estágios para a resolução do PASE utilizando a abordagem desse trabalho	26
Figura 4.1 – Exemplo 1 de posicionamento de Pontos Fixos	28
Figura 4.2 – Exemplo 1 de posicionamento de Pontos Fixos e Pontos de Steiner	28
Figura 4.3 – Exemplo 1 de execução do algoritmo ABC	29
Figura 4.4 – Exemplo 2 de posicionamento de Pontos Fixos	30
Figura 4.5 – Exemplo 2 de posicionamento de Pontos Fixos e Pontos de Steiner	30
Figura 4.6 – Exemplo 2 de execução do algoritmo ABC	31

LISTA DE TABELAS

Tabela 3.1 – Matriz de restrição para o PASE com $P=4$ e $S=2$	23
Tabela 4.1 – Tabela de comparativo entre as abordagens ABC e Wsmith para o PASE	33
Tabela 4.2 – Tabela de comparativo entre as abordagens ABC e GRASP para o PASE	33

SUMÁRIO

1	INTRODUÇÃO	12
2	PROBLEMA DA ÁRVORE DE STEINER EUCLIDIANO . . .	13
2.1	Grafo	14
2.2	Árvore Geradora Mínima	14
2.3	Formulação Matemática para o PASE no R^n	16
2.4	Aplicações Práticas do Problema da Árvore de Steiner Euclidiana	19
2.4.1	Aplicação do PASE na Inferência Filogenética	19
2.4.2	Aplicação do PASE na Configuração Molecular	19
2.4.3	Aplicação do PASE no Projeto de Redes	20
3	MÉTODOS	22
3.1	Programação Linear	22
3.1.1	Programação Linear Inteira	22
3.2	GLPK	23
3.3	Sequência de Sobol	23
3.4	Normalização	24
3.5	Colônia de Abelhas Artificial	24
3.5.1	Descrição do Algoritmo	25
4	RESULTADOS	27
4.1	Comparativos	31
4.1.1	Wsmith	32
4.1.2	GRASP	32
5	CONSIDERAÇÕES FINAIS	34
6	TRABALHOS FUTUROS	35

REFERÊNCIAS	36
A ROTINAS DO PROGRAMA SOBOL_DATASET	38

1 INTRODUÇÃO

Muitos problemas práticos do mundo real podem ser representados como um grafo como, por exemplo, ligações em circuitos elétricos, caminhos entre cidades, túneis de minas conexões de dutos de ar, conexões de redes de computadores, dentre outros. Muitos deles são problemas bastante conhecidos e explorados na literatura que envolvem grafos e suas aplicações como Problema da Coloração de Grafos, Problema do Caminho Mínimo, Problema do Caixeiro Viajante, Problema da Árvore Geradora Mínima, Problema do Caminho Hamiltoniano, dentre vários outros. Bose et al. (2001) West et al. (2001) Narula e Ho (1980)

Diversas heurísticas e meta-heurísticas foram aplicadas nesses problemas, levando a abordagens diferentes com resultados ótimos, porém nem sempre com bons tempos computacionais. Buscou-se nesse trabalho uma otimização de abordagens já existentes e desenvolvimento de uma estratégia para a resolução do Problema da Árvore de Steiner Euclidiano esperando uma melhora no tempo computacional comparado com trabalhos já executados, essa comparação foi obtida através de medição de tempo computacional (CPU) executando-se os algoritmos futuramente descritos, também foi esperado uma melhora em relação ao custo da solução final para cada problema levando em consideração o número de Pontos Fixos e a dimensão do espaço euclidiano. comparações foram feitas com os trabalhos que serviram como referência para desenvolvimento desse estudo, além de outros.

Utilizando a ferramenta GLPK que facilita em encontrar a solução para problemas de Programação Linear, Programação inteira, dentre outros, a qual nesse trabalho foi usada para resolver o Problema de Programação Linear Inteira formulado e detalhado em outros capítulos.

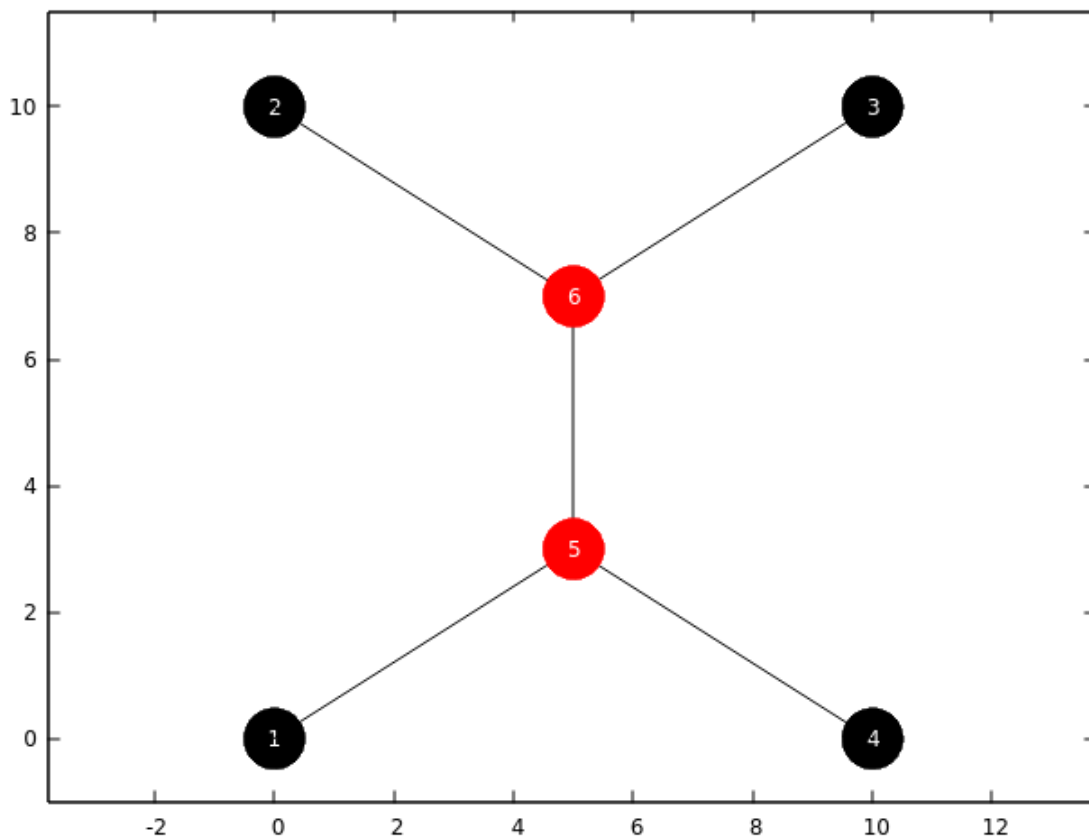
Consta ainda nesse trabalho a descrição do método bio-inspirado chamado Colônia de Abelhas Artificial. Método baseado no comportamento de indivíduos de uma colônia de abelhas na busca de melhores espaços para exploração de alimento com base em diversos fatores, também de forma adicional foi utilizada um algoritmo de geração de sequência de SOBOL juntamente com uma forma de normalização de seus resultados para diminuir a discrepância entre as coordenadas geradas inicialmente e manter dentro do limite necessário para o algoritmo da Colônia de Abelhas Artificial.

2 PROBLEMA DA ÁRVORE DE STEINER EUCLIDIANO

Neste capítulo é feita uma abordagem acerca de diversos aspectos a respeito do Problema da Árvore de Steiner Euclidiano (PASE) onde está incluso a descrição do problema, formulação matemática, soluções possíveis e algumas aplicações práticas encontradas na literatura científica.

O Problema da Árvore de Steiner Euclidiano consiste em encontrar uma árvore com o custo mínimo para conexões entre seus pontos, dados P pontos ($p_1, p_2, p_3, \dots, p_{P-1}, p_P$) no R^n com métrica Euclidiana, utilizando ou não pontos adicionais que auxiliam na minimização do caminho, chamados Pontos de Steiner, os quais devem haver em número igual a $P-2$ como citado por (ROCHA, 2008), se mostra na Figura 3.1 e que serão abordados com mais detalhes no decorrer deste trabalho.

Figura 2.1 – Árvore Steiner no R^2 com 4 pontos fixos (1, 2, 3 e 4) e 2 pontos de Steiner (5 e 6).



2.1 Grafo

Um grafo G é uma tripla ordenada $(V(G), E(G), \psi(G))$ que consiste basicamente de um conjunto não-vazio $V(G)$ de vértices, um conjunto $E(G)$, disjunto de $V(G)$, de arestas, e uma função de incidência ψ que associa cada aresta de G a um par não ordenado de vértices (não necessariamente distintos) de G . Se e é uma aresta e u e t são vértices tais que $\psi_G(e) = uv$, então é dito que u e v estão unidos pela aresta e . Os vértices u e v são chamados de extremidades de e . Bondy, Murty et al. (1976)

A seguir temos um exemplo de grafo para servir de esclarecimento acerca da definição.

Exemplo 1:

$$G = (V(G), E(G), \psi_G)$$

Onde:

$$\begin{aligned} V(G) &= \{v1, v2, v3, v4, v5\} \\ E(G) &= \{e1, e2, e3, e4, e5, e6, e7, e8\} \end{aligned}$$

E ψ_G é definido por:

$$\begin{aligned} \psi_G(e1) &= v1v2, \psi_G(e2) = v2v3, \psi_G(e3) = v3v3, \psi_G(e4) = v3v4 \\ \psi_G(e5) &= v2v4, \psi_G(e6) = v4v5, \psi_G(e7) = v2v5, \psi_G(e8) = v2v5 \end{aligned}$$

2.2 Árvore Geradora Mínima

A Árvore Geradora Mínima (AGM) é uma árvore de extensão de um grafo não direcionado e ponderado, um subgrafo, de modo que o seu custo total é minimizado com a escolha das arestas de menor custo possíveis de forma que todos os vértices possam ser alcançados entre si. Entenda-se por custo total a soma dos custos de cada aresta utilizada na AGM. Essa Árvore deve ter custo total menor ou igual a qualquer outra Árvore possível no grafo.

