



PROBLEMA EUCLIDIANO DE STEINER EM ESPAÇOS DE DIMENSÃO
MAIOR OU IGUAL A TRÊS : MODELOS EXATOS E HEURÍSTICAS

Brígida Alexandre Sartini

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientador: Marcia Helena Costa Fampa

Rio de Janeiro
Outubro de 2013

PROBLEMA EUCLIDIANO DE STEINER EM ESPAÇOS DE DIMENSÃO
MAIOR OU IGUAL A TRÊS : MODELOS EXATOS E HEURÍSTICAS

Brígida Alexandre Sartini

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ
COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE)
DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR
EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Marcia Helena Costa Fampa, D.Sc.

Prof. Fernanda Maria Pereira Raupp, D.Sc.

Prof. Luiz Satoru Ochi, D.Sc.

Prof. Felipe Maia Galvão França, D.Sc.

Prof. Nelson Maculan Filho, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

OUTUBRO DE 2013

Sartini, Brígida Alexandre

Problema Euclidiano de Steiner em Espaços de Dimensão Maior ou Igual a Três : Modelos Exatos e Heurísticas/Brígida Alexandre Sartini. – Rio de Janeiro: UFRJ/COPPE, 2013.

XI, 68 p.: il.; 29, 7cm.

Orientador: Marcia Helena Costa Fampa

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2013.

Referências Bibliográficas: p. 66 – 68.

1. Problema Euclideano de Steiner. 2. Programação Inteira Não Linear Mista. 3. Heurística. I. Fampa, Marcia Helena Costa. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Ao meu pai Ítalo(in memoriam)
e à minha mãe Rita.

Agradecimentos

Gostaria de agradecer a minha orientadora Prof^a. Marcia Helena Costa Fampa por seus ensinamentos, por me incentivar em todo esse tempo e pela paciência infinita.

Agradeço imensamente ao Prof. Nelson Maculan Filho a ajuda nos momentos mais difíceis e por compartilhar sua sabedoria.

A todos os meus professores e colegas do PESC/COPPE, agradeço especialmente a Wendel Xavier de Melo, Alan de Freitas, Flávio Montenegro, Jesus Ossian Cunha, Renan Vicente Pinto e Virgínia Silva da Costa pelo convívio e apoio.

Aos membros da banca pelas sugestões e correções.

Aos funcionários do PESC/COPPE.

Aos amigos André Luiz dos Santos, Aquiles Braga de Queiroz, Carla Gomes, Claudia Mazza, Ronaldo Gregório.

E às pessoas que me ajudaram de diversas formas.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

PROBLEMA EUCLIDIANO DE STEINER EM ESPAÇOS DE DIMENSÃO
MAIOR OU IGUAL A TRÊS : MODELOS EXATOS E HEURÍSTICAS

Brígida Alexandre Sartini

Outubro/2013

Orientador: Marcia Helena Costa Fampa

Programa: Engenharia de Sistemas e Computação

Nesta monografia apresentamos o Problema de Steiner Euclidiano e os dois modelos matemáticos existentes para este problema. Apresentamos os algoritmos de programação não linear inteira mista e os resultados numéricos obtidos com o uso de um pacote recentemente desenvolvido para esta classe de problemas, aplicado aos modelos do Problema de Steiner Euclidiano, em instâncias do \mathbb{R}^3 . Propomos a aplicação de um framework chamado POPMUSIC para o Problema de Steiner.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

EUCLIDEAN STEINER PROBLEM IN SPACES OF DIMENSION GREATER
THAN OR EQUAL THREE: EXACT MODELS AND HEURISTICS

Brígida Alexandre Sartini

October/2013

Advisor: Marcia Helena Costa Fampa

Department: Systems Engineering and Computer Science

This work presents the Euclidean Steiner Problem and two mathematical models for this problem. We present algorithms for mixed integer nonlinear programming and numerical results obtained using a newly developed package for this class of problems, applied to the Euclidean Steiner Problem, in \mathbb{R}^3 . We propose to apply a framework for Steiner's Problem called POPMUSIC.

Sumário

Lista de Figuras	x
Lista de Tabelas	xi
1 Introdução	1
2 O Problema de Steiner Euclidiano	3
2.1 Histórico	3
2.2 Definições e Propriedades Básicas	6
2.2.1 Soluções do Problema de Steiner	10
2.2.2 Razão de Steiner	13
3 Modelos	15
3.1 Modelos matemáticos do Problema de Steiner Euclidiano	15
3.1.1 Modelo 1	15
3.1.2 Modelo 2	18
3.1.3 Estudo Teórico sobre Redução do Big-M	20
4 Algoritmos para Problemas de Programação Não Linear Inteiros	
Mistos	22
4.1 Forma Geral do MINLP	23
4.2 Método Branch and Bound	24
4.3 Decomposição Generalizada de Benders	24
4.3.1 Algoritmo da Decomposição Generalizada de Benders para (MINLP ₁)	27

4.4	Método da Aproximação Externa	28
4.4.1	Algoritmo de Aproximação Exterior para (MINLP ₁)	31
4.5	Algoritmo Híbrido	32
4.5.1	Algoritmo Branch and Cut baseado em Aproximação Exterior	34
4.5.2	Híbrido	36
5	Resultados Numéricos	38
5.1	Tabelas	39
6	Heurística para o PSE: Algoritmo Microcanônico	44
6.1	Algoritmo de Otimização Microcanônica	44
6.1.1	Inicialização	45
6.1.2	Sampling	45
7	Framework POPMUSIC	47
7.1	Algoritmo POPMUSIC	48
8	Problema de Steiner em Poliedros Regulares	53
8.1	Tetraedro: $p = 4$	54
8.2	Octaedro: $p = 6$	54
8.3	Cubo: $p = 8$	58
8.4	Icosaedro: $p = 12$	59
8.5	Dodecaedro: $p = 20$	61
8.6	Análise de Redução do Big-M do Modelo 2	62
8.7	Análise da Mudança de Escala das instâncias	63
9	Conclusões	64
	Referências Bibliográficas	66

Lista de Figuras

2.1	Método de Torricelli	4
2.2	Linhas de Simpson	4
2.3	Vértice correspondente a ângulo maior do que 120 graus	5
2.4	Árvore mais curta com a inserção de um ponto de Steiner	7
2.5	Árvores com a mesma topologia: 1 – 5, 2 – 5, 3 – 6, 4 – 6, 5 – 6	8
2.6	Representação de uma topologia cheia com 4 nós dados e 2 nós de Steiner	9
3.1	Grafo da instância com 6 pontos fixos e 4 de Steiner	16
8.1	Árvore Mínima de Steiner Aproximada - Tetraedro	54
8.2	Soluções Não isomorfas	56
8.3	Solução	56
8.4	Árvore Mínima de Steiner Aproximada - Octaedro	57
8.5	Árvore Mínima de Steiner Aproximada - Cubo	59
8.6	Árvore Mínima de Steiner Aproximada - Icosaedro	60
8.7	Árvore Mínima de Steiner Aproximada - Dodecaedro	62

Lista de Tabelas

5.1	Aspectos gerais das instâncias de OR-Library estudadas [1]	41
5.2	Resultados obtidos com algoritmo B-Hyb do BONMIN	42
5.3	Resultados obtidos com algoritmo B-BB do BONMIN	42
5.4	Resultados obtidos com algoritmo B-OA do BONMIN	43
5.5	Melhores Resultados Obtidos	43
7.1	Tabela Comparativa	51

Capítulo 1

Introdução

Esta monografia trata do estudo do Problema de Steiner Euclidiano em n dimensões (PSE) e de algoritmos que obtêm uma solução para este problema. Apresenta-se pela primeira vez, de nosso conhecimento, resultados numéricos obtidos com a aplicação de um pacote computacional recentemente desenvolvido para problemas não lineares inteiros mistos (MINLP) aos modelos matemáticos encontrados na literatura para o PSE. Realiza-se o estudo de uma das heurísticas existentes para este problema com o intuito de desenvolver métodos híbridos, que utilizem tanto os modelos matemáticos como procedimentos heurísticos para obter boas soluções numéricas aproximadas para o problema. Propõe-se a aplicação de um framework chamado POPMUSIC ao Problema de Steiner.

O PSE consiste em, dados p pontos no espaço euclidiano de dimensão n , encontrar a árvore de menor comprimento que conecta estes pontos, podendo usar ou não pontos extras cuja inserção reduza o comprimento total da árvore. Estes pontos extras são chamados de pontos de Steiner. Na solução do PSE deve-se determinar a localização dos nós de Steiner e os arcos que os conectam aos pontos fixos. O PSE é considerado um problema NP-difícil [2] e na prática, constitui um problema de alta complexidade computacional, o único algoritmo exato existente para resolvê-lo em \mathbb{R}^n demanda um tempo computacional elevado para um número p de pontos dados ainda relativamente pequeno, de modo que se torna conveniente a utilização de métodos heurísticos nesse problema.

Na literatura encontramos artigos que apresentam um modelo matemático para o PSE, como um problema de programação não linear inteiro misto (MINLP), para o qual existem algoritmos, como aproximação exterior, decomposição generalizada de Benders, algoritmo de planos de corte estendidos e algoritmos híbridos desenvolvidos pelo grupo COIN-OR [3] que permitem encontrar soluções exatas para problemas MINLP convexos e que são heurísticas para os não convexos. Desenvolvemos um estudo destes algoritmos.

Esta monografia possui a seguinte estrutura: No capítulo 2, apresentaremos um histórico do PSE, as definições e propriedades básicas e os algoritmos existentes na literatura que resolvem o PSE, no plano e em dimensão maior do que 2. No capítulo 3, descrevemos os novos modelos matemáticos para o PSE. No capítulo 4 estudamos os algoritmos para problemas não lineares inteiros mistos. A seguir, no capítulo 5, os resultados computacionais com o pacote BONMIN são apresentados. Será descrita no capítulo 6 a heurística microcanônica e no capítulo 7 o framework POPMUSIC aplicado ao Problema de Steiner usado em conjunto com a heurística microcanônica. No capítulo 8 analisamos o desempenho do algoritmo híbrido em instâncias do \mathbb{R}^3 , as quais a solução exata é conhecida. No nono capítulo são apresentadas as conclusões.

Capítulo 2

O Problema de Steiner Euclidiano

2.1 Histórico

O Problema de Steiner Euclidiano (PSE) tem origem no século XVII, quando Fermat propôs o seguinte problema: encontrar um ponto no plano cuja soma das distâncias deste ponto a outros 3 pontos dados seja mínima. Este problema proposto no artigo "*Treatise on minima and maxima*", ficou conhecido como Problema de Fermat. Torricelli propôs uma solução geométrica para este problema em 1640. Ele afirmou que o ponto procurado era a interseção de três círculos circunscritos a três triângulos equiláteros construídos externamente ao triângulo formado pelos três pontos dados, com a medida do lado de cada triângulo da construção sendo a distância entre os pontos. Este ponto ficou conhecido como ponto de Torricelli. A Figura (2.1) mostra a construção do ponto de Torricelli, sendo os pontos A, B e C os pontos dados e o ponto T o de Torricelli associado a eles.

Em 1647, Cavalieri, em seu livro "*Excercitationes Geometricae*", mostrou que os segmentos de reta que unem o ponto de Torricelli aos três pontos dados formam ângulos de 120° . Em 1750, Simpson provou que os 3 segmentos de retas que unem os vértices externos de cada triângulo equilátero aos respectivos vértices opostos do triângulo formado pelos pontos dados se interceptam no ponto de Torricelli. Estes segmentos são conhecidos por linhas de Simpson. A Figura (2.2) mostra as linhas de Simpson se interceptando no ponto de Torricelli.