Árvore Geradora Mínima com Restrição de Graus pode ser definida formalmente obtendo como entrada um grafo completo $G=(V,E)$ com arestas não negativas e uma restrição de $d \geq 2$ e como saída uma Árvore Geradora de G com um custo mínimo de modo que o grau de cada aresta seja na árvore seja $\leq d$. Bui e Zrncic (2006), Amarir, Zegrari e Idrissi (2016)

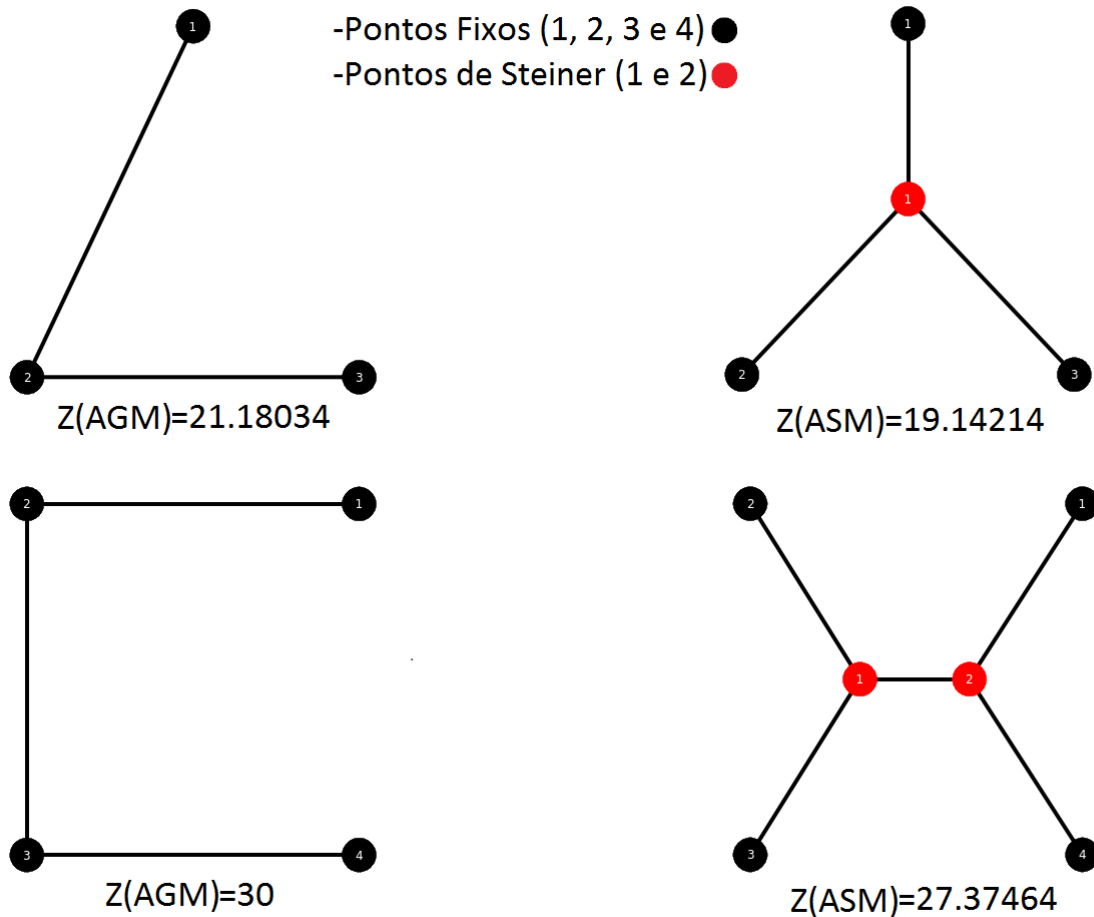
Uma solução do PASE com custo mínimo chamada de Árvore de Steiner Mínima (ASM) deve obedecer as seguintes regras que são essenciais para o entendimento do problema e cria uma base para a solução abordada nesse trabalho:

- Um Ponto Fixo em uma Árvore Mínima de Steiner deve ter grau (valência) igual a 1;
- Um Ponto de Steiner em uma Árvore Mínima de Steiner deve ter grau (valência) igual a 3;
- O grau formado pelas arestas de um ponto de Steiner e seus pontos adjacentes deve ser de 120° exatamente;
- Uma Árvore Mínima de Steiner para um problema com P pontos, deve conter P-2 Pontos de Steiner;
- Todos os Pontos de Steiner devem ter no mínimo 1 ligação com pontos fixos;

A figura 3.2 representa AGMs e ASMs, havendo dois exemplos de cada, com $P=3$ e $P=4$ que é o número de pontos fixos onde $Z(AGM)$ e $Z(ASM)$ representam respectivamente o custo total da Árvore Geradora Mínima e da Árvore de Steiner Mínima. As coordenadas dos pontos utilizados foram: Na AGM e ASM de 3 pontos, Pontos Fixos $P1=\{5,10\}$, $P2=\{0,0\}$ e $P3=\{10,0\}$, Pontos de Steiner $P1=\{5,5\}$; Na AGM e ASM de 4 pontos, Pontos Fixos $P1=\{10,10\}$, $P2=\{0,10\}$, $P3=\{0,0\}$ e $P4=\{10,0\}$, Pontos de Steiner $P1=\{3.3,5\}$ e $P2=\{6.6,5\}$.

Para o problema das Árvores Geradoras Mínimas se encontra facilmente soluções em tempo polinomial, mas no caso das Árvores de Steiner Mínimas devido a possibilidade de adição de pontos intermediários e com a restrição de graus este é um problema NP-Difícil, portanto boas heurísticas de tempo polinomial são desejáveis e mais difíceis de alcançar. E já que o Problema da Árvore Geradora Mínima com essa restrição de graus descrita anteriormente é um problema NP-Difícil e tão complicado quanto o próprio problema do PASE não nos é interessante utilizá-la como ferramenta para a resolução, já que o objetivo é encontrar uma solução ótima, ou seja, a melhor solução possível. (DREYER; OVERTON, 2002), (BUI; ZRNCIC, 2006), Santuari (2003)

Figura 2.2 – Exemplo de AGMs e ASMs com 3 e 4 pontos fixos (em preto) com cálculo de custo (Z) em que as ASMs contam respectivamente com 1 e 2 Pontos de Steiner (em vermelho).



2.3 Formulação Matemática para o PASE no R^n

Nesta seção será detalhado o modelo matemático para o Problema da Árvore de Steiner Euclidiano no R^n . Dado um grafo $G=(V,E)$, onde N é o número de dimensões do espaço Euclidiano onde o grafo está sendo representado, $P=\{1, 2, \dots, p-1, p\}$ é o conjunto de índices os quais são associados aos pontos Pontos Fixos $x_1, x_2, \dots, x_{p-1}, x_p$, e sendo $S=\{p+1, p+2, \dots, 2p-3, 2p-2\}$ o conjunto de índices os quais são associados aos Pontos de Steiner $x_{p+1}, x_{p+2}, \dots, x_{2p-3}, x_{2p-2}$, temos com a união dos pontos de Steiner com os Pontos Fixos o número de vértices a Árvore de Steiner $V = P \cup S$. Uma aresta de G nesse caso pode ser denotada por $[i, j]$, sendo i e j dois pontos quaisquer pertencentes a V . Devemos também denotar os conjuntos $E_1 = \{[i, j] | i \in P, j \in S\}$, este referente ao conjunto de arestas que ligam um Ponto Fixo qualquer a um Pontos de Steiner qualquer e $E_2 = \{[i, j] | i \in S, j \in S\}$, este referente ao conjunto das arestas que ligam um Ponto de Steiner qualquer com um outro Ponto de Steiner qualquer de forma que $E = E_1 \cup E_2$. Rocha (2008) Wu e Chao (2004)

A denotação da distância Euclidiana entre os pontos x_i e x_j é representada por $distancia_{i,j} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$, logo podemos definir um $y_{i,j} \in \{0, 1\} \forall [i, j] \in E$, onde $y_{i,j}$ recebe valor igual a 1 se a aresta pertencente a E referente aos pontos x_i e x_j está contida na solução ótima da Árvore de Steiner e $y_{i,j}$ recebe valor igual a 0 no caso contrário. A seguir teremos a formulação matemática do PASE o R^n com base nas denotações anteriores. Maculan, Michelon e Xavier (2000)

$$\text{Min} \sum_{i=1}^p \sum_{j=1}^q d_{i,j} x_{i,j} + \sum_{i=p+1}^q \sum_{j=i+1}^q d_{i,j} x_{i,j} \quad (1)$$

$$x_j \in R^n, j = p + 1, \dots, 2p - 2 \quad (2)$$

$$\sum_{j=p+1}^q x_{i,j} = 1, \forall i \in P \quad (3)$$

$$\sum_{i=1}^s x_{i,j} \sum_{k=p, k \neq j}^q x_{k,j} = 3, \forall j \in S \quad (4)$$

$$x_{i,j} \in \{0, 1\} \text{ e } [i, j] \in E \quad (5)$$

O item (1) da formulação refere-se à função objetivo do problema, onde está incluso a minimização do custo das ligações entre todos os Pontos Fixos com Pontos de Steiner, tato quanto as ligações dos Pontos de Steiner entre si. Analisando o item referente à descrição (2) podemos observar que x_j é um Ponto de Steiner. As restrições do item (3) são utilizadas para controlar o grau (valência) dos Pontos Fixos modo que seja de 1 enquanto as restrições do item (4) são utilizadas para controlar o grau (valência) dos Pontos de Steiner de modo que seja igual a 3 e evita que hajam arestas referentes a um ponto de Steiner ligado a si mesmo e por fim o item (5) limita os valores de $x_{i,j}$ pertencentes ao conjunto $\{0, 1\}$ de modo que esse valor designe se a aresta está contida, valor 1, ou não, valor 0, na solução.