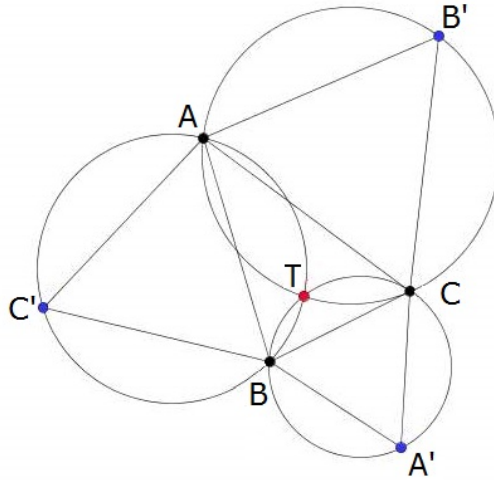


Figura 2.1: Método de Torricelli

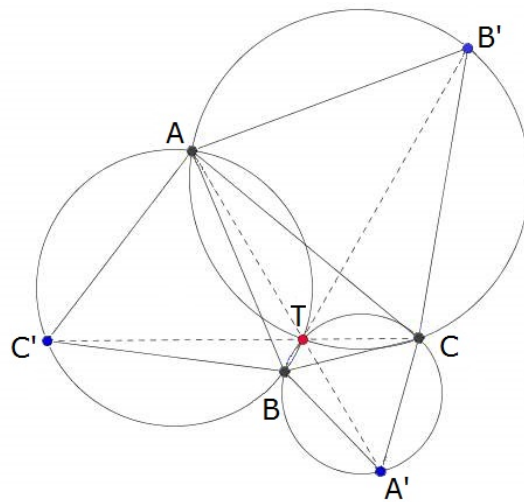


Figura 2.2: Linhas de Simpson

Em 1834, Heinen provou que se o triângulo formado pelos pontos dados possui um ângulo maior ou igual a 120° , o ponto de Torricelli se localiza fora da região interior do triângulo, como pode ser observado na Figura (2.3) e que o vértice associado a este ângulo é o ponto que minimiza a soma das distâncias, na Figura 2.3, o vértice A é o ponto minimizador. Este fato foi demonstrado por Bertrand em 1853. Heinen também mostrou que as 3 linhas de Simpson possuem o mesmo comprimento, igual à soma das distâncias do ponto de Torricelli aos 3 pontos dados, isto é, seguindo a Figura 2.2, temos $\overline{AA'} + \overline{BB'} + \overline{CC'} = \overline{AT} + \overline{BT} + \overline{CT}$.

No livro "Fluxions" de Simpson, ele incluiu um exercício, conhecido como Pro-

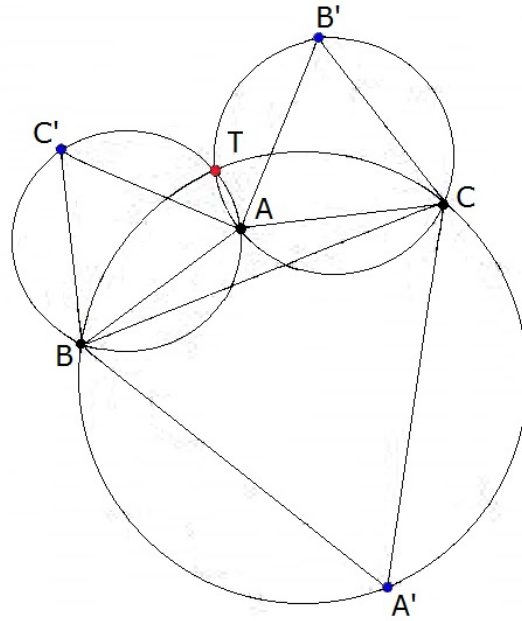


Figura 2.3: Vértice correspondente a ângulo maior do que 120 graus

blema Geral de Fermat, que procura um ponto no plano ou em espaço de dimensão maior, cuja soma das distâncias de n pontos dados a este ponto seja mínimo. Vários matemáticos procuraram resolver o problema, entre eles o suíço Jacob Steiner, que o generalizou permitindo a inclusão de mais pontos. Jarník e Kössler em 1934 propuseram o problema de encontrar a menor rede que conecta n pontos no plano. Eles estudaram o caso particular em que os pontos dados são vértices de polígonos regulares e encontraram soluções para $n = 3, 4, 5$. Courant e Robbins fizeram referências a este problema em 1941 no livro "What Is Mathematics?", fazendo uma conexão entre o problema de Fermat e encontrar a menor rede que liga 3 pontos. Eles nomearam a questão de encontrar o menor caminho possível que liga n pontos dados num plano por segmentos de retas, podendo-se acrescentar k pontos, com $k \leq n - 2$ como Problema da Árvore Mínima de Steiner e pela fama do livro, o problema passou a ser conhecido desta forma. Em 1961, Melzak foi quem primeiro estabeleceu propriedades básicas das árvores e propôs um algoritmo para solução do problema, usando construções geométricas. Mas, este algoritmo encontra solução do problema no plano, dado que no plano só há 2 possibilidades de escolha de

localização de um ponto, obtidos através de construções com régua e compasso. Em dimensões maiores a escolha se torna infinita. Além disso, o algoritmo resolve o problema para uma quantidade de pontos pequena, a complexidade do problema cresce no mínimo exponencialmente à medida que aumenta o número de pontos dados. Gilbert e Pollak, em 1968, deram um tratamento minucioso ao problema de Steiner, nomearam a solução do problema de **Árvore Mínima de Steiner** e os vértices da árvore mínima não pertencentes ao conjunto de pontos dados de pontos de Steiner. Em 1992, Smith apresentou o primeiro algoritmo exato para resolver o problema de Steiner euclidiano tridimensional, para uma quantidade pequena de pontos e que serviu de base para desenvolver heurísticas nessa dimensão.

2.2 Definições e Propriedades Básicas

O PSE em \mathbb{R}^n é definido da seguinte forma: Dado um conjunto de pontos em \mathbb{R}^n , chamados de pontos (ou nós) terminais, encontrar uma rede de comprimento mínimo que conecte estes pontos, sendo possível utilizar pontos adicionais na construção desta rede. A noção de distância considerada entre os pontos é a distância Euclidiana. Estes pontos adicionais que podem ser utilizados são chamados de **pontos (ou nós) de Steiner**. Representaremos nas figuras os pontos dados por nós pretos e os pontos de Steiner por nós brancos. Como o objetivo é encontrar a rede conexa de menor comprimento que conecta os pontos dados, não haverá ciclos na solução, uma vez que retirando-se uma das arestas do ciclo, o comprimento da rede seria menor. Portanto, a solução do problema será sempre uma árvore. A árvore que é solução do PSE é chamada **Árvore Mínima de Steiner**. Empregamos as notações de grafos para representar soluções do problema.

Um ponto de Steiner não pode ter grau 1 numa árvore, pois essa ligação a este ponto aumentaria o comprimento total da árvore, não sendo mínimo, portanto. Também não poderá ter grau 2, pois se este ponto de Steiner liga-se a 2 outros pontos distintos, estes poderiam ser unidos diretamente por uma aresta, encurtando o comprimento da árvore. Portanto, o grau de um ponto de Steiner numa árvore

mínima de Steiner é, no mínimo, 3.

Se duas arestas se encontram num ponto formando um ângulo menor que 120° , é possível obter 2 pontos, um em cada extremidade da aresta, e localizar um novo ponto de Steiner (o ponto de Torricelli em relação a estes 3 pontos), encurtando o comprimento total da árvore. A Figura 2.4 mostra o novo ponto de Steiner e as arestas pontilhadas a árvore de comprimento mais curto que conecta os pontos. Logo, duas arestas em uma árvore mínima de Steiner não formam ângulo menor do que 120° .

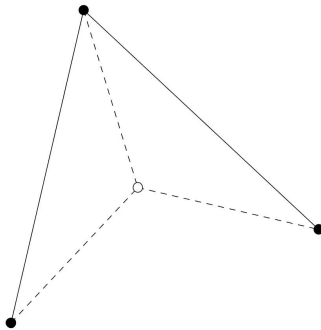


Figura 2.4: Árvore mais curta com a inserção de um ponto de Steiner

Se em um ponto de Steiner incidem duas arestas com ângulo maior do que 120° e este ponto possuir grau maior do que 3, então algum par de arestas incidentes terá ângulo menor do que 120° , resultando que a árvore não será mínima. Logo, os ângulos formados pelas arestas incidentes a um ponto de Steiner numa árvore mínima de Steiner têm medida exatamente igual a 120° . Como consequência destas observações, o grau de um ponto de Steiner numa árvore mínima é igual a 3.

A partir disto enunciamos algumas propriedades básicas da árvore

- (P1) Um ponto de Steiner em uma árvore mínima de Steiner tem grau exatamente igual a 3.
- (P2) As arestas que conectam o ponto de Steiner aos nós adjacentes formam ângulos de 120° entre si e se situam em um plano.

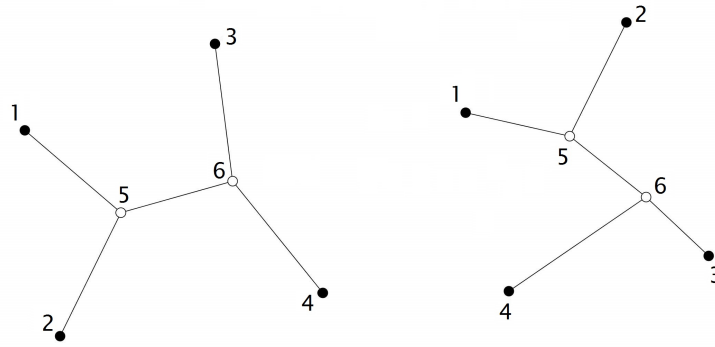


Figura 2.5: Árvores com a mesma topologia: 1 – 5, 2 – 5, 3 – 6, 4 – 6, 5 – 6

(P3) Um nó terminal tem grau entre 1 e 3.

(P4) Uma árvore mínima de Steiner com p pontos terminais possui, no máximo, $p - 2$ pontos de Steiner.

As demonstrações destas propriedades podem ser encontradas com mais detalhes em [4].

Definição 2.2.1. *Uma árvore que satisfaz estas propriedades é chamada de **Árvore de Steiner**. A **Árvore Mínima de Steiner (AMS)** é a árvore de Steiner de menor comprimento.*

A seguir apresentamos definições que são utilizadas nos algoritmos na busca pela Árvore Mínima de Steiner.

Definição 2.2.2. *Dada uma árvore de Steiner, a **topologia** associada à esta árvore, é a fixação do número de pontos de Steiner e das arestas (ligações) entre os nós da árvore. Não necessariamente é conhecida a posição geométrica dos pontos de Steiner.*

Na Figura 2.5 temos um exemplo de árvores com a mesma topologia. Os nós fixos são $\{1, 2, 3, 4\}$, os nós de Steiner $\{5, 6\}$ e as ligações 1 – 5, 2 – 5, 3 – 6, 4 – 6, 5 – 6.

Definição 2.2.3. *Uma topologia é dita **topologia de Steiner** se cada ponto de Steiner tem grau 3 e cada nó terminal tem grau menor ou igual a 3. Uma topologia*

de Steiner com p pontos terminais é dita uma **topologia cheia de Steiner** se existem $p - 2$ pontos de Steiner. A árvore associada à topologia cheia de Steiner é chamada de **árvore cheia de Steiner**.

Na Figura 2.6 temos uma representação de uma topologia cheia.

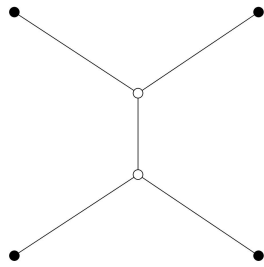


Figura 2.6: Representação de uma topologia cheia com 4 nós dados e 2 nós de Steiner

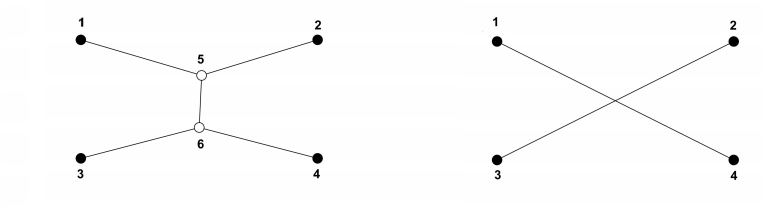
São consideradas operações sobre a árvore de Steiner a **retirada ou encolhimento** de uma aresta, quando esta é retirada da árvore e os dois pontos ligados a ela se tornam um só ponto. A **divisão** de ponto é a introdução de uma aresta formando novas ligações com os pontos da árvore.