A seguir temos um exemplo da formulação matemática para o caso de 4 Pontos Fixos com as coordenadas $P=(1\{0,0\}, 2\{0,10\}, 3\{10,10\}, 4\{10,0\})$ e Pontos de Steiner $S=(5\{4.998855, 2.886471\}, 6\{4.998306, 7.112765\})$ exatamente da forma que ela é gerada e gravada em arquivo de formato .lp pelas rotinas da ferramenta GLPK:

Gera-se a função objetivo obtida a partir de todas as possibilidades de arestas do problema, com seus respectivos custos e informação dos pontos a aresta une, dados todos os Pontos Fixos e Pontos de Steiner.

$$+ 5.7723708152771 \text{ x1i5} + 8.69335842132568 \text{ x1i6} + 8.69429969787598 \text{ x2i5} \\ + 5.77227735519409 \text{ x2i6} + 8.69561576843262 \text{ x3i5} + 5.77521181106567 \text{ x3i6}$$

$$+ 5.77435398101807 x_{4i5} + 8.6953067779541 x_{4i6} + 4.22629308700562 x_{5i6}$$

Logo após temos as restrições, onde os itens p1, p2, p3 e p4 correspondem a todas as possibilidades de ligação dos Pontos Fixos 1, 2, 3 e 4, respectivamente e assegura que apenas uma ligação será feita para esses pontos. Os itens k6 e k7 representam todas as possibilidades de ligação dos Pontos de Steiner 5 e 6 respectivamente.

$$\mathbf{p1: + x_{1i6} + x_{1i5} = 1}$$

$$\mathbf{p2: + x_{2i6} + x_{2i5} = 1}$$

$$\mathbf{p3: + x_{3i6} + x_{3i5} = 1}$$

$$\mathbf{p4: + x_{4i6} + x_{4i5} = 1}$$

$$\mathbf{q5: + x_{5i6} = 1}$$

$$\mathbf{k6: + x_{5i6} + x_{4i5} + x_{3i5} + x_{2i5} + x_{1i5} = 3}$$

$$\mathbf{k7: + x_{5i6} + x_{4i6} + x_{3i6} + x_{2i6} + x_{1i6} = 3}$$

Seguindo na formulação matemática temos os limites onde se garante os valores 0 ou 1 para as variáveis correspondentes a todas as arestas $x_{i,j}$ possíveis na árvore. Onde essas variáveis vão receber valor igual a 1 se forem utilizadas na solução final ou valor 0 se não forem utilizadas.

$$0 \leq x_{1i5} \leq 1$$

$$0 \leq x_{1i6} \leq 1$$

$$0 \leq x_{2i5} \leq 1$$

$$0 \leq x_{2i6} \leq 1$$

$$0 \leq x_{3i5} \leq 1$$

$$0 \leq x_{3i6} \leq 1$$

$$0 \leq x_{4i5} \leq 1$$

$$0 \leq x_{4i6} \leq 1$$

$$0 \leq x_{5i6} \leq 1$$

Por último mostramos a lista de arestas $x_{i,j}$, contendo todas as arestas possíveis do problema em questão.

x_{1i5}

x_{1i6}

x_{2i5}

x_{2i6}

x_{3i5}

x_{3i6}

x_{4i5}

x_{4i6}

x_{5i6}

A partir da formulação matemática do PASE no R^n podemos observar que cada possível solução considerando as restrições anteriores está associada a um subgrafo conexo de G que tem como característica em cada Ponto Fixo grau igual a 1 e em cada Ponto de Steiner grau igual a 3, logo o número de arestas $x_{i,j}$ com valores iguais a 1 é de $2P - 3$. Desta forma cada solução vai definir uma Árvore Geradora do grafo G .

2.4 Aplicações Práticas do Problema da Árvore de Steiner Euclidiana

Diversos problemas práticos reais podem empregar o problema da árvore de Steiner Euclidiana (PASE) no R^2 e no R^3 como estrutura teórica em sua resolução, logo, resolver eficientemente o problema em questão, levará à solução dos problemas que podem ser modelados como o PASE.

Algumas aplicações que permitem ser modeladas como um PASE são: configuração molecular (bioquímica), projeto de redes (elétrica, linhas de metrô, dutos, etc.) e também problemas gerais de mineração de dados, como inferência filogenética. Montenegro, Torreão e Maculan (2003) A seguir, algumas destas aplicações serão apresentadas em maiores detalhes.

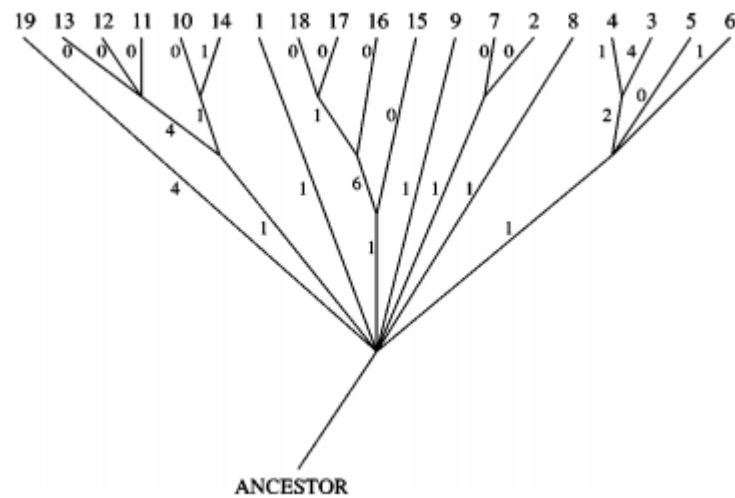
2.4.1 Aplicação do PASE na Inferência Filogenética

No trabalho Microcanonical optimization algorithm for the Euclidean Steiner problem in R^n with application to phylogenetic inference Montenegro, Torreão e Maculan (2003), foi introduzida uma abordagem heurística para o PASE, usando uma meta heurística chamada algoritmo de otimização microcanônica, que explora o paralelo entre sistemas de física estatística e problemas de otimização de alta dimensionalidade, de forma semelhante à do trabalho de Kirkpatrick, Gelatt e Vecchi (1983) que foi pioneiro nessa área de pesquisa. O trabalho mostrado como exemplo utiliza uma heurística que realiza uma pesquisa local sobre o espaço dos vetores de descrição de topologia que resultam do esquema enumerativo descrito e referenciado em seu artigo e, como resultado, produziu boas soluções, para instâncias de 50 pontos dados em 50 dimensões. como o desempenho desse algoritmo aumenta proporcionalmente de acordo com o número de dimensões, foi admitido que pode fornecer uma ferramenta adequada para problemas de agrupamento (clustering) de alta dimensionalidade, incluindo os de inferência filogenética, da qual uma instância é tratada por eles. A seguir, na **Figura 2.3**, tem-se um exemplo de como a estrutura de árvore de Steiner euclidiana bidimensional aparece em algumas árvores de inferência filogenética.

2.4.2 Aplicação do PASE na Configuração Molecular

Mcgregor Smith apresentou em seus trabalhos várias propriedades que foram generalizadas do espaço euclidiano bidimensional para o tridimensional, e ainda metodologias

Figura 2.3 – Árvore evolucionária para 19 espécies.(MONTENEGRO; TORREÃO; MACULAN, 2003)

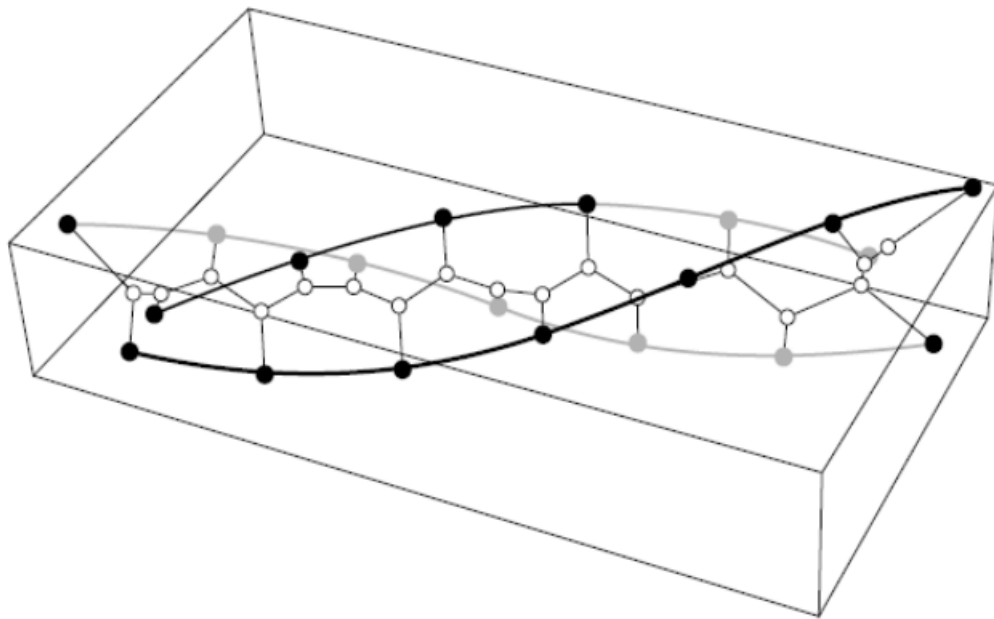


algorítmicas que, devido à importância do problema de Steiner demonstrada na literatura científica, podem ser utilizadas em diversas áreas de aplicações. Uma das aplicações mais interessantes é aquela relacionada ao Problema de Configuração de Energia Mínima, descrita anteriormente em outros artigos, e enfatizada por McGregor Smith em Smith (1998), que proporcionam boas ideias para o trabalho do Problema de Enovelamento de Proteínas e DNA, sendo muito importante para vários ramos das Ciências na atualidade, principalmente aquelas que buscam a produção de drogas para curas de doenças como Câncer. Oliveira (2005) propôs em sua tese de doutorado um algoritmo denominado APOLONIO, baseado no algoritmo de Smith (1992), como também realizou um estudo do problema de árvore mínima de Steiner euclidiana tridimensional (PASET), relacionando a função razão de Steiner ao problema de configuração de energia mínima de macromoléculas de interesse biológico. A seguir, na **Figura 2.4**, tem-se exemplo de como a estrutura de árvore de Steiner euclidiana tridimensional aparece em algumas configurações moleculares enquanto na **Figura 2.5** mostra se a ilustração de uma rede de dutos de ventilação.