Definição 2.2.4. Uma topologia T é uma **degeneração** de uma topologia T' se T pode ser obtida por meio de retirada de arestas da topologia T' . Uma árvore de Steiner que corresponde a uma dada topologia mas com arestas de comprimento zero é dita ter **topologia degenerada**. É possível mostrar que qualquer árvore mínima de Steiner com uma topologia não cheia pode ser associada a uma topologia cheia de Steiner degenerada.

Definição 2.2.5. Para uma topologia T , define-se a **árvore relativamente mínima** como a árvore de menor comprimento com relação à topologia T . Esta árvore pode existir ou não dependendo da topologia.

Por exemplo, se considerarmos os 4 vértices de um retângulo como 4 pontos dados e a seguinte topologia T : 1 - 5, 2 - 5, 3 - 6, 4 - 6, 5 - 6, não existe para esta topologia

a árvore relativamente mínima, uma vez que a árvore de menor comprimento ocorre quando a distância entre os pontos 5 e 6 é zero. Mas isto gera uma topologia diferente da topologia T, pois temos 1 - 4, 2 - 3. Esta situação está ilustrada na Figura 2.2.



Uma árvore mínima de Steiner deve ser também uma árvore relativamente mínima com relação a sua topologia.

Alguns resultados que podem ser encontrados em [4]:

- (P5) Numa topologia cheia, o grau de cada nó terminal é 1.
- (P6) Para $n \geq 4$, existem pelo menos dois pontos de Steiner na árvore cheia de Steiner, na qual cada um é adjacente a dois terminais.
- (P7) Existe no máximo uma árvore de Steiner para uma dada topologia de Steiner.
- (P8) Existe no máximo uma árvore relativamente mínima para uma dada topologia.

Outro resultado importante é que toda topologia de Steiner se decompõe de forma única em topologias cheias de Steiner, logo uma árvore de Steiner se decompõe de modo único em árvores cheias de Steiner.

2.2.1 Soluções do Problema de Steiner

Melzak foi o primeiro a apresentar um algoritmo para resolver o PSE, com $n = 2$, baseado na enumeração de todas as topologias de Steiner e determinação da árvore de Steiner de menor comprimento correspondente a cada topologia. Winter [5] e Zachariensen [6] [7] criaram o melhor algoritmo para obter solução exata para o

PSE no plano, o Geosteiner, que resolve instâncias com milhares de nós terminais. Estes métodos não se aplicam em dimensões maiores, uma vez que, são consideradas somente 2 possibilidades para localizar o ponto no plano, enquanto que, em dimensões maiores, temos infinitas possibilidades de localizações.

No caso $n \geq 3$, Gilbert e Pollak propuseram enumerar todas as topologias de Steiner e calcular a árvore de menor comprimento associada a cada topologia. O fato é que, dados p pontos em \mathbb{R}^n e $k \in \{0, 1, 2, \dots, p - 2\}$ pontos de Steiner, eles provaram que o número total de topologias diferentes com k pontos de Steiner é

$$C_{p,k+2} = \frac{(p+k-2)!}{k!2^k}.$$

Significando que, por exemplo, para o problema com 10 pontos apenas, o número total de topologias cheias de Steiner será 2.027.025.

Smith propôs um algoritmo de Branch and Bound (B&B) para encontrar a árvore mínima de Steiner em \mathbb{R}^n , no qual se desenvolve um esquema de enumeração de todas as topologias cheias de um conjunto dado de p terminais. Para isso, ele utiliza uma estrutura chamada vetor topologia a qual identifica uma topologia cheia de Steiner. A árvore de B&B começa com o nó raiz sendo a única topologia cheia correspondente ao conjunto de três pontos terminais e cada nó em uma profundidade k na árvore B&B consiste em todas as topologias cheias de Steiner com $k + 3$ nós terminais. A ramificação é feita acrescentando-se um novo nó terminal que se liga por uma nova aresta e um novo ponto de Steiner a uma das arestas anteriormente existentes. O processo de minimização é feito através da resolução de um sistema. Smith mostrou que esta operação de acrescentar um nó e uma aresta nunca diminui o comprimento mínimo da árvore, portanto, se o comprimento mínimo de uma árvore de Steiner em algum nó da árvore B&B é maior do que o comprimento da menor árvore de Steiner já encontrada até então, o nó poderá ser removido da árvore.

Nos trabalhos citados, não foi apresentada uma formulação matemática para o problema de Steiner. Os artigos de Maculan, Xavier e Michelon [8] e Fampa e Maculan [9] apresentam modelos para o Problema de Steiner, os quais serão abordados

no próximo capítulo. Por se tratar de um problema não linear inteiro e NP-difícil, há poucos trabalhos na literatura sobre a solução exata do Problema de Steiner Euclidiano.

2.2.1.1 Heurísticas

Várias heurísticas foram propostas para o Problema Euclidiano de Steiner em dimensões $n \geq 3$. Para conjuntos de pontos fixos em \mathbb{R}^3 , a maioria das heurísticas encontradas têm como ideia central dividir o PSE em subproblemas, resolvê-los e concatenar as subárvores, de modo a obter uma aproximação da árvore de Steiner mínima. A heurística Kalpakis-Ravada-Sherman, em [10] e [11], divide o conjunto de nós terminais em subconjuntos com no máximo p pontos cada. Essa divisão é feita recursivamente utilizando a mediana das coordenadas dos pontos dados. Cada um destes subconjuntos é um subproblema e utiliza-se o algoritmo exato para resolvê-lo. As árvores mínimas de Steiner são conectadas através da árvore mínima geradora dos pontos dados e pontos de Steiner de cada partição. Smith-Weiss-Patel [12] também usaram esta ideia para determinar a árvore mínima de Steiner em \mathbb{R}^3 , utilizando uma estrutura chamada *sausage*.

O algoritmo de Chapeau-Blondeau et al. [13], denominado Relaxação Dinâmica, se baseia em um modelo físico para o Problema de Steiner Euclidiano. Partindo de uma configuração inicial de uma árvore de Steiner, esta evolui até um mínimo local do comprimento da árvore. Nesta heurística utiliza-se a idéia da evolução dinâmica de uma película de sabão, a qual relaxa sob forças de tensão superficial e estabiliza-se em uma configuração de equilíbrio, que equivale à minimização local do comprimento da árvore de Steiner. Os testes computacionais mostraram que é uma heurística rápida mas que não encontrou nenhuma solução ótima para 4 conjuntos de pequenas instâncias ($8 \leq p \leq 11$). Montenegro [14] fez a extensão desta heurística para dimensões maiores do que dois, chamada Relaxação Dinâmica Estendida. Em [15], Forte apresenta algoritmos baseados na metaheurística Busca Local Iterativa e nas heurísticas citadas sobre Relaxação Dinâmica. Para instâncias de pequeno porte, de

até 11 pontos, os algoritmos baseados na Busca Local Iterativa encontraram soluções ótimas em quantidade maior se comparadas com outros algoritmos da literatura. Os algoritmos baseados em Relaxação Dinâmica Estendida encontram um número expressivo de soluções ótimas.

Para dimensões maiores ou iguais a três, encontramos propostas de heurísticas e metaheurísticas, como a de Lundy, cujo algoritmo é baseado na metaheurística *Simulated Annealing*. Um algoritmo genético, considerando os vetores topologias como cromossomos, foi desenvolvido por Maculan e Montenegro [16].

2.2.2 Razão de Steiner

Uma medida usada para avaliar a qualidade da solução de uma heurística é a razão de Steiner. Como várias heurísticas utilizam a árvore geradora mínima como solução inicial e buscam a melhora desta para determinar a árvore mínima de Steiner, utiliza-se como critério para avaliar a performance da heurística a razão ρ definida por

$$\rho = \frac{|AS|}{|AGM|},$$

onde $|\cdot|$ é o comprimento de uma árvore, AS é árvore de Steiner e AGM é árvore geradora mínima para os p pontos dados.

Comparando a razão de Steiner ρ^* de uma árvore mínima de Steiner resultante de uma algoritmo exato e a razão de Steiner ρ_h de uma árvore de Steiner solução de uma heurística, temos que $\rho^* \leq \rho_h$.

Dada uma distribuição de pontos dados em \mathbb{R}^n , há uma conjectura sobre existir um limitante inferior para o menor valor que a razão de Steiner assume para tal distribuição. Gilbert and Pollak nunca encontraram uma razão menor do que $\frac{\sqrt{3}}{2}$. Definiram esse limitante por

$$\rho_n = \min_{AMS \in \mathbb{R}^n} \rho(AMS),$$

onde AMS é a árvore mínima de Steiner.

Até o momento a única afirmação que foi provada é que, para dimensões maiores que dois, tem-se que $\rho_n \geq 0.615827$.

Capítulo 3

Modelos

3.1 Modelos matemáticos do Problema de Steiner Euclidiano

3.1.1 Modelo 1

Em [8], Maculan, Michelon e Xavier formularam um problema de programação não convexo inteiro misto para o PSE. Considera-se um grafo especial $G = (V, E)$, onde $P = \{1, 2, \dots, p-1, p\}$ é o conjunto de índices associados aos pontos dados denotados por $x^1, x^2, \dots, x^{p-1}, x^p$ e $S = \{p+1, p+2, \dots, 2p-3, 2p-2\}$ o conjunto de índices associados aos pontos de Steiner, denotados por $x^{p+1}, x^{p+2}, \dots, x^{2p-3}, x^{2p-2}$. Seja $V = P \cup S$. Denote por $[i, j]$ uma aresta do grafo G , com $i, j \in V$ tal que $i < j$. Consideramos $E = E_1 \cup E_2$, onde $E_1 = \{[i, j] | i \in P, j \in S\}$ e $E_2 = \{[i, j] | i \in S, j \in S\}$.

Considera-se a distância Euclidiana entre pontos $x^i, x^j \in \mathbb{R}^n$, isto é, $\|x^i - x^j\| = \sqrt{\sum_{l=1}^n (x_l^i - x_l^j)^2}$. Define-se uma variável $y_{ij} \in \{0, 1\}$ para a aresta $[i, j] \in E$, onde $y_{ij} = 1$, se a aresta $[i, j]$ pertence à solução ótima do problema e $y_{ij} = 0$ caso contrário.

Com esta configuração, o grafo do modelo possui p pontos fixos e $p-2$ pontos de Steiner, que é o número máximo de pontos de Steiner (propriedade P4 da seção

2.2), ou seja, considera-se a princípio, uma topologia cheia (Definição 2.2.3).

Por exemplo, se a instância tiver 6 pontos fixos ($p = 6$), o grafo terá 4 pontos de Steiner e se configura da seguinte forma:

$G = (V, E)$, onde $V = P \cup S$, sendo $P = \{1, 2, 3, 4, 5, 6\}$ e $S = \{7, 8, 9, 10\}$. A localização de cada ponto fixo será denotada por x^1, \dots, x^6 e os de Steiner por x^7, \dots, x^{10} . Temos o conjunto de arestas E_1 que ligam os pontos fixos aos de Steiner, $E_1 = \{[1, 7], [1, 8], \dots, [1, 10], [2, 7], \dots, [2, 10], \dots, [6, 10]\}$ e E_2 o conjunto de arestas que ligam os pontos de Steiner entre si, $E_2 = \{[7, 8], [7, 9], [7, 10], [8, 9], [8, 10], [9, 10]\}$.

A Figura 3.1 mostra o grafo G deste exemplo.