2.4.3 Aplicação do PASE no Projeto de Redes

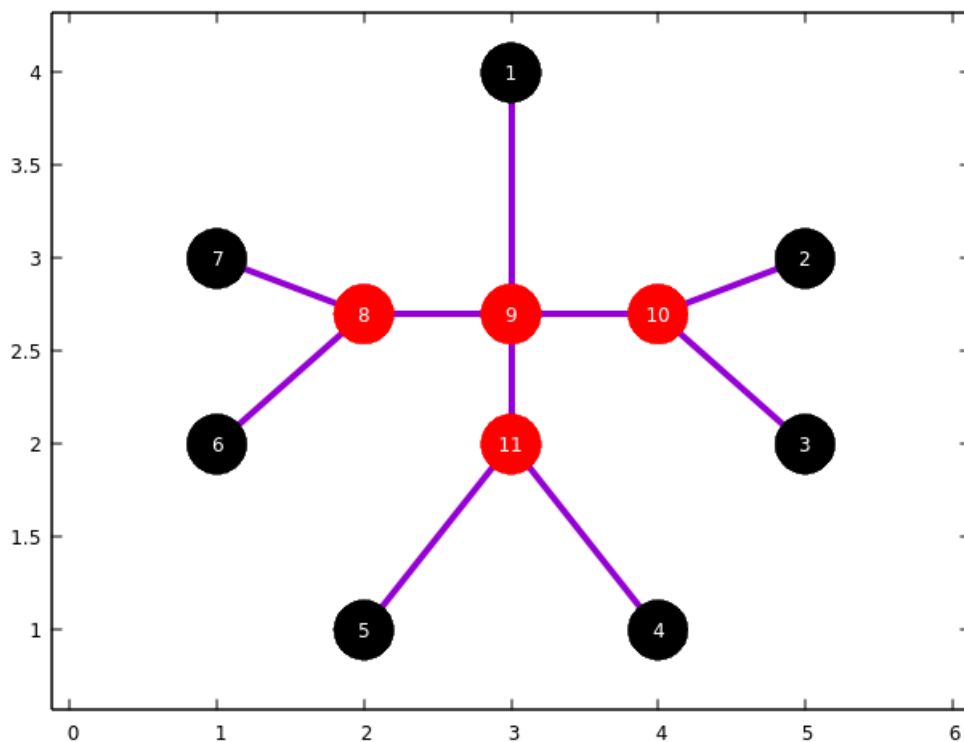
O projeto dos mais variados tipos de rede (de computadores, minas, dutos de ventilação, etc) com comprimento mínimo é de enorme importância e tem grande impacto econômico em diversas situações. No caso da necessidade de realização do projeto de uma rede de tamanho mínimo em uma estrutura tridimensional, pode-se criar um modelo baseado no problema da árvore de Steiner euclidiano tridimensional (PASET), onde os pontos fixos são os pontos da rede que se deseja conectar. Assim sendo, resolvendo-se o PASET, obtém-se a rede de custo mínimo que interliga estes pontos fixos, fazendo uso, se necessário de pontos extras de conexão (pontos de Steiner). Na Figura 3.5, tem-se um

Figura 2.4 – Hélice tripla do DNA com uma Árvore de Steiner Mínima para $P=20$ (SMITH, 1998)



exemplo de uma rede de dutos de ventilação, onde os círculos pretos são os pontos fixos, os círculos vermelhos são os pontos de Steiner e as arestas em roxo são as tubulações.

Figura 2.5 – Rede de dutos de ventilação



3 MÉTODOS

Este capítulo apresenta a estratégia em que foi tratado o PASE durante este trabalho, sendo Programação Linear Inteira, bem como a abordagem e descrição da ferramenta utilizada (GLPK), uma ferramenta de geração de sequência de sobol (SOBOL_DATASET) e um algoritmo de otimização baseados em enxame (ABC).

3.1 Programação Linear

A Programação Linear, um dos recursos da Pesquisa Operacional, busca encontrar soluções ótimas para problemas onde os modelos podem ser representados por expressões lineares, o qual é um dos modelos considerados mais comuns. Marins (2011)

A forma utilizada é, principalmente, encontrar a solução ótima com base na maximização ou minimização da Função Objetivo, estruturando em um sistema de inequações e equações. Como um modelo de otimização, o modelo de programação linear vai apresentar as seguintes características: Goldbarg e Luna (2005)

- Proporcionalidade: a quantidade de recurso a ser consumido em uma determinada atividade deve ser sempre proporcional ao nível da mesma no final do problema e o custo final dela deve ser proporcional ao nível de operação da atividade;
- Não Negatividade: Se necessita sempre a possibilidade de desenvolver as atividades em um nível não negativo qualquer e dado um recurso, obrigatoriamente, deve ser possível ele ser utilizado em qualquer proporção.
- Aditividade: O custo total resulta na soma das parcelas totais de cada uma das atividades;
- Separabilidade: Sempre será possível separar o custo ou o recurso das operações de uma atividade.

3.1.1 Programação Linear Inteira

Um dos tipos de modelo de Programação Linear é a Programação Inteira. Suas soluções são limitadas a um conjunto discreto finito de números, no caso o conjunto de número inteiros \mathbb{Z} . Como a Programação inteira, também chamada de Programação Linear inteira, é um tipo de Programação Linear, ela tem sua formulação do modelo muito semelhante a da Programação Linear onde a diferença se dá nas restrições das variáveis. Goldbarg e Luna (2005)

No processo de modelagem da Função de Minimização da Função Objetivo referente ao PASE é que as variáveis são binárias e portanto é um problema de Programação Linear Inteira.

3.2 GLPK

O GLPK (GNU Linear Programming Kit) uma ferramenta que consiste em um conjunto de rotinas escritas em linguagem de programação ANSI C e organizada de forma que funcione como uma biblioteca que geralmente é utilizada para resolução de problemas de Programação Linear, como no caso deste trabalho, Programação Inteira, dentre outros. Mais precisamente para para este caso de Programação Linear Inteira que é solucionar o problema da AGM com restrição de graus para o PASE assumimos a formulação matemática anterior, a qual pretende-se minimizar o custo da função ojetivo. As variáveis auxiliares chamadas na formulação de $x_{i,j}$ são utilizadas como variáveis auxiliares, Z é a Função Objetivo e $d_{i,j}$ são coeficientes objetivos. As variáveis auxiliares pode ser chamadas também de linhas, pois elas correspondem às linhas da matriz de restrição, da mesma forma as variáveis estruturais podem ser chamadas de colunas pois elas correspondem às colunas da matriz de restrição. (MAKHORIN, 2010)

A seguir um exemplo de matriz de restrição para PASE com $P=4$ e $S=2$ onde mostra-se todas as possibilidades de ligação entre Pontos Fixos e Pontos de Steiner, no caso do exemplo x_{1i5} , x_{1i6} , x_{2i5} , x_{2i6} , x_{3i5} , x_{3i6} , x_{4i5} e x_{4i6} , e ainda entre Pontos de Steiner com outros Pontos de Steiner, x_{5i6} . Onde as linhas representadas por 1, 2, 3 e 4 referem-se aos Pontos Fixos e as linhas 5 e 6 representam os Pontos de Steiner.

Tabela 3.1 – Matriz de restrição para o PASE com $P=4$ e $S=2$

	x_{15}	x_{16}	x_{25}	x_{26}	x_{35}	x_{36}	x_{45}	x_{46}	x_{56}
1	1	1	0	0	0	0	0	0	0
2	0	0	1	1	0	0	0	0	0
3	0	0	0	0	1	1	0	0	0
4	0	0	0	0	0	0	1	1	0
5	1	0	1	0	1	0	1	0	1
6	0	1	0	1	0	1	0	1	1

3.3 Sequência de Sobol

Foi desejado gerar nesse trabalho um conjunto de pontos x_1, x_2, \dots, x_n bem distribuídos no espaço N-dimensional visando os benefícios em questão de tempo computacional que levaria adotar essa boa distribuição como primeira entrada da população de abelhas do algoritmo ABC. Uma medida utilizada para qualificar essa distribuição é a discrepância, definida em Bratley e Fox (1988).

Quanto menor a discrepância melhor o espaçamento. Acredita-se, embora não tenha sido provado ainda, que nenhum espaço N -dimensional infinito pode ter discrepância em ordem de magnitude menor que $\log N$. Sequências que visam baixa discrepância são chamadas de quasi-random.