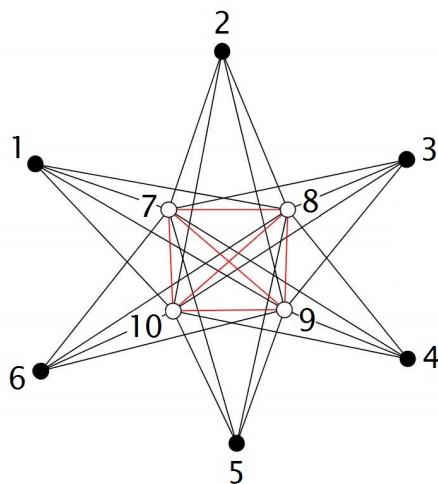


Figura 3.1: Grafo da instância com 6 pontos fixos e 4 de Steiner

Temos, então o seguinte modelo

(P) :

$$\min \quad \sum_{[i,j] \in E} \|x^i - x^j\| y_{ij} \quad (3.1)$$

$$\text{s.a.:} \quad x^i \in \mathbb{R}^n, \quad i \in S \quad (3.2)$$

$$\sum_{j \in S} y_{ij} = 1, \quad i \in P \quad (3.3)$$

$$\sum_{i \in P} y_{ij} + \sum_{k < j, k \in S} y_{kj} + \sum_{k > j, k \in S} y_{jk} = 3, \quad j \in S \quad (3.4)$$

$$\sum_{k < j, k \in S} y_{kj} = 1, \quad j \in S - \{p+1\} \quad (3.5)$$

$$y_{ij} \in \{0, 1\}, \quad [i, j] \in E \quad (3.6)$$

A função objetivo é a soma das distâncias entre os nós, ou seja, comprimentos das ligações do tipo nó fixo-nó de Steiner e do tipo Steiner-Steiner. As variáveis de decisão de (P) são $x_i \in \mathbb{R}^n$ (3.2) e $y_{ij} \in \{0, 1\}$ (3.6). (P) é formulado de modo que uma solução ótima seja uma topologia cheia de Steiner para p pontos dados em \mathbb{R}^n . As restrições (3.2) indicam que os pontos de Steiner, que são variáveis do problema, estão no \mathbb{R}^n . As restrições (3.3) determinam que o grau de cada um dos vértices terminais do problema seja igual a 1. Voltando ao exemplo com 6 pontos fixos, temos as restrições

$$\begin{aligned} y_{1,7} + y_{1,8} + y_{1,9} + y_{1,10} &= 1 \\ y_{2,7} + y_{2,8} + y_{2,9} + y_{2,10} &= 1 \\ &\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ y_{6,7} + y_{6,8} + y_{6,9} + y_{6,10} &= 1 \end{aligned}$$

A primeira restrição deste conjunto, $y_{1,7} + y_{1,8} + y_{1,9} + y_{1,10} = 1$, força que o ponto fixo 1 seja ligado a somente um dos 4 pontos de Steiner, por exemplo, caso a aresta $[1, 8]$ esteja na solução do problema, a variável binária $y_{1,8}$ é igual a 1 e as outras variáveis na equação serão iguais a zero, de modo que o ponto fixo 1 terá grau 1. De forma análoga, as demais restrições do conjunto obrigam que cada um dos pontos fixos tenha grau 1.

As restrições (3.4) determinam que cada vértice de Steiner tenha grau igual a 3.

Exemplificando com a instância de 6 pontos fixos, temos, para $j = 8$, uma restrição deste tipo sendo

$$y_{1,8} + y_{2,8} + y_{3,8} + y_{4,8} + y_{5,8} + y_{6,8} + y_{7,8} + y_{8,9} + y_{8,10} = 3$$

Quando esta restrição é satisfeita, apenas 3 das variáveis binárias que representam as ligações do ponto de Steiner 8 aos demais vértices do grafo serão iguais a 1, as outras serão iguais a 0. De modo que o nó de Steiner 8 terá grau 3.

As restrições (3.5) evitam a formação de subciclos no grafo, pois a solução deve ser uma árvore. Cada solução viável y_{ij} que satisfaz as restrições do modelo, estão associadas a um subgrafo conexo de G , para o qual, cada vértice tem grau 1 ou 3. Também observa-se que todas as topologias de Steiner são isomorfas às soluções viáveis geradas pelas restrições.

Uma propriedade em [4] afirma que os pontos de Steiner sempre estão localizados dentro da envoltória convexa dos p pontos terminais, portanto, sendo M a maior distância entre os pontos fixos, $M = \max\{\|x^i - x^j\|, \text{ para } 1 \leq i < j \leq p\}$, então temos que $\|x^i - x^j\| \leq M$ para todo $[i, j] \in E$.

3.1.2 Modelo 2

No artigo [9], Fampa e Maculan desenvolvem outro modelo, considerando a construção do grafo do modelo anterior e a notação $a^1, a^2, \dots, a^{p-1}, a^p$ para o conjunto de pontos fixos no \mathbb{R}^n , temos o seguinte modelo

$$\min \quad \sum_{[i,j] \in E} d_{ij} \quad (3.7)$$

$$\text{s.a.: } d_{ij} \geq \|a^i - x^j\| - M(1 - y_{ij}), \quad [i, j] \in E_1, \quad (3.8)$$

$$d_{ij} \geq \|x^i - x^j\| - M(1 - y_{ij}), \quad [i, j] \in E_2, \quad (3.9)$$

$$d_{ij} \geq 0, \quad [i, j] \in E, \quad (3.10)$$

$$\sum_{j \in S} y_{ij} = 1, \quad i \in P \quad (3.11)$$

$$\sum_{i < j, i \in S} y_{ij} = 1, \quad j \in S - \{p + 1\}, \quad (3.12)$$

$$y_{ij} \in \{0, 1\}, \quad [i, j] \in E, \quad (3.13)$$

$$d_{ij} \in \mathbb{R}_+, \quad [i, j] \in E, \quad (3.14)$$

$$x^i \in \mathbb{R}^n, \quad i \in S \quad (3.15)$$

Neste modelo, as variáveis de decisão são $x_i \in \mathbb{R}^n$, que indica a localização no espaço dos pontos de Steiner, e d_{ij} , variável não-negativa que representa no modelo a "distância" entre pontos (fixos ou de Steiner) de índice i e índice j . As restrições (3.11) a (3.13) coincidem com as restrições (3.3), (3.5) e (3.6). As restrições (3.8) e (3.9) utilizam a constante "big-M", onde $M = \max\{\|a^i - a^j\|, \text{ para } 1 \leq i < j \leq p\}$ é a maior distância entre os vértices fixos do problema. Quando $y_{ij} = 1$, o termo com M é eliminado e temos a restrição $d_{ij} \geq \|a^i - x^j\|$, caso $y_{ij} = 0$, a restrição $d_{ij} \geq \|a^i - x^j\| - M$ se torna redundante para M suficientemente grande. O mesmo ocorre em relação a restrição (3.9).

Além disso, é demonstrado que

(P5) Se $\bar{x}^j \in \mathbb{R}^n$, $j \in S$ e $\bar{y}_{ij} \in \{0, 1\}$, $[i, j] \in E$ é uma solução ótima do problema, então temos $d_{ij} = \|a^i - \bar{x}^j\|$ ou $d_{ij} = 0$, para todo $[i, j] \in E_1$ e $d_{ij} = \|x^i - \bar{x}^j\|$ ou $d_{ij} = 0$, para todo $[i, j] \in E_2$.

(P6) $y_{ij} \in \{0, 1\}$, $(i, j) \in E$ é associado com uma topologia cheia de Steiner se, e só

se as seguintes equações são satisfeitas

$$\sum_{j \in S} y_{ij} = 1, \quad i \in P \quad (3.16)$$

$$\sum_{k < j, k \in S} y_{kj} = 1, \quad j \in S - \{p + 1\} \quad (3.17)$$

$$\sum_{i \in P} y_{ij} + \sum_{k < j, k \in S} y_{kj} + \sum_{k > j, k \in S} y_{jk} = 3, \quad j \in S \quad (3.18)$$

A demonstração destas afirmações pode ser encontrada em [9].

Estas propriedades garantem que (3.7) - (3.15) é uma formulação para o PSE. Se considerarmos uma solução 0-1 qualquer de (3.16) e (3.17), observamos que esta solução está associada a um subgrafo conexo de G . As propriedades (P1) a (P5) garantem que a solução obtida ao resolver o modelo é uma árvore de Steiner, pois, na solução ótima, os pontos de Steiner terão grau 3, caso contrário, a solução não corresponderá a menor árvore com $p - 2$ pontos de Steiner. Desta forma, não é necessário acrescentar a restrição (3.18) ao modelo.

Com o objetivo de obter limites inferiores para o valor ótimo da solução deste problema, a restrição de integralidade é relaxada, obtendo-se um problema convexo, mas não diferenciável em todos os pontos dados. Os autores reescrevem o modelo, tornando-o um problema de programação convexa na forma cônica. Com técnicas de algoritmos de pontos interiores, eles mostram que é possível obter uma solução ótima aproximada para esta relaxação em tempo polinomial.

3.1.3 Estudo Teórico sobre Redução do Big-M

O modelo proposto por Fampa e Maculan é não diferenciável e utiliza a constante "Big-M", denotada no modelo por M , a maior distância entre os pontos fixos dados. Realizamos estudos objetivando diminuir essa constante Big-M, pois um valor do Big-M elevado pode gerar dificuldades para o método de solução para (P), como levar a um grande número de iterações ou propagação de erros numéricos. Temos o seguinte resultado

Teorema: Seja o Problema Euclidiano de Steiner com n nós terminais. Suponha

que S é uma árvore de Steiner ótima para o problema, com topologia cheia (degenerada ou não). Considere $d_1 \geq d_2 \geq d_3 \geq \dots \geq d_{\frac{n(n-1)}{2}}$ as distâncias Euclidianas colocadas em ordem não crescente, entre todos os pares de nós terminais. Então o tamanho de todas as arestas da árvore S tem comprimento menor ou igual a d_{n-1} .

Demonstração: Seja e uma aresta da árvore S . Eliminando esta aresta da árvore, geramos duas árvores desconexas T_1 e T_2 . É possível conectar novamente as duas árvores conectando dois nós terminais, um na árvore T_1 e outro na árvore T_2 . O menor número de possíveis conexões é igual a $n - 1$ e ocorre quando uma das árvores tem apenas um nó terminal. Supondo o pior caso, no qual estas $n - 1$ conexões têm comprimentos d_1, d_2, \dots, d_{n-1} , podemos conectar novamente as duas árvores com uma aresta de comprimento d_{n-1} , logo a aresta e que foi eliminada da árvore ótima S não pode ter comprimento maior do que d_{n-1} .

Iniciamos um estudo sobre problemas de programação disjuntiva generalizada, uma vez que esta classe de problemas tem sido apresentada como um modelo alternativo para problemas MINLP, usando disjunções e proposições lógicas. Analisamos a possibilidade de escrever o modelo (P) como um problema de programação disjuntiva, pois no artigo de Lee e Grossmann [17], eles realizaram um estudo para alguns problemas do tipo MINLP da área de química, cujas restrições da relaxação constituem uma envoltória convexa mais "justa" do que a do problema escrito no modelo com Big-M [17]. Encontramos dificuldades para encontrar pacotes que pudessem resolver o problema de programação disjuntiva.

Capítulo 4

Algoritmos para Problemas de Programação Não Linear Inteiros Mistos

Otimização não linear inteira mista é um campo da ciência que vem sendo estudada por diversas áreas que necessitam de uma modelagem matemática mais adequada para seus problemas reais, tais como as áreas de processos químicos, engenharia de sistemas, redes, entre outras. Pesquisadores vêm se dedicando a desenvolver e implementar algoritmos para resolver problemas de programação não linear inteiro misto (MINLP) cujas variáveis são do tipo inteiras e reais. Devido ao desenvolvimento ao longo dos anos e às implementações bem sucedidas de algoritmos para problemas de programação linear inteira mista e programação não linear, os esforços dos pesquisadores caminharam naturalmente para a criação de algoritmos que combinassem as técnicas de ambas as áreas, produzindo avanços em relação aos problemas MINLP.

Os métodos conhecidos incluem técnicas baseados em Branch and Bound (BB) [18] e [19], Decomposição Generalizada de Benders (GDB), Aproximação Exterior (OA) e Plano de Corte Estendido. Estes métodos, em geral, resolvem subproblemas não lineares. Primeiramente, vejamos a forma geral de um MINLP.

4.1 Forma Geral do MINLP

Considere o problema não linear inteiro misto (MINLP) no seguinte formato

$$\left\{ \begin{array}{l} \min \quad f(x, y) \\ \text{s.a} \quad g_i(x, y) \leq 0, \quad i = 1, \dots, q \\ \quad \quad h_i(x, y) = 0, \quad i = 1, \dots, l \\ \quad \quad x \in X \subseteq \mathbb{R}^n, \quad y \in Y \subseteq \mathbb{Z}^m \end{array} \right. \quad (4.1)$$

onde X é um conjunto não vazio convexo compacto em \mathbb{R}^n , Y é um conjunto polidral de pontos inteiros finito em \mathbb{Z}^m . Consideramos também uma instância deste problema onde $f : X \times Y \rightarrow \mathbb{R}$ e $g_i : X \times Y \rightarrow \mathbb{R}$ $i = 1, \dots, q$ são funções convexas e diferenciáveis e $h_i : X \times Y \rightarrow \mathbb{R}$, $i = 1, \dots, l$ é linear. Sejam $g = (g_1, \dots, g_i, \dots, g_q)$ e $h = (h_1, \dots, h_i, \dots, h_l)$. Na maioria das aplicações encontramos o problema (MINLP) somente com as restrições de desigualdade:

$$(\text{MINLP}_1) \left\{ \begin{array}{l} \min \quad f(x, y) \\ \text{s.a} \quad g(x, y) \leq 0, \quad i = 1, \dots, q \\ \quad \quad x \in X \subseteq \mathbb{R}^n, \quad y \in Y \subseteq \mathbb{Z}^m \end{array} \right. \quad (4.2)$$

As estratégias para abordar este tipo de problema são:

- Relaxar a variável discreta, tornando-a contínua, obtendo uma relaxação não linear contínua do $(\text{MINLP})_1$ com região viável convexa. Ao resolver este subproblema, obtemos um limite inferior para o $(\text{MINLP})_1$;
- Fixar um valor para a variável inteira y , obtendo um subproblema não linear de variáveis contínuas em x , o que fornece um limite superior para o $(\text{MINLP})_1$.
- Construir para a função f e para g_i 's funções lineares ou convexas que aproximam estas funções, coincidindo em alguns pontos, resultando em problemas de programação inteira mista linear ou convexa, os quais resultam em limites inferiores para $(\text{MINLP})_1$.

A seguir, descreveremos algoritmos para resolver o (MINLP) [20] que são baseados nas estratégias mencionadas anteriormente.

4.2 Método Branch and Bound

Relaxando a integralidade da variável y , podemos considerar $\alpha \leq y \leq \beta$ e obtemos o seguinte problema de programação não-linear

$$(NLP) \left\{ \begin{array}{l} \min \quad f(x, y) \\ \text{s.a} \quad g(x, y) \leq 0 \\ \quad \quad h(x, y) = 0 \\ \quad \quad x \in X \subseteq \mathbb{R}^n, \quad y \in \text{conv}(Y) \end{array} \right. \quad (4.3)$$

Supondo que a solução ótima de cada subproblema de (NLP) é um ponto regular, isto é, os vetores gradientes das restrições ativas são linearmente independentes, e considerando as hipóteses sobre as funções f , g e h , então temos que qualquer solução local de (NLP) é uma solução global e esta solução é obtida pela aplicação das condições de KKT.

O procedimento de Branch and Bound é análogo ao método conhecido para problemas de programação inteira, a ramificação da árvore é feita relaxando a integralidade da variável y , impondo limites inferior e superior para cada componente de y . Resolvendo o subproblema em cada nó k , obtemos um limite inferior Z^k e um limite superior U^k .

4.3 Decomposição Generalizada de Benders

Esta é uma técnica conhecida para o problema linear inteiro misto. Vamos abordar a técnica para resolver o problema (MINLP)₁. Seja

$$S = \{(x, y) \in X \times Y | g(x, y) \leq 0\}$$

e

$$V = \{y \in Y \mid \text{existe } x \in X \text{ tal que } g(x, y) \leq 0\}$$

Para todo $y \in Y$, considere o seguinte problema não linear

$$(\text{NLP}(y)) \begin{cases} \min & f(x, y) \\ \text{s.a.} & g(x, y) \leq 0 \\ & x \in X. \end{cases} \quad (4.4)$$

Note que a solução ótima de $(\text{NLP}(y))$ é uma solução viável de $(\text{MINLP})_1$, logo o valor ótimo $v(\text{NLP}(y))$ é um limite superior para $(\text{MINLP})_1$. Suponha que para todo $y \in V$, uma solução ótima de $(\text{NLP}(y))$ é um ponto regular.

A função objetivo do problema dual associado a $(\text{NLP}(y))$ é

$$d_y(\lambda) = \min_{x \in X} L(x, y, \lambda) = f(x, y) + \lambda^T g(x, y) \quad (4.5)$$

onde $\lambda \in \mathbb{R}_+^q$, e portanto, o problema dual Lagrangeano de $(\text{NLP}(y))$ é

$$(D_y) \max_{\lambda \in \mathbb{R}_+^q} d_y(\lambda) \quad (4.6)$$

Assim, temos

$$\begin{aligned} \min_{(x,y) \in S} f(x, y) &\equiv \min_{y \in V} v(\text{NLP}(y)) \\ &\equiv \min_{y \in V} \left(\max_{\lambda \in \mathbb{R}_+^q} \min_{x \in X} L(x, y, \lambda) \right) \\ &\equiv \begin{cases} \min \alpha \\ \text{s.a.} & \alpha \geq \min_{x \in X} L(x, y, \lambda), \forall \lambda \geq 0 \\ & y \in V \end{cases} \end{aligned} \quad (4.7)$$

Como V é implícito, é necessário representá-lo explicitamente por certas restrições de desigualdade. Para todo $y \in Y$, considere o seguinte problema de verificação de viabilidade:

$$\min_{x \in X} \max\{g_1(x, y), \dots, g_q(x, y)\}, \quad (4.8)$$

o qual é equivalente a

$$(\text{NLPF}(y)) \begin{cases} \min & \beta \\ \text{s.a.} & \beta \geq g_i(x, y) \quad i = 1, \dots, q, \\ & x \in X. \end{cases} \quad (4.9)$$

Para todo $y \in Y$, $(\text{NLP}(y))$ é inviável se e somente se $(\text{NLPF}(y))$ tem valor ótimo β^* positivo. O dual lagrangeano de $(\text{NLPF}(y))$ é

$$(\text{DF}(y)) \begin{cases} \max & \min_{x \in X} \mu^T g(x, y) \\ \text{s.a.} & \mu \in \Lambda = \left\{ \sum_{i=1}^q \mu_i = 1, \mu_i \geq 0, i = 1, \dots, q \right\} \end{cases} \quad (4.10)$$

Portanto, $y \in V$ pode ser caracterizado pelas restrições de desigualdades

$$0 \geq \min_{x \in X} \mu^T g(x, y), \quad \forall \mu \in \Lambda. \quad (4.11)$$

Incorporando (4.11) em (4.7), obtemos o seguinte problema mestre

$$(\text{MGDB}) \begin{cases} \min & \alpha \\ \text{s.a.} & \alpha \geq \min_{x \in X} L(x, y, \lambda), \quad \forall \lambda \geq 0, \\ & 0 \geq \min_{x \in X} \mu^T g(x, y), \quad \forall \mu \in \Lambda, \\ & y \in Y. \end{cases} \quad (4.12)$$

Teorema 4.1. [20] *O problema (MGDB) é equivalente ao problema (MINLP_1) .*

Para obtermos um problema linear inteiro misto, considere a seguinte relaxação de (MGDB):

$$(\text{MGDB}_k) \begin{cases} \min & \alpha \\ \text{s.a.} & \alpha \geq L(x^i, y^i, \lambda^i) + \nabla_y^T L(x^i, y^i, \lambda^i)(y - y^i), \quad i \in I^k, \\ & 0 \geq (\mu^i)^T [g(x^i, y^i) + \nabla_y^T g(x^i, y^i)(y - y^i)], \quad i \in J^k, \\ & y \in Y, \end{cases} \quad (4.13)$$

onde (x^i, λ^i) é o par primal-dual ótimo de $(\text{NLP}(y^i))$ se $(\text{NLP}(y^i))$ é viável e (x^i, μ^i) é o par primal-dual ótimo de $(\text{NLPF}(y))$ se $(\text{NLP}(y^i))$ é inviável, $i = 1, \dots, k$ e $|I^k \cup J^k| = k$. (MGDB_k) é uma relaxação do problema (MGBD) e produz um limite inferior para (MGBD) e a solução y^{k+1} de (MGDB_k) pode ser usada para gerar o problema $(\text{NLP}(y^{k+1}))$ na iteração seguinte.

4.3.1 Algoritmo da Decomposição Generalizada de Benders para (MINLP_1)

Passo 0:

- Escolha $y^1 \in Y$. Faça $LB^0 = -\infty$, $UB^0 = +\infty$, $I^0 = J^0 = \emptyset$, $k = 1$.

Passo 1:

- Resolva $(\text{NLP}(y^k))$.
 - Se $(\text{NLP}(y^k))$ é viável, obtemos uma solução ótima x^k e um multiplicador de Lagrange ótimo λ^k . Faça $I^k = I^{k-1} \cup \{k\}$ e $J^k = J^{k-1}$. Faça $UB^k = \min\{UB^{k-1}, f(x^k, y^k)\}$. Se $UB^k = f(x^k, y^k)$, faça $(x^*, y^*) = (x^k, y^k)$.
 - Se $(\text{NLP}(y^k))$ é inviável, resolva $(\text{NLPF}(y^k))$ e obtenha uma solução ótima x^k e um multiplicador de Lagrange μ^k . Faça $J^k = J^{k-1} \cup \{k\}$ e $I^k = I^{k-1}$.

Passo 2:

- Resolva o problema mestre (MGBD_k) e obtenha uma solução ótima (α^k, y^{k+1}) . Faça $LB^k = \alpha^k$.
- Se $LB^k \geq UB^k$, PARE e (x^*, y^*) é uma solução ótima para (MINLP_1) .
- Senão, faça $k:=k+1$ e volte ao PASSO 1.

Teorema 4.2. [20] *O algoritmo 1 pára em uma solução ótima (x^*, y^*) do problema (MINLP_1) em um número finito de iterações.*

A demonstração deste teorema pode ser encontrada em [20].