SOBOL_DATASET é um programa escrito na linguagem de programação C++ o qual cria um conjunto de dados quasi-random. Onde utiliza-se de entrada a dimensão espacial requerida, o número de pontos a ser gerado no espaço e um número de pontos gerados a se ignorar. Esse número de pontos são ignorados para que se faça o que foi chamado de aquecimento. Uma estratégia comumente utilizada e também utilizada nessa pesquisa é ignorar um número de pontos equivalente a maior potência de 2 que é igual ou maior que o número de pontos requeridos.

O programa disponibilizado na página da FLORIDA STATE UNIVERSITY (Universidade estadual da Florida) é distribuído sob a licença GNU LGPL e contém as rotinas descritas no apêndice A.

3.4 Normalização

O programa mencionado na seção anterior tem como saída pontos entre 0 e 1 no espaço, o que não é interessante para o nosso uso, portanto é necessário que se adote uma forma de obter desses pontos, valores que se adéquem aos requeridos pelo algoritmo ABC.

A forma adotada nesse trabalho é a normalização e a fórmula a seguir mostra a estratégia adotada.

$$ValN = \frac{f(x) - f_{min}}{f_{max} - f_{min}}$$

Onde:

$ValN$ corresponde ao valor gerado pelo SOBOL_DATASET

f_{max} valor máximo do limite desejado

f_{min} valor mínimo do limite desejado

$f(x)$ valor gerado pelo SOBOL_DATASET dentro do limite determinado por f_{min} e f_{max}

3.5 Colônia de Abelhas Artificial

Algumas técnicas de otimização se fundam no uso de candidatos a solução dos problemas, bem como métodos para atualização dessa população e isso as diferencia das técnicas habituais. Devemos ressaltar que nessa forma de abordagem, as interações coletivas entre os indivíduos dessa população geralmente nos leva a uma forma de comportamento ou inteligência coletiva. Isso é utilizado como base para a formulação dos algoritmos chamados bio-inspirados na busca da melhor solução possível. E entre as estratégias mais

recentes na literatura de algoritmos dessa categoria um que vem se destacando muito para a resolução de problemas de Otimização Combinatória é o Algoritmo de Colônia de Abelhas Artificial, ABC (Artificial Bee Colony), em inglês. Silva-Neto e Becceneri (2011), Pham et al. (2006)

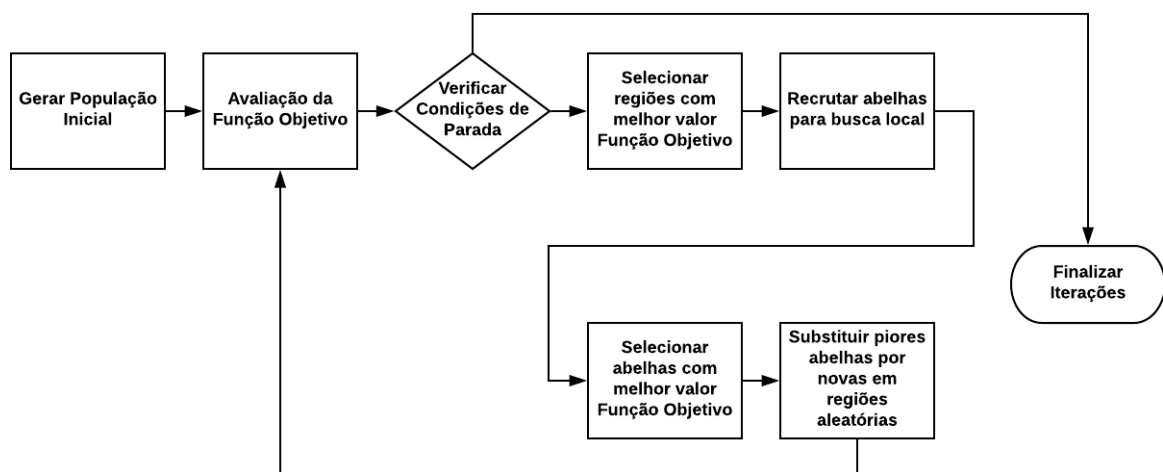
Como podemos observar na natureza, uma colônia de abelhas pode se estender por cerca de 10 km e em várias direções ao mesmo tempo em busca de exploração de várias fontes de alimento simultaneamente. Contextualizado nesse ambiente o ABC fundamenta-se nos comportamentos de indivíduos da colônia os quais trocam informações entre si sobre as possíveis fontes de alimento, como qual a direção, quantidade de alimento, qualidade, dentre outras informações. Com essas informações, é tomada uma decisão sobre quais as melhores regiões par se explorar, onde são descartadas as outras possibilidades. E, assim, se repete esse ciclo com outras regiões sendo visitadas pelos indivíduos a cada iteração. Silva-Neto e Becceneri (2011), Karaboga e Basturk (2007)

3.5.1 Descrição do Algoritmo

Assim como a maioria das heurísticas encontradas na literatura, o ABC é baseado em uma população de candidatos em potencial para a solução do problema de otimização. Dentre várias abordagens do ABC será descrita a implementação de Pham et al. (2006) que é inspirada no comportamento natural de busca por alimento.

A figura a seguir, Figura 4.1, apresenta em forma de fluxograma as etapas do algoritmo as quais serão detalhadas posteriormente.

Figura 3.1 – Fluxograma do algoritmo de Colônia de Abelhas Artificial



A seguir a descrição:

Passo 1 gerar uma população inicial, no caso de todos os testes realizados neste trabalho foram em um número de 50 abelhas, de forma quase randômica, onde n_{ae} são abelhas

escoteiras (exploradoras, batedoras) dentro dos m locais selecionados para a busca definidos dentro de um espaço, cada abelha é formada pelas coordenadas dos Pontos de Steiner;

Passo 2 fazer uma avaliação da Função Objetivo para todas as abelhas da população;

Passo 3 interromper a iteração caso todas as condições de parada tenham sido satisfeitas, as quais são um máximo de 8000 iterações ou 100 iterações sem alteração do custo da Árvore de Steiner considerada até então a melhor solução;

Passo 4 selecionar os m_l as melhores regiões com menor valor da Função Objetivo dentre os m locais visitados de forma que possam ser refinados;

Passo 5 recrutar n_{ep} abelhas para fazer uma busca local em cada uma das regiões m_l ;

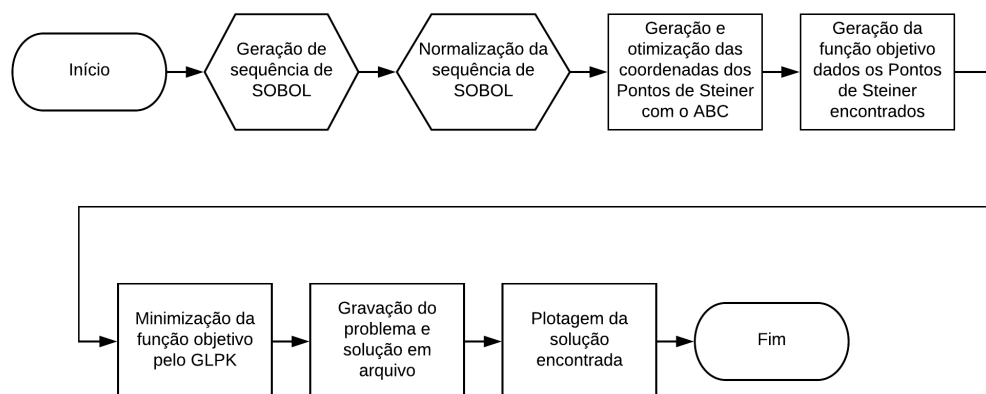
Passo 6 fazer a seleção da abelha com o menor valor da Função Objetivo de cada região;

Passo 7 substituir as abelhas com piores resultados por um número $n_{ae}-m_l$ para novas regiões aleatórias com raio de tamanho n_{gh} ;

Passo 8 voltar a execução do passo 2.

Esta abordagem possibilita encontrar as melhores coordenadas dos pontos de Steiner ao final da execução do algoritmo. Na **Figura 3.2** se encontra um fluxograma que ilustra todos os passos utilizados para se resolver PASE e a sua ordem de execução.

Figura 3.2 – Fluxograma dos estágios para a resolução do PASE utilizando a abordagem desse trabalho



4 RESULTADOS

A seguir dois exemplos de execução do algoritmo desenvolvido para resolução do PASE no R^n utilizando como entrada dados lidos de um arquivo de texto a respeito do número de Pontos Fixos, número de dimensões do espaço onde se encontra o problema descrito e as coordenadas dos respectivos Pontos Fixos. Os dados referentes às coordenadas dos Pontos de Steiner são gerados a partir do algoritmo ABC e melhorados a cada iteração.

As figuras **Figura 4.1** e **Figura 4.4** mostram o posicionamento dos pontos fornecidos como entrada antes da execução do algoritmo. Pontos Fixos em preto e Pontos de Steiner em vermelho. Em sequência as figuras **Figura 4.2** e **Figura 4.5** ilustram o resultado da execução com os Pontos de Steiner, em seguida as figuras **Figura 4.3** e **Figura 4.6** ilustram a saída do algoritmo ABC com seus Pontos de Steiner e as arestas geradas com o menor custo devido ao posicionamento dos pontos tornando-se satisfatória a saída.