4.4 Método da Aproximação Externa

A idéia deste método é alternar a resolução de um subproblema de programação não linear (NLP) e a resolução de um subproblema-mestre linear inteiro misto (MILP), assim como ocorre na Decomposição Generalizada de Benders. A função objetivo e as restrições são linearizadas e constrói-se uma relaxação (MILP) do problema. A diferença está na forma de gerar o subproblema-mestre. Supondo as mesmas hipóteses para o desenvolvimento do algoritmo anterior e os conjuntos S e V definidos como na seção anterior, queremos resolver o problema $(\text{MINLP})_1$. Para todo $y^i \in V$, seja x^i uma solução ótima do problema $(\text{NLP}(y_i))$. Temos,

$$\begin{aligned}
 \min_{(x,y) \in S} f(x, y) &= \min_{y^i \in V} \min_{x \in X} \{f(x, y^i) | g(x, y^i) \leq 0\} \\
 &= \min_{y^i \in V} \min_{x \in X} f(x^i, y^i) + \nabla^T f(x^i, y^i) \begin{pmatrix} x - x^i \\ 0 \end{pmatrix} \\
 \text{s.a.} \quad &g(x^i, y^i) + \nabla g(x^i, y^i) \begin{pmatrix} x - x^i \\ 0 \end{pmatrix} \leq 0 \\
 &x \in X \\
 &= \min_{y^i \in V} \min \alpha \tag{4.14} \\
 \text{s.a.} \quad &\alpha \geq f(x^i, y^i) + \nabla^T f(x^i, y^i) \begin{pmatrix} x - x^i \\ 0 \end{pmatrix} \\
 &0 \geq g(x^i, y^i) + \nabla g(x^i, y^i) \begin{pmatrix} x - x^i \\ 0 \end{pmatrix} \\
 &x \in X, \alpha \in \mathbb{R}
 \end{aligned}$$

Seja $T = \{i | y^i \in V \text{ e } x^i \text{ resolve } (\text{NLP}(y^i))\}$. Considere o seguinte problema-mestre

$$(MOAV) \left\{ \begin{array}{l} \min \quad \alpha \\ \text{s.a.} \quad \alpha \geq f(x^i, y^i) + \nabla^T f(x^i, y^i) \begin{pmatrix} x - x^i \\ y - y^i \end{pmatrix} \quad i \in T, \\ \\ 0 \geq g(x^i, y^i) + \nabla^T g(x^i, y^i) \begin{pmatrix} x - x^i \\ y - y^i \end{pmatrix} \quad i \in T, \\ \\ x \in X, y \in V, \alpha \in \mathbb{R} \end{array} \right.$$

Teorema 4.3. [20] *O problema (MOAV) é equivalente ao problema (MINLP₁).*

Demonstração. Seja (x^*, y^*) uma solução ótima de (MINLP₁), logo (α^*, x^*, y^*) é uma solução ótima de (4.14), com $\alpha^* = f(x^*, y^*)$. Como as funções f e g são convexas, para todo $i \in T$, temos $\alpha \geq f(x^i, y^i)$ e $0 \geq g(x^i, y^i)$, portanto, (α, x^i, y^i) é um ponto viável para (MOAV). Segue que $\hat{\alpha} = v(MOAV) \leq \alpha^*$. Como existe i tal que $(x^i, y^i) = (x^*, y^*)$, segue da primeira restrição de (4.14) que $\hat{\alpha} \geq f(x^*, y^*) = \alpha^*$. \square

Agora vamos representar V por um conjunto de restrições de (x, y) e relaxar o conjunto de índices T obtendo de forma iterativa os pontos (x^i, y^i) . Para todo $y \in Y$, considere o problema de viabilidade (NLPF(y)), definido anteriormente.

Lema 4.1. *Seja $y^i \in Y$ tal que (NLPF(y^i)) é inviável. Seja x^i a solução ótima do problema de verificação de viabilidade (NLPF(y^i)). Então y^i é inviável ao seguinte sistema de desigualdades*

$$0 \geq g_j(x^i, y^i) + \nabla^T g_j(x^i, y^i) \begin{pmatrix} x - x^i \\ y - y^i \end{pmatrix}, j = 1, \dots, q \quad (4.15)$$

para todo $x \in X$.

Demonstração. Suponha por absurdo que y^i é viável e satisfaz a desigualdade (4.15) para algum $\tilde{x} \in X$. Então,

$$0 \geq g_j(x^i, y^i) + \nabla_x^T g_j(x^i, y^i)(\tilde{x} - x^i), j = 1, \dots, q \quad (4.16)$$

Como x^i é solução ótima de $(\text{NLPF}(y^i))$, pelas condições de KKT existem multiplicadores ótimos μ_j , $j = 1, \dots, q$ tais que

$$\begin{aligned} \sum_{j=1}^q \mu_j \nabla_x g_j(x^i, y^i) &= 0, \\ \sum_{j=1}^q \mu_j &= 1, \\ \mu_j &\geq 0, \quad \forall j = 1, \dots, q \end{aligned} \tag{4.17}$$

Multiplicando (4.16) e (4.17) por μ_j e somando para $j = 1, \dots, q$, obtemos que

$$0 \geq \sum_{j=1}^q \mu_j \nabla_x g_j(x^i, y^i)$$

Por outro lado, como x^i é solução ótima de $(\text{NLPF}(y^i))$ e $(\mu_1, \dots, \mu_q)^T$ é solução ótima do problema dual $(DF(y^i))$, segue do Teorema de Dualidade Forte

$$\alpha^* = \sum_{j=1}^q \mu_j g_j(x^i, y^i)$$

onde α^* é solução ótima de $(\text{MLPF}(y^i))$. Logo, $\alpha^* \leq 0$, o que contradiz a inviabilidade de $(\text{NLP}(y^i))$. \square

Construímos o problema-mestre, considerando F o conjunto de índices de todos os $y^i \in Y$ tais que $(\text{NLP}(y^i))$ é inviável. A restrição (4.15) exclui todos os $y^i \in F$. Incorporando a restrição (4.15) ao problema (MOAV) e substituindo V por Y , temos

$$(\text{MOA}) \left\{ \begin{array}{l} \min \quad \alpha \\ \text{s.a.} \quad \alpha \geq f(x^i, y^i) + \nabla^T f(x^i, y^i) \begin{pmatrix} x - x^i \\ y - y^i \end{pmatrix} \quad i \in T, \\ 0 \geq g(x^i, y^i) + \nabla^T g(x^i, y^i) \begin{pmatrix} x - x^i \\ y - y^i \end{pmatrix} \quad i \in T, \\ 0 \geq g(x^i, y^i) + \nabla^T g(x^i, y^i) \begin{pmatrix} x - x^i \\ y - y^i \end{pmatrix} \quad i \in F, \\ x \in X, y \in Y, \alpha \in \mathbb{R} \end{array} \right.$$

A cada iteração, substituímos os pontos $x^i, i \in T$ e $i \in F$ em (MOA), pelos pontos dos conjuntos

$$T^k = \{i | y^i \in V \text{ e } x^i \text{ resolve NLP}(y^i), i = 1, \dots, k\} \text{ e}$$

$$F^k = \{i | \text{NLP}(y^i) \text{ é inviável e } x^i \text{ resolve NLPF}(y^i), i = 1, \dots, k\}$$

$$(MOA_k) \left\{ \begin{array}{l} \min \alpha \\ \text{s.a. } \alpha \geq f(x^i, y^i) + \nabla^T f(x^i, y^i) \begin{pmatrix} x - x^i \\ y - y^i \end{pmatrix} \quad i \in T^k, \\ 0 \geq g(x^i, y^i) + \nabla^T g(x^i, y^i) \begin{pmatrix} x - x^i \\ y - y^i \end{pmatrix} \quad i \in T^k, \\ 0 \geq g(x^i, y^i) + \nabla^T g(x^i, y^i) \begin{pmatrix} x - x^i \\ y - y^i \end{pmatrix} \quad i \in F^k, \\ x \in X, y \in Y, \alpha \in \mathbb{R} \end{array} \right.$$

4.4.1 Algoritmo de Aproximação Exterior para (MINLP₁)

Passo 0:

- Escolha $y^1 \in Y$. Faça $LB^0 = -\infty, UB^0 = +\infty, T^0 = F^0 = \emptyset, k = 1$.

Passo 1:

- Resolva (NLP(y^k)).
 - Se (NLP(y^k)) é viável, obtemos uma solução ótima x^k e um multiplicador ótimo λ^k . Faça $T^k = T^{k-1} \cup \{k\}$ e $J^k = J^{k-1}$. Faça $UB^k = \min\{UB^{k-1}, f(x^k, y^k)\}$. Se $UB^k = f(x^k, y^k)$, faça $(x^*, y^*) = (x^k, y^k)$.
 - Se (NLP(y^k)) é inviável, resolva (NLPF(y^k)) e obtenha uma solução ótima x^k . Faça $F^k = F^{k-1} \cup \{k\}$.

Passo 2:

- Resolva o problema mestre (MOA_k) e obtenha uma solução ótima $(\alpha^k, x^{k+1}, y^{k+1})$. Faça $LB^k = \alpha^k$.

- Se $LB^k \geq UB^k$, PARE e (x^*, y^*) é uma solução ótima para (MINLP_1) .
- Senão, faça $k:=k+1$ e volte ao PASSO 1.

4.5 Algoritmo Híbrido

Em [21], Bonami et al. desenvolveram e implementaram um algoritmo híbrido, no qual o algoritmo de Branch and Bound e o de Aproximação Exterior são os casos extremos. Eles implementaram 3 algoritmos:

- Branch and Bound para programação não linear;
- Aproximação Exterior;
- Híbrido baseado em Branch and Cut e Aproximação Exterior;

O algoritmo de aproximação exterior apresentado é descrito da seguinte forma.

Considerando o problema de programação não linear

$$(\text{NLP}) \left\{ \begin{array}{l} \min \quad f(x, y) \\ \text{s.a} \quad g(x, y) \leq 0 \\ \quad \quad x \in X, \quad y \in \tilde{Y} \end{array} \right.$$

onde \tilde{Y} é um conjunto tal que (NLP) seja uma relaxação linear de (MINLP).

Seja

$$T = \{(x^1, y^1), \dots, (x^K, y^K)\}$$

o conjunto de pontos qualquer em $X \times Y$. O problema-mestre é

$$P^{OA}(T) \left\{ \begin{array}{l} \min \quad \alpha \\ \text{s.a} \quad \nabla f(x^K, y^K)^T \begin{pmatrix} x - x^K \\ y - y^K \end{pmatrix} + f(x^K, y^K) \leq \alpha \\ \nabla g(x^K, y^K)^T \begin{pmatrix} x - x^K \\ y - y^K \end{pmatrix} + g(x^K, y^K) \leq 0 \\ \forall (x^K, y^K) \in T \\ x \in X, \quad y \in Y \cap \mathbb{Z}^n, \\ \alpha \in \mathbb{R} \end{array} \right.$$

Passo 0:

- Faça $LB = -\infty$, $UB = +\infty$, $(x_0, y_0) :=$ solução ótima de NLP.
- $T = \{(x_0, y_0)\}$, $k:=1$;
- Escolha uma tolerância ϵ ;

Passo 1:

- Enquanto $UB - LB > \epsilon$ e $P^{OA}(T)$ é viável, faça
- Seja $(\hat{\alpha}, \hat{x}, \hat{y})$ a solução ótima de $P^{OA}(T)$
- $LB := \hat{\alpha}$;
 - Se $(NLP(\hat{y}))$ é viável, então,
 - * Seja $y^k := \hat{y}$ e x^k a solução ótima de $NLP(\hat{y})$;
 - * $UB := \min(UB, f(x^k, y^k))$;
 - Caso contrário, seja $y^k := \hat{y}$ e x^k a solução ótima do problema de viabilidade $(NLPF(\hat{y}))$;
 - $T := T \cup \{(x^k, y^k)\}$;
 - $k := k + 1$;

4.5.1 Algoritmo Branch and Cut baseado em Aproximação Exterior

A fim de obter melhor desempenho do método de Aproximação Exterior, Quesada e Grossmann propuseram um algoritmo que mescla o uso de programação linear e não linear com o método branch and cut, evitando a solução sequencial de vários MILP's. Uma simples árvore de busca é apresentada, durante a qual problemas de programação não linear são resolvidos e usados para progressivamente fornecer limites mais fortes para a relaxação MILP. Assim como no Algoritmo de Aproximação Exterior, o método utiliza a relaxação linear inteira mista. Porém, ao invés de resolver as aproximações sucessivas do problema mestre até a otimalidade, é realizado um procedimento branch and cut, onde a aproximação exterior é atualizada nos nós da árvore de busca. Para isso eles usam os problemas $(P_{\bar{Y}}^{OA}(T))$, definido a seguir, em $\bar{Y} \subseteq Y$ e $(NLP(\bar{y}))$, definidos anteriormente.

$$P_{\bar{Y}}^{OA}(T) \left\{ \begin{array}{l} \min \quad \alpha \\ \text{s.a} \quad \nabla f(x^K, y^K)^T \begin{pmatrix} x - x^K \\ y - y^K \end{pmatrix} + f(x^K, y^K) \leq \alpha \\ \nabla g(x^K, y^K)^T \begin{pmatrix} x - x^K \\ y - y^K \end{pmatrix} + g(x^K, y^K) \leq 0 \\ \forall (x^K, y^K) \in T \\ x \in X, \quad y \in \bar{Y} \cap \mathbb{Z}^n, \\ \alpha \in \mathbb{R} \end{array} \right.$$

Seja i um nó da árvore de busca e $Y^i \subseteq Y$ um conjunto viável poliédrico para y neste nó. A relaxação contínua $(P_{Y^i}^{OA}(T))$ é resolvida obtendo uma solução (x^*, y^*) e um limite inferior sobre o valor ótimo da subárvore cujo nó raiz é i . Se y^* é inteiro, uma solução viável para $(MINLP)_1$ pode existir neste nó e pode ser encontrada resolvendo-se $(NLP(y^*))$. Seja \bar{x} uma solução ótima para $(NLP)(y^*)$. Se (\bar{x}, y^*) é viável para P , o limite superior é atualizado. Caso contrário, $T := T \cup \{(\bar{x}, y^*)\}$.