Exemplo 1 referente a Árvore de Steiner pertencente ao espaço R^2 para os conjuntos $P=5$ de Pontos Fixos fornecidos através de um arquivo e $S=3$ de Pontos de Steiner fornecidos pelo algoritmo ABC, indicando os pontos e suas coordenadas no R^2 :

- $P=(1\{0,0\}, 2\{0,10\}, 3\{10,0\}, 4\{10,10\}, 5\{5,15\})$
- $S=(6\{3.258911, 2.616963\}, 7\{2.251395, 9.126785\}, 8\{5.504521, 11.743847\})$

Figura 4.1 – Exemplo 1 de posicionamento de Pontos Fixos

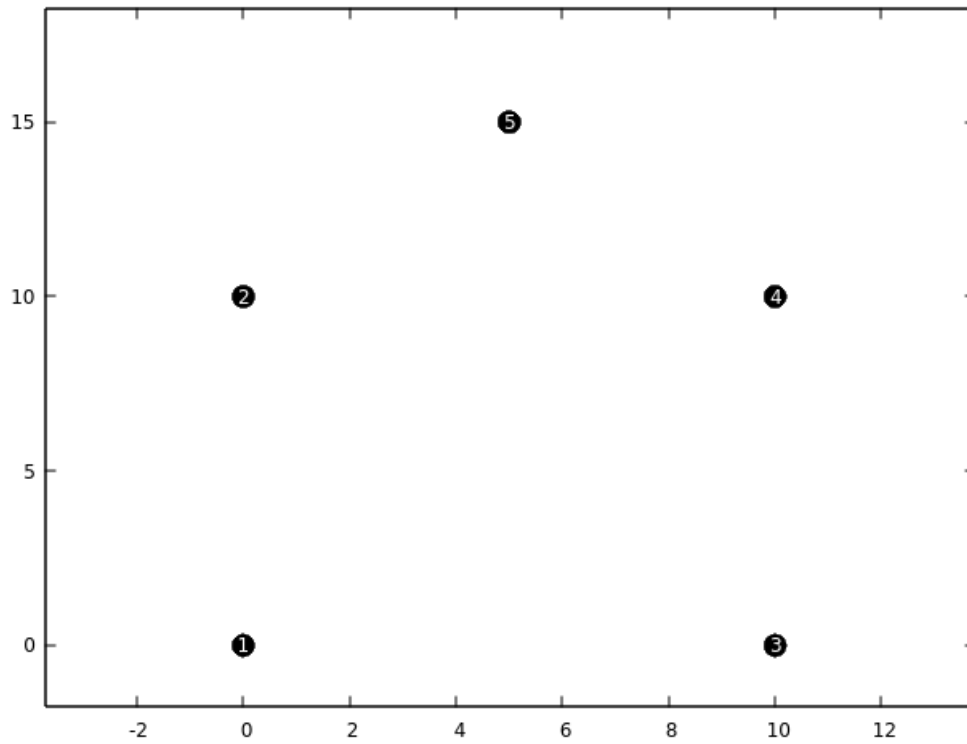
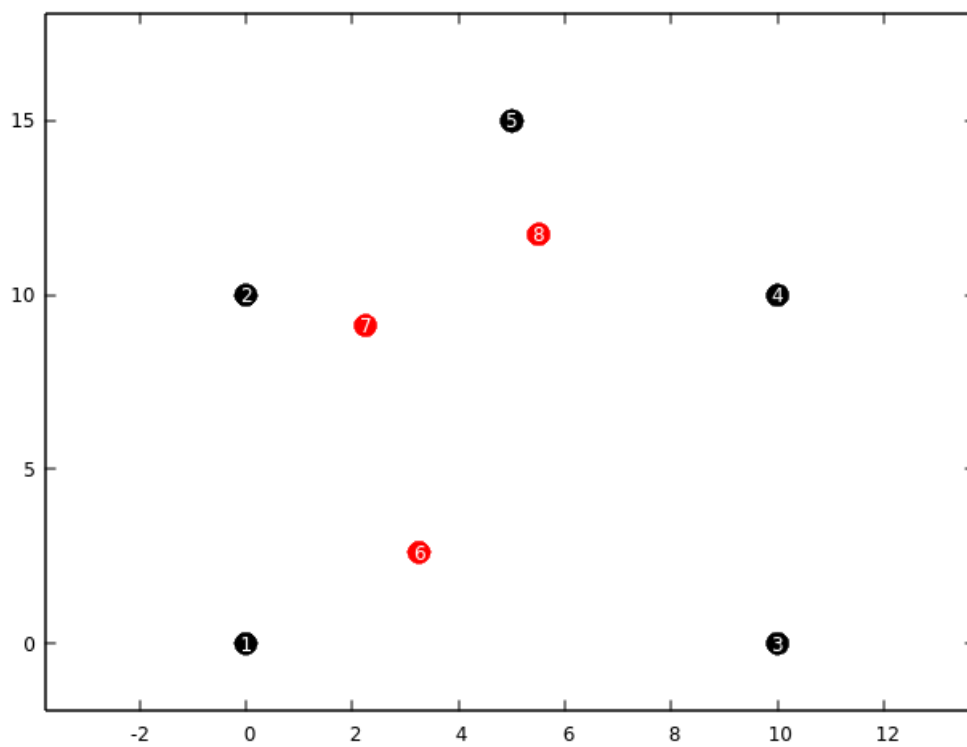
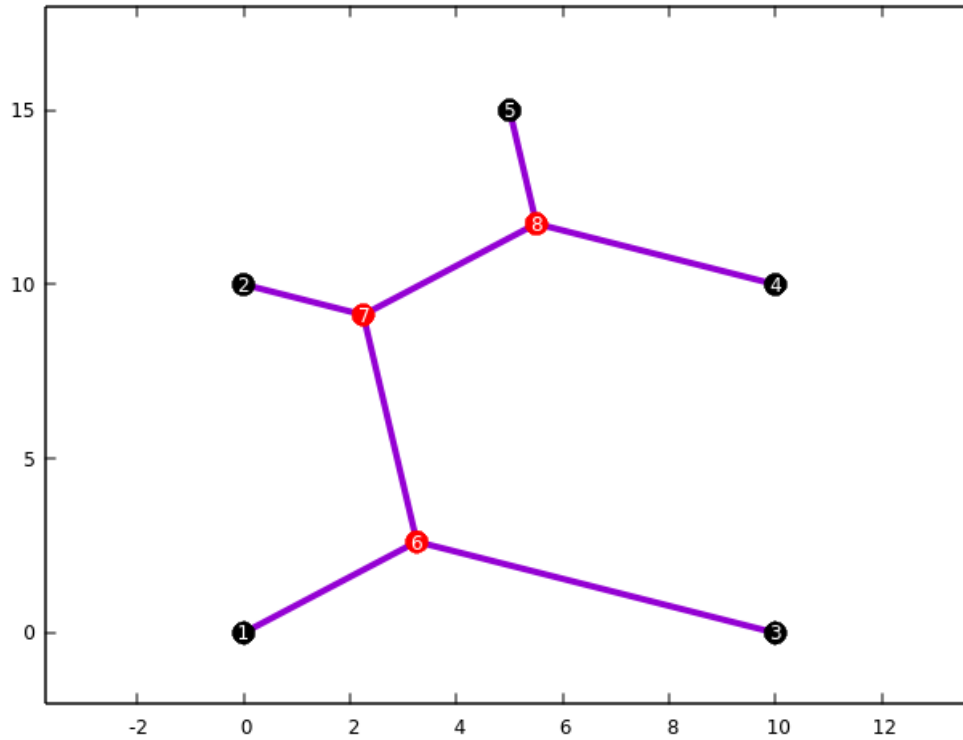


Figura 4.2 – Exemplo 1 de posicionamento de Pontos Fixos e Pontos de Steiner



Resultando em custo $Z = 32.704$, referente a soma do custo das arestas utilizadas na melhor solução possível para este caso, as quais foram: x_{1i6} , x_{2i7} , x_{3i6} , x_{4i8} , x_{5i8} , x_{6i7} , x_{7i8} .

Figura 4.3 – Exemplo 1 de execução do algoritmo ABC



Exemplo 2 referente a Árvore de Steiner pertencente ao espaço R^2 para os conjuntos $P=7$ de Pontos Fixos fornecidos através de um arquivo e $S=4$ de Pontos de Steiner fornecidos pelo algoritmo ABC, indicando os pontos e suas coordenadas no R^2 :

- $P=(1\{5,7\}, 2\{2,7\}, 3\{9,3\}, 4\{0,4\}, 5\{7,9\}, 6\{6,0\}, 7\{3,9\})$
- $S=(8\{3.128074,7.825262\}, 9\{5.000000,7.000000\}, 10\{2.000000,7.000000\}, 11\{5.758924,6.978221\}, 12\{7.876955,3.034298\})$