A seguir temos o algoritmo deste Branch and cut com Aproximação exterior.

Mostra-se que o algoritmo converge para uma solução ótima em um tempo finito, sob as hipóteses de convexidade, diferenciabilidade e continuidade das funções.

Passo 0:

- Faça $UB = +\infty$, $(x_0, y_0) :=$ solução ótima de NLP.
- $T = \{(x_0, y_0)\}$, $Y^0 := Y$;
- Inicialize uma lista de nós ativos $\mathcal{L} := \{0\}$;
- $i := 0$;

Passo 1:

- Enquanto $\mathcal{L} \neq \emptyset$ faça
 - Escolha um nó em \mathcal{L} ;
 - Seja (α^k, x^k, y^k) uma solução ótima da relaxação $P_{Y^k}^{OA}(T)$;
 - Enquanto y^k é inteiro e $\alpha^k < UB$ faça
 - * Se $(\text{NLP}(y^k))$ é viável, então seja \bar{x} sua solução ótima;
 - * Caso contrário seja \bar{x} solução ótima do problema de viabilidade $(\text{NLPF}(y^k))$;
 - Se (\bar{x}, y^k) é viável para (MINLP_1) , então $UB := \min(UB, f(\bar{x}, y^k))$;
 - $T := T \cup \{(\bar{x}, y^k)\}$;
 - Seja (α^k, x^k, y^k) a solução ótima da relaxação atualizada $P_{Y^k}^{OA}(T)$
- Se $\alpha^k < UB$ (branch)
 - Então escolha uma variável y_j com uma componente fracionária y_j^k para **branching**
 - Faça $Y^{i+1} := Y^k \cap \{y \in \mathbb{R}^n : y_j \leq \lfloor y_j^k \rfloor\}$;
 - Faça $Y^{i+2} := Y^k \cap \{y \in \mathbb{R}^n : y_j \geq \lceil y_j^k \rceil\}$;
 - $\mathcal{L} := \mathcal{L} \cup \{i+1, i+2\} - \{k\}$;

- $i := i + 2$;
- Caso contrário (eliminação do nó por limites) $\mathcal{L} := \mathcal{L} - \{k\}$;

4.5.2 Híbrido

O algoritmo híbrido é uma versão avançada do algoritmo anterior. São resolvidos mais problemas (NLP) com o objetivo de reduzir o tamanho da árvore. Uma das maneiras é resolver relaxações (NLP(Y^k)) em nós adicionais da árvore. Se isto é feito em todos os nós da árvore de busca, então se torna o algoritmo Branch and Bound para problemas (NLP). Outra maneira é fazer uma enumeração parcial dos nós da árvore e aplicar iterações do tipo aproximação exterior nestes nós. No algoritmo isto é feito no nó raiz durante um tempo τ estipulado. Se τ é colocado como infinito, o algoritmo se reduz ao de aproximação exterior.

Passo 0:

- Faça $UB = +\infty$, $(x_0, y_0) :=$ solução ótima de NLP.
- $T = \{(x_0, y_0)\}$, $Y^0 := Y$;
- Enquanto o limite de tempo τ não é excedido, faça
 - Resolva $P^{OA}(T)$;
 - Se uma solução (x^*, y^*) de $P^{OA}(T)$ é encontrada, então
 - * Seja \bar{x} a solução ótima de NLP(y^*) ou de (NLPF(y^*)), se for inviável;
 - * Se (\bar{x}, y^*) é viável para (MINLP₁), então $UB := \min(UB, f(\bar{x}, y^*))$;
 - * $T := T \cup \{(\bar{x}, y^*)\}$;
- Inicialize uma lista de nós ativos $\mathcal{L} := \{0\}$, $i := 0$, $l := 0$, $L := 10$;
- Enquanto $\mathcal{L} \neq \emptyset$, faça
 - Escolha um nó k em \mathcal{L} ;
 - Se $l \equiv 1 \pmod{L}$, então (resolva a relaxação neste nó)

- * Seja (\bar{x}, \bar{y}) a solução ótima de $(\text{NLP}(Y^k))$;
 - * Se \bar{x} é inteiro, então $UB := \min(UB, f(\bar{x}, \bar{y}))$;
 - $T := T \cup \{(\bar{x}, \bar{y})\}$;
 - Aplique corte à relaxação de $P_{Y^k}^{QA}(T)$.
 - Se a relaxação é viável, seja (α^k, x^k, y^k) a solução ótima, caso contrário seja $\alpha = +\infty$;
 - Enquanto y^k é inteiro e $\alpha^k < UB$ faça
 - * Se $(\text{NLP}(y^k))$ é viável, então seja \bar{x} sua solução ótima;
 - * Caso contrário, seja \bar{x} solução ótima do problema de viabilidade $(\text{NLPF}(y^k))$;
 - Se (\bar{x}, y^k) é viável para (MINLP_1) , então $UB := \min(UB, f(\bar{x}, y^k))$;
 - $T := T \cup \{(\bar{x}, y^k)\}$;
 - Seja (α^k, x^k, y^k) a solução ótima da relaxação atualizada $P_{Y^k}^{QA}(T)$
- Se $\alpha^k < UB$ (branch) então
 - Escolha uma variável y_j com uma componente fracionária y_j^k
 - Faça $Y^{i+1} := Y^k \cap \{y \in \mathbb{R}^n : y_j \leq \lfloor y_j^k \rfloor\}$;
 - Faça $Y^{i+2} := Y^k \cap \{y \in \mathbb{R}^n : y_j \geq \lceil y_j^k \rceil\}$;
 - $\mathcal{L} := \mathcal{L} \cup \{i+1, i+2\} - \{k\}$;
 - $i := i+2$;
 - Caso contrário, (eliminação do nó por limites) $\mathcal{L} := \mathcal{L} - \{k\}$;

Capítulo 5

Resultados Numéricos

Experimentos computacionais com os modelos exatos do problema de Steiner foram realizados usando o solver BONMIN (Basic Open-source Nonlinear Mixed INteger programming). O BONMIN é um código aberto, distribuído pelo COIN-OR [3], que, segundo os pesquisadores que o idealizaram, resolve problemas não-lineares inteiros mistos de forma exata quando o problema é convexo e é uma heurística para problemas não convexos. Os dados de entrada de cada instância foram escritos na linguagem AMPL [22]. Mais detalhes sobre o BONMIN podem ser encontrados em [23]. Através de um parâmetro, é possível escolher o algoritmo para a resolução do problema (MINLP). As opções disponíveis são:

- B-BB, um algoritmo baseado no Branch and Bound para problemas (MINLP). Resolve uma relaxação contínua não linear em cada nó da árvore de busca.
- B-OA, algoritmo de aproximação exterior baseado em algoritmos de decomposição.
- B-QG, implementação do algoritmo Branch and Cut de Quesada e Grossmann. É um algoritmo de aproximação exterior baseado no Branch and Bound.

- B-Hyb, algoritmo híbrido entre o Branch and Cut e o de Aproximação Exterior, baseado em Branch and Bound.

Detalhes destes algoritmos se encontram em [21]. Os autores consideram que o algoritmo híbrido B-Hyb foi o que apresentou melhor desempenho em grande número de testes de problemas MINLP convexos. Há casos em que o B-OA foi mais rápido do que os demais e há casos nos quais o B-BB pode ser mais adequado. O algoritmo B-QG é o algoritmo B-BB com alguns parâmetros desabilitados. Para MINLPs não convexos os autores recomendam o uso do B-BB, mesmo que para tais problemas ele não garanta o ótimo global, sendo possível melhorar a qualidade das soluções, com ajuste de alguns parâmetros.

Os experimentos computacionais foram realizados com dois modelos. O modelo 1 (3.1) - (3.6) é o modelo não convexo, de Maculan, Michelon e Xavier, [8]. O modelo 2 (3.7)- (3.15) é convexo, desenvolvido por Fampa e Maculan [9]. As instâncias foram retiradas do site OR-Library [1]. Essa fonte foi utilizada por apresentar instâncias resolvidas pelo algoritmo exato, sendo a única que disponibilizava o valor da função objetivo, o número de pontos de Steiner e suas coordenadas.

5.1 Tabelas

Na Tabela (5.1) apresentamos os dados referentes às 15 instâncias consideradas neste trabalho. A coluna indicada por I contém o número da instância do problema de Steiner. A coluna p indica o número de pontos dados e a coluna n indica a dimensão do problema. A coluna "solução ótima" fornece a solução ótima conhecida do Problema de Steiner, disponível no site OR-Library [1].

Nas Tabelas (5.2), (5.3) e (5.4), mostramos os resultados obtidos com a aplicação do pacote BONMIN aos modelos matemáticos do PSE para as 15 instâncias. As colunas indicam a o valor da função objetivo (z^*) obtido pelo BONMIN, o tempo (em segundos) para obter a solução, o limite inferior no nó raiz e o gap

$$\left(\frac{z^* - \text{valor \acute{o}timo conhecido}}{\text{valor \acute{o}timo conhecido}} * 100 \right).$$

Os algoritmos B-BB e B-Hyb obtiveram \acute{o}timos locais para os modelos 1 e 2 para todas as inst\^ancias com exce\c\~ao das inst\^ancias 8,9,10,11, que foram interrompidas por limite, conforme indicado na sa\xedda do BONMIN, que n\~ao explicita qual limite levou \`a interrup\c\~ao. As inst\^ancias 7,8,9,10,11 e 12 apresentaram valores de gaps altos, acreditamos que, \`a medida que o n\~umero de pontos fixos aumenta, a parte combinat\~oria do problema tem influ\^encia nestes resultados. Como foi utilizada a vers\~ao acad\^emica do AMPL, n\~ao foi poss\xedvel resolver a inst\^ancia 12 utilizando o modelo 2, por causa da limita\c\~ao no n\~umero de vari\~aveis e de restri\c\~oes. Isto foi indicado nas tabelas pelo s\xedmbolo '-'. O algoritmo B-OA n\~ao resolveu o modelo 1, a informa\c\~ao reportada no processamento \xe9 que a relaxa\c\~ao cont\xednua \xe9 ilimitada. Para o modelo 2 B-OA encontrou \acute{o}timos locais e novamente para as inst\^ancias 8,9,10 e 11 houve interrup\c\~ao por limite.