Figura 4.4 – Exemplo 2 de posicionamento de Pontos Fixos

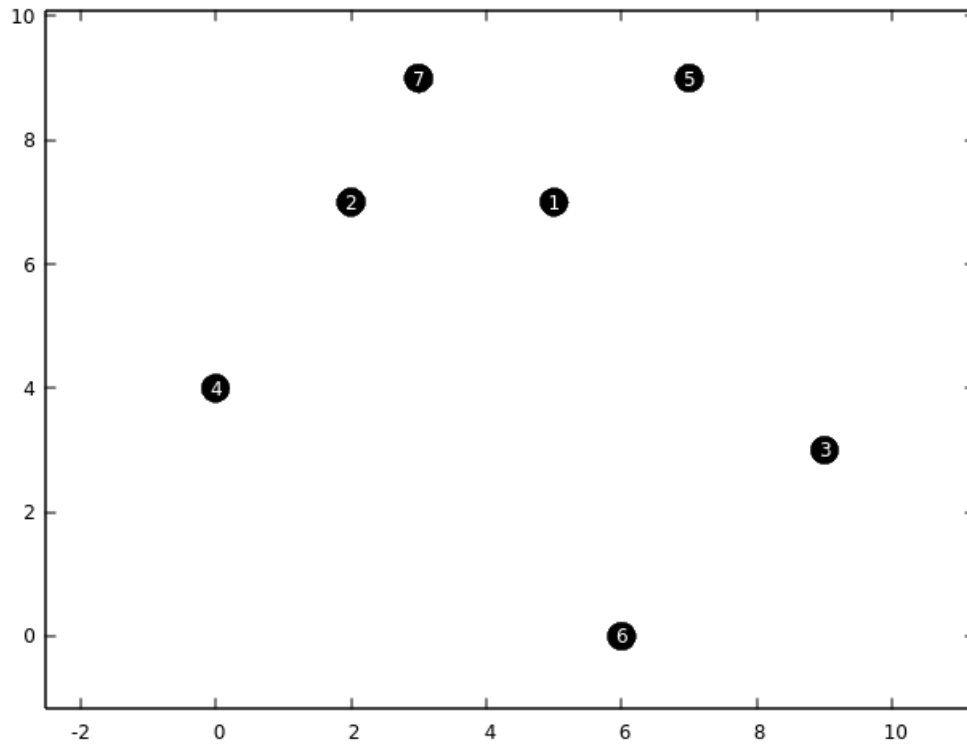
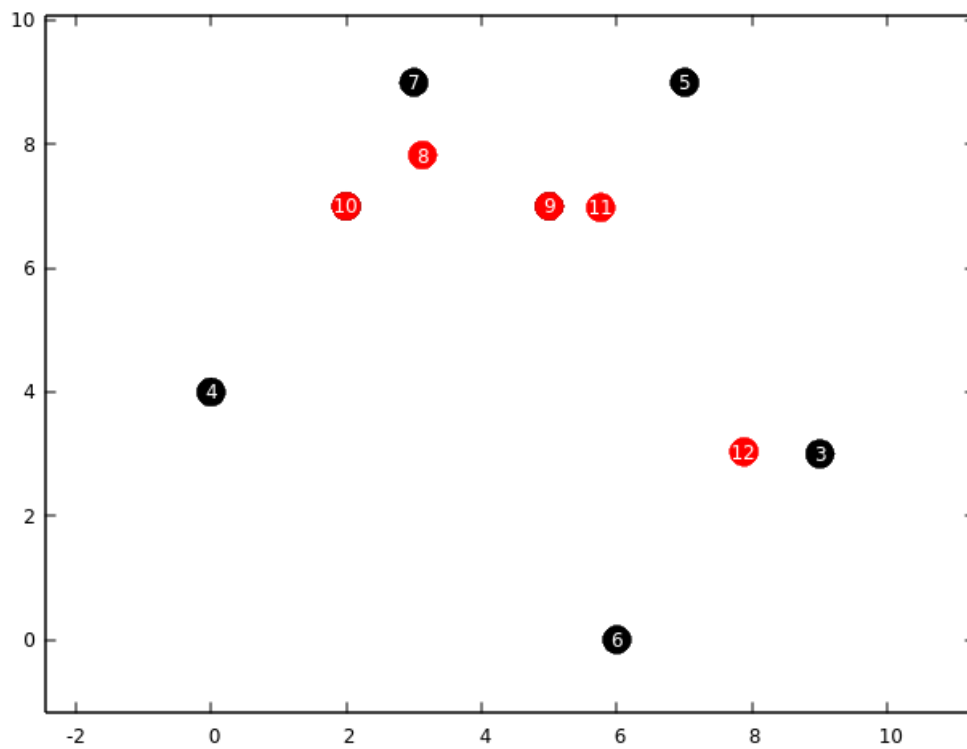
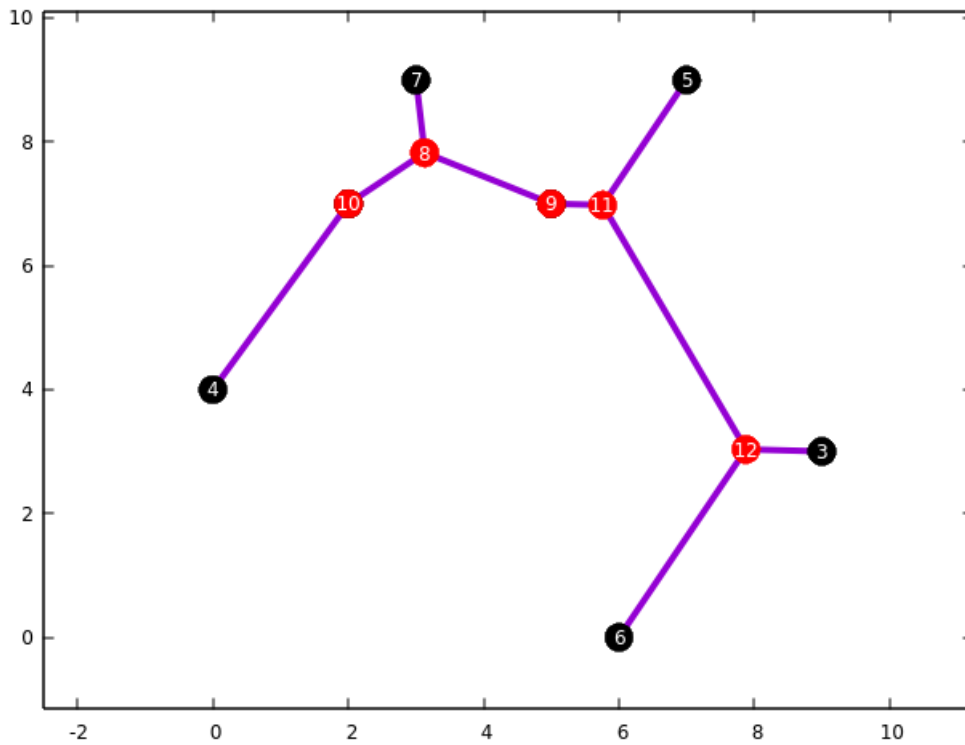


Figura 4.5 – Exemplo 2 de posicionamento de Pontos Fixos e Pontos de Steiner



Resultando em custo $Z = 20.530$, referente a soma do custo das arestas utilizadas na melhor solução possível para este caso, as quais foram: x_{1i9} , x_{2i10} , x_{3i12} , x_{4i10} , x_{5i11} , x_{6i12} , x_{7i8} , x_{8i9} , x_{8i10} , x_{9i11} , x_{11i12} .

Figura 4.6 – Exemplo 2 de execução do algoritmo ABC



4.1 Comparativos

Nesta seção é feita uma breve descrição de duas abordagens diferentes da desenvolvida nesse trabalho e feito um comparativo de resultados com o objetivo de se avaliar o desempenho da técnica utilizada e se obter uma visão de futuros trabalhos e ajustes a serem feitos de forma que se possa aperfeiçoar o algoritmo de Colônia de Abelhas Artificial e também cogitar a inclusão ou exclusão de técnicas e ferramentas que propiciem um avanço nesta pesquisa.

Os testes foram executados numa máquina com a seguinte configuração: Intel® Xeon(R) CPU E3-1240 V2 @ 3.40GHz 8 núcleos, 12 GiB de RAM e sistema operacional Linux Ubuntu 18.04 Bionic Beaver.

Os resultados mostrados para cada número de Pontos Fixos foram obtidos através de uma média simples de 10 resultados executados em cada algoritmo para a mesma quantidade de pontos e com as mesmas coordenadas em cada situação. O quesito tempo foi medido utilizando a biblioteca *time.h* da linguagem de programação C. Esta biblioteca contém as definições de funções para obter e manipular informações referentes a data e tempo.

4.1.1 Wsmith

Em Smith (1992) foi proposto um método exaustivo para encontrar a solução exata para o PASE em dimensão $n \geq 3$, este método consistia basicamente em enumerar todas as possíveis soluções para o problema e minimizá-las, selecionando dentre estas a melhor solução. A enumeração das soluções é realizada através dos vetores topologia, cada um representando uma possível topologia cheia, que são aplicados, em seguida, no contexto de um algoritmo branch-and-bound, o que evita a minimização de algumas soluções ruins. Sendo P o número de pontos dados, a estratégia geral deste algoritmo é:

1. Encontrar todas as FST de tamanho P .
2. Otimizar as coordenadas dos pontos de Steiner de todas as FST geradas no passo 1, de forma a encontrar a árvore de menor custo.

A primeira consideração a respeito do algoritmo de Smith é que ele só procura por topologias completas (FST). A razão para esta simplificação é que o método de otimização utilizado por Smith permite considerar as outras topologias como degenerações da topologia completa, onde alguns pontos de Steiner se sobrepõem aos pontos normais. Apesar disto, o número de FST ainda é exponencial em n . No segundo passo mencionado acima, o de otimizar as coordenadas dos pontos de Steiner, o algoritmo de Smith procede de maneira iterativa, sendo que a regra de parada utiliza a propriedade de não ter nenhum ângulo entre arestas da árvore alvo (de Steiner) maior que 120 graus. Smith afirma em seu trabalho que seu algoritmo só é capaz de resolver problemas com doze pontos, aproximadamente, mas em qualquer dimensão. Este fato inviabiliza a utilidade de seu algoritmo em casos práticos (instâncias com número grande de pontos). Apesar disto, o trabalho de Smith (1992) é importante pela possibilidade de utilização de forma eficiente do passo de minimização do método de Smith para encontrar as posições ótimas dos pontos de Steiner para um vetor topologia dado. (ROCHA, 2008)

Na **Tabela 4.1** vemos os resultados da abordagem ABC comparados com o algoritmo Wsmith para o PASE com 9, 18 e 20 pontos respectivamente, onde nota-se uma diferença considerável em tempo computacional na medida em que se acrescenta pontos de Steiner ao problema, juntamente com um menor custo da solução proposta pelo ABC.

4.1.2 GRASP

GRASP (Greedy Randomized Adaptive Search Procedure – Procedimento de Busca Gulosa Adaptativa Aleatorizada) é uma metaheurística de multipartida, proposta em um trabalho desenvolvido por Raidl (2006), em que cada iteração é composta de uma fase de construção e de uma fase de aprimoramento. Durante o processo, a melhor solução até o momento é armazenada e atualizada sempre que a fase de aprimoramento resultar em uma

Tabela 4.1 – Tabela de comparativo entre as abordagens ABC e Wsmith para o PASE

	Pontos	Tempo(s)	Tamanho
ABC	9	9,428	38,255
Wsmith	9	0,432	32,677
ABC	18	306,221	59,588
Wsmith	18	655,346	59,933
ABC	20	485,179	66,715
Wsmith	20	33877,502	68,429

solução de melhor aproveitamento. Ao final de um número pré determinado de iterações, o algoritmo retorna a melhor solução encontrada. A fase de construção gera uma solução viável para o problema através de um procedimento parcialmente guloso e parcialmente aleatório. A cada etapa da construção a seleção dos elementos que vão compor a solução é feita aleatoriamente em um subconjunto dos melhores elementos candidatos, denominada Lista Restrita de Candidatos (LRC). A fase de aprimoramento é uma busca local. Feo e Resende (1995) (ROCHA, 2008)

Na **Tabela 4.2** vemos os resultados da abordagem ABC comparados com o algoritmo GRASP para o PASE com 9, 18 e 20 pontos respectivamente, onde nota-se uma diferença considerável em tempo computacional na medida em que se acrescenta pontos de Steiner ao problema, dessa vez com a técnica concorrente tendo melhores resultados que o ABC se tratando de tempo, juntamente com um menor custo da solução proposta pelo ABC.

Tabela 4.2 – Tabela de comparativo entre as abordagens ABC e GRASP para o PASE

	Pontos	Tempo(s)	Tamanho
ABC	9	9,428	38,255
GRASP	9	1,3	32,677
ABC	18	306,221	59,588
GRASP	18	2,8	90,435
ABC	20	485,179	66,715
GRASP	20	25,95	92,307

5 CONSIDERAÇÕES FINAIS

Foi apresentado nesse trabalho um método de resolução para o PASE no R^n o qual gerou na maioria dos testes executados a solução ótima de disposição de Pontos de Steiner e conexão entre os pontos, obedecendo a formulação matemática do problema abordado. Assim tendo sido testado, exemplificado e comparado com outras soluções no decorrer do mesmo, fazendo com que os resultados tenham sido bastante satisfatórios em relação a expectativa ao início do trabalho e ao que foi proposto a ser feito durante o seu desenvolvimento.

O algoritmo implementado lendo esses dados de arquivos de texto, contendo coordenadas dos Pontos Fixos em grande parte das vezes geradas de forma aleatória para testes, em seguida aplicando o algoritmo ACB para inclusão de pontos de Steiner em uma localização no espaço de forma a minimizar o custo da ASM, calculando os pesos de todas as possíveis arestas e retornando a melhor combinação possível com o menor custo. O método proposto utiliza de Programação Linear Inteira para a formulação matemática do Problema da Árvore de Steiner Euclidiano no R^n e ferramenta de resoluções de problema com esse aspecto GLPK e do algoritmo de Colônia de Abelhas Artificial somado com algoritmo de geração de coordenadas quasi-random de baixa discrepância.

6 TRABALHOS FUTUROS

Os resultados obtidos pela abordagem utilizada para o PASE são muito promissores. Ainda assim, estudos maiores e novos desenvolvimentos podem ser feitos com o objetivo de buscar uma melhor eficiência em casos com menos pontos e com dimensões maiores que 3. Como principais sugestões, têm-se:

- Realizar testes com instâncias em dimensões maiores ($n > 3$) que as utilizadas neste trabalho e analisar o comportamento das técnicas propostas para as mesmas.
- Incorporação do algoritmo formulado por Smith (1992), de onde pretende-se incluir a sua primeira saída, a primeira disposição de Pontos de Steiner, na primeira entrada do ABC e realização de testes para comparar com a solução atual com o intuito de ter ganhos em tempo computacional e tempo real.
- Implementação de um limite no espaço onde os Pontos de Steiner podem ser gerados afim de melhorar os resultados atuais com a diminuição da distância de eventuais pontos gerados em locais inapropriados.

REFERÊNCIAS

- AMARIR, A.; ZEGRARI, F.; IDRISSE, A. Degree constrained spanning tree problem solving using hybrid modified genetic algorithm. In: **ACM. Proceedings of the International Conference on Big Data and Advanced Wireless Technologies**. [S.l.], 2016. p. 34.
- BONDY, J. A.; MURTY, U. S. R. et al. **Graph theory with applications**. [S.l.]: Citeseer, 1976. v. 290.
- BOSE, P. et al. Routing with guaranteed delivery in ad hoc wireless networks. **Wireless networks**, Springer-Verlag New York, Inc., v. 7, n. 6, p. 609–616, 2001.
- BRATLEY, P.; FOX, B. L. Algorithm 659: Implementing sobol’s quasirandom sequence generator. **ACM Trans. Math. Softw.**, ACM, New York, NY, USA, v. 14, n. 1, p. 88–100, mar. 1988. ISSN 0098-3500. Disponível em: <<http://doi.acm.org/10.1145/42288.214372>>.
- BUI, T. N.; ZRNCIC, C. M. An ant-based algorithm for finding degree-constrained minimum spanning tree. In: **ACM. Proceedings of the 8th annual conference on Genetic and evolutionary computation**. [S.l.], 2006. p. 11–18.
- DREYER, D. R.; OVERTON, M. L. Two heuristics for the euclidean steiner tree problem. 2002.
- FEO, T. A.; RESENDE, M. G. Greedy randomized adaptive search procedures. **Journal of global optimization**, Springer, v. 6, n. 2, p. 109–133, 1995.
- GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear: modelos e algoritmos**. [S.l.]: Elsevier, 2005.
- KARABOGA, D.; BASTURK, B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. **Journal of global optimization**, Springer, v. 39, n. 3, p. 459–471, 2007.
- KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. **science**, American Association for the Advancement of Science, v. 220, n. 4598, p. 671–680, 1983.
- MACULAN, N.; MICHELON, P.; XAVIER, A. E. The euclidean steiner tree problem in r n: A mathematical programming formulation. **Annals of Operations Research**, Springer, v. 96, n. 1-4, p. 209–220, 2000.
- MAKHORIN, A. Glpk linear programming kit: Implementation of the revised simplex method. **Glpk documentation**, Moscow Aviation Institute, Moscow, Russia, 2010.
- MARINS, F. A. S. Introdução a pesquisa operacional. **São Paulo: Cultura Acadêmica: Universidade Estadual Paulista**, 2011.

- MONTENEGRO, F.; TORREÃO, J. R.; MACULAN, N. Microcanonical optimization algorithm for the euclidean steiner problem in \mathbb{R}^n with application to phylogenetic inference. **Physical Review E**, APS, v. 68, n. 5, p. 056702, 2003.
- NARULA, S. C.; HO, C. A. Degree-constrained minimum spanning tree. **Computers & Operations Research**, Elsevier, v. 7, n. 4, p. 239–249, 1980.
- OLIVEIRA, N. V. de. O problema de steiner e a estrutura das biomacromoléculas. 2005.
- PHAM, D. T. et al. -the bees algorithm—a novel tool for complex optimisation problems. In: **Intelligent Production Machines and Systems**. [S.l.]: Elsevier, 2006. p. 454–459.
- RAIDL, G. R. A unified view on hybrid metaheuristics. In: SPRINGER. **International Workshop on Hybrid Metaheuristics**. [S.l.], 2006. p. 1–12.
- ROCHA, M. L. **APLICAÇÕES DE ALGORITMOS PARALELOS E HÍBRIDOS PARA O PROBLEMA DE ÁRVORE DE STEINER EUCLIDIANA NO \mathbb{R}^n** . Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2008.
- SANTUARI, A. **Steiner tree NP-completeness proof**. [S.l.], 2003.
- SILVA-NETO, A.; BECCENERI, J. **Inteligência Computacional Aplicada a Problemas Inversos em Transferência Radiativa**. [S.l.]: SBMAC, 2011.
- SMITH, J. M. Steiner minimal trees in \mathbb{E}^3 : Theory, algorithms, and applications. In: **Handbook of Combinatorial Optimization**. [S.l.]: Springer, 1998. p. 1143–1216.
- SMITH, W. D. How to find steiner minimal trees in euclidean-space. **Algorithmica**, Springer, v. 7, n. 1-6, p. 137–177, 1992.
- WEST, D. B. et al. **Introduction to graph theory**. [S.l.]: Prentice hall Upper Saddle River, 2001. v. 2.
- WU, B. Y.; CHAO, K.-M. Steiner minimal trees. URL: <http://www.csie.ntu.edu.tw/~kmchao/tree07spr/Steiner.pdf>. (Last accessed: 13.03. 2017), 2004.

A ROTINAS DO PROGRAMA SOBOL_DATASET

- MAIN é a função principal do SOBOL_DATASET.
- I8_BIT_HI1 retorna a posição do bit maior que 1 na base 2 em um inteiro.
- I8_BIT_LO0 retorna a posição do bit menor que 0 na base 2 em um inteiro.
- I8_SOBOL gera um novo vetor quasi-random de sobol a cada chamada.
- I8_SOBOL_GENERATE gera um conjunto de dados de sobol.
- R8MAT_WRITE escreve um arquivo R8MAT.
- TIMESTAMP imprime o conjunto de dados atual.