Na Tabela (5.5) apresentamos os melhores resultados obtidos com as 6 poss\xedveis combina\c\~oes de algoritmos e modelos e a combina\c\~ao que levou ao resultado. Observamos que os melhores resultados foram obtidos pelo algoritmo B-Hyb aplicado ao modelo 2. B-Hyb levou aos melhores resultados para 12 das 15 inst\^ancias consideradas. Em 10 inst\^ancias o gap obtido foi menor que 2%. Observamos tamb\^em que os tempos computacionais utilizados para o pacote resolver o modelo 1 foi em geral bem menor, quando comparado ao modelo 2. Este fato se deve ao modelo 1 ser n\~ao convexo, fazendo com que os algoritmos parem com a obten\c\~ao de \acute{o}timos locais. Apesar de o modelo 2 ser convexo, n\~ao obtivemos a solu\c\~ao \acute{o}tima exata para a maioria das inst\^ancias. Mesmos para aquelas nas quais o pacote BONMIN indica a obten\c\~ao da solu\c\~ao \acute{o}tima (1-7 e 12-15), os gaps n\~ao foram nulos, provavelmente devido a erros num\^ericos na solu\c\~ao dos problemas n\~ao lineares. Com rela\c\~ao ao limite inferior no n\~o raiz, o modelo 1 levou a resultados bem melhores. No entanto, pelo fato do modelo ser n\~ao convexo, n\~ao temos a garantia de que o valor encontrado seja de fato, um limite inferior para o problema. Os limites ruins fornecidos para o modelo 2, est\~ao associados a utiliza\c\~ao de um Big-M no modelo. Estes limites

ruins prejudicam o tempo de convergência dos algoritmos. Finalmente concluímos que apesar do modelo 1 ter obtido soluções piores, ele ainda obtém boas soluções num tempo computacional pequeno e pode ser uma boa opção para compor métodos híbridos.

I	p	n	solução ótima
1	3	2	0.1989684831
2	4	2	2.73205
3	4	2	0.5032861762
4	5	2	1.664399322
5	5	2	1.818179307
6	6	2	1.500499823
7	6	2	1.598751727
8	7	2	2.077671105
9	8	2	0.7660261405
10	8	2	2.138789039
11	9	2	1.273376125
12	10	2	1.642792159
13	4	4	3.44949
14	5	5	4.43355
15	6	6	5.40629

Tabela 5.1: Aspectos gerais das instâncias de OR-Library estudadas [1]

I	B-Hyb							
	Modelo 1				Modelo 2			
	z^*	t(s)	lim	gap(%)	z^*	t(s)	lim	gap(%)
1	0.2103	0.01	0.016	5.70	0.20097	0.04	0.2009	1.01
2	2.73607	0.04	2	0.15	2.73205	0.84	0.7070	0.00
3	0.576299	0.02	0.0971	14.51	0.512506	0.27	0.0154	1.83
4	1.82159	0.06	0.512	9.44	1.68232	21.71	$-1.7e^{-7}$	1.08
5	1.83758	0.06	0.6047	1.07	1.81818	31.14	$-1.5e^{-7}$	0.00
6	1.55605	0.12	0.337	3.70	1.5001	517.60	$-2.9e^{-7}$	0.03
7	1.79943	0.06	0.5192	12.55	1.62316	159.57	$-2.7e^{-7}$	1.53
8	2.35998	0.06	0.7538	13.59	2.26406	1521.16	$-4.4e^{-7}$	8.97
9	0.85623	0.20	0.0708	11.78	0.805423	2631.79	$-5.2e^{-7}$	5.14
10	2.39537	0.14	0.4666	12.00	2.79967	22.21	$-6.2e^{-7}$	30.9
11	1.56503	0.20	0.1888	22.90	1.44089	3600.00	$-8.3e^{-7}$	13.16
12	2.1985	0.44	0.3117	33.83	-	-	-	-
13	3.5	0.03	3	1.46	3.46399	0.84	1.2713	0.42
14	4.53758	0.06	4	2.35	4.43355	28.57	$-1.7e^{-7}$	0.00
15	5.57208	0.15	3.6	3.07	5.40629	2331.35	$-2.7e^{-7}$	0.00

Tabela 5.2: Resultados obtidos com algoritmo B-Hyb do BONMIN

I	B-BB							
	Modelo 1				Modelo 2			
	z^*	t(s)	lim	gap(%)	z^*	t(s)	lim	gap(%)
1	0.2103	0.01	0.016	5.70	0.2103	0.05	0.016	5.70
2	2.73607	0.04	2	0.15	2.73607	0.84	2	0.15
3	0.576299	0.02	0.0971	14.51	0.576299	0.27	0.0971	14.51
4	1.82159	0.06	0.512	9.44	1.82159	21.77	0.512	9.44
5	1.83758	0.06	0.6047	1.07	1.83758	31.08	0.6047	1.07
6	1.55605	0.14	0.337	3.70	1.55605	518.04	0.337	3.70
7	1.79943	0.05	0.5192	12.55	1.79943	160.07	0.5192	12.55
8	2.35998	0.08	0.7538	13.59	2.35998	1530.47	0.7538	13.59
9	0.85623	0.20	0.0708	11.78	0.85623	2627.43	0.0708	11.78
10	2.39537	0.14	0.4666	12.00	2.39537	22.25	0.4666	12.00
11	1.56503	0.21	0.1888	22.90	1.56503	3600.00	0.1888	22.90
12	2.1985	0.46	0.31177	33.83	-	-	-	-
13	3.5	0.03	3	1.46	3.5	0.83	3	1.46
14	4.53758	0.08	4	2.35	4.53758	28.76	4	2.35
15	5.57208	0.16	3.6	3.07	5.57208	2347.67	3.6	3.07

Tabela 5.3: Resultados obtidos com algoritmo B-BB do BONMIN

I	Modelo 2			
	B-OA			
	z^*	t(s)	lim	gap(%)
1	0.20097	0.04	0.016	1.01
2	2.828	0.11	2	3.51
3	0.512506	0.17	0.0971	1.83
4	2.17687	0.22	0.512	30.79
5	1.89105	0.22	0.6047	4.01
6	1.68937	3.82	0.337	12.59
7	1.99412	1.97	0.5192	24.73
8	2.42752	3.22	0.7538	16.84
9	1.14469	657.20	0.0708	49.43
10	2.65064	657.30	0.4666	23.93
11	1.44089	3600.00	0.1888	13.16
12	-	-	-	
13	3.46399	0.88	3	0.42
14	5.65597	2.04	4	27.57
15	5.42066	380.86	3.6	0.27

Tabela 5.4: Resultados obtidos com algoritmo B-OA do BONMIN

I	z^*	gap(%)	Algoritmo/Modelo
1	0.20097	1.01	B-Hyb/Mod2. B-OA/Mod2
2	2.73205	0.00	B-Hyb/Mod2
3	0.512506	1.83	B-Hyb/Mod2. B-OA/Mod2
4	1.68232	1.08	B-Hyb/Mod2
5	1.81818	0.00	B-Hyb/Mod2
6	1.5001	0.03	B-Hyb/Mod2
7	1.62316	1.53	B-Hyb/Mod2
8	2.26406	8.97	B-Hyb/Mod2
9	0.805423	5.14	B-Hyb/Mod2
10	2.39537	12.00	B-Hyb/Mod1. B-BB/Mod1. B-BB/Mod2
11	1.44089	13.16	B-Hyb/Mod2. B-OA/Mod2
12	2.1985	33.83	B-Hyb/Mod1. B-BB/Mod1
13	3.46399	0.42	B-Hyb/Mod2
14	4.43355	0.00	B-Hyb/Mod2
15	5.40629	0.00	B-Hyb/Mod2

Tabela 5.5: Melhores Resultados Obtidos

Capítulo 6

Heurística para o PSE: Algoritmo Microcanônico

Neste capítulo descrevemos o algoritmo de otimização microcanônica, uma heurística baseada em conceitos da Física Estatística, em simulações da evolução de um sistema físico, isolado termicamente e em equilíbrio em uma dada energia interna.

6.1 Algoritmo de Otimização Microcanônica

O algoritmo de otimização microcanônica (μO) proposta por Montenegro *et al.* em [24] para o (PSE) propicia, entre os métodos propostos na literatura, obter melhores soluções com melhor tempo, principalmente para as instâncias do PSE com maior dimensão. A partir de uma solução inicial arbitrária, dois procedimentos principais, chamados inicialização e sampling são aplicados alternadamente. A fase de inicialização realiza uma busca local no espaço de soluções a fim de alcançar uma configuração de mínimo local. Obtida essa configuração, na fase de sampling, busca-se escapar do mínimo local gerado na inicialização procurando outras configurações de custo equivalente, ou seja, fazendo mudanças que não alterem demasiadamente a energia E da solução por um determinado número de iterações. Esta fase gera amostras de configurações de energia fixa. O algoritmo, então, inicia a partir de uma nova solução corrente e fica alternando entre as fases de inicialização e sampling até

que algum critério de parada seja satisfeito. A seguir descreveremos mais detalhes de cada uma das fases desta heurística.

6.1.1 Inicialização

Nesta fase, o algoritmo parte de uma solução inicial arbitrária e propõe movimentos os quais são aceitos apenas quando levam a configurações de custo menor. Há uma opção de tornar a busca mais agressiva, fazendo que o algoritmo selecione o melhor movimento, dentre um conjunto de movimentos possíveis. No caso da busca não agressiva, como não se pode determinar rigorosamente quando o mínimo local foi atingido, utiliza-se como critério de parada desta fase um número máximo de movimentos rejeitados.

6.1.2 Sampling

Nesta fase o objetivo da metaheurística é sair do mínimo local alcançado na fase anterior mas ao mesmo tempo, tentando manter-se próximo do valor de custo da solução obtida até então. O algoritmo usa uma variável livre chamada de demônio, a qual gera pequenas perturbações da solução corrente. Em cada iteração, movimentos randômicos são propostos, os quais são aceitos apenas se essa variável demônio consegue absorver ou produzir a diferença do custo.

A fase de sampling gera uma sequência de configurações cuja energia é conservada, talvez, exceto por pequenas variações que são modeladas pela variável demônio. Esta é formada por dois parâmetros: sua capacidade denotada por $DMAX$ e seu valor inicial denotado por D_I . Chamando de E_S a energia (custo) associada à solução S obtida na fase de inicialização, de D a energia do demônio e E a da solução, na fase de sampling, devemos ter $E + D = E_S + D_I = \text{constante}$. Logo, a energia da solução é escrita $E = E_S + D_I - D$ e portanto teremos que, nesta fase, serão geradas soluções cuja energia ficará limitada ao intervalo $[E_S + D_I - DMAX, E_S + D_I]$. Desse modo, os parâmetros D_I e $DMAX$ se tornam os mais importantes a serem definidos na implementação. Em [25] foi proposto que esses parâmetros fossem escolhidos a

partir da lista de movimentos rejeitados (da fase de inicialização). Considerando os valores diferença de custo da solução ordenados de forma crescente, tem-se um panorama da região de busca da solução. Daí, escolhe-se os dois menores valores para a energia inicial D_I e a capacidade máxima $DMAX$, pois estes valores seriam os mais adequados para definir a perturbação da solução corrente.

Esta heurística será utilizada no processo de otimização do framework POPMUSIC que abordaremos a seguir.

Capítulo 7

Framework POPMUSIC

O framework "Partial Optimization Metaheuristic Under Special Intensification Conditions", conhecido por POPMUSIC, foi criada por Éric Taillard [26][27]. Ele e Stefan Voss a aplicaram com sucesso em problemas de otimização combinatória grandes, como problema de clustering, roteamento de veículos, balanceamento de partes mecânicas [28] e rotulação cartográfica de pontos [29]. A ideia básica do POPMUSIC consiste em otimizar localmente subpartes de uma solução inicial. Estas otimizações locais são repetidas até encontrar um ótimo local. A partir de uma solução inicial S , define-se o que é uma parte desta solução. Então a solução inicial é decomposta em um conjunto de partes s_1, s_2, \dots, s_p . Define-se uma medida, um grau de relação entre essas partes. Escolhe-se uma parte, chamada de parte semente, e um número $r < p$ de partes $s_{i_1}, s_{i_2}, \dots, s_{i_r}$ que são mais relacionadas, segundo esta medida, com a parte semente s_i , formando um subproblema R_i . Realiza-se um processo de otimização no subproblema, se houver melhora, mantém esta parte. Se não houver melhora, esta parte volta a ser considerada para formar novo subproblema. Forma-se outro subproblema e repete-se o processo, até que não haja mais subproblemas que possam ser otimizados. A seguir apresentamos o pseudocódigo do POPMUSIC.

7.1 Algoritmo POPMUSIC

- 1 Entrada: Solução S composta das partes s_1, \dots, s_p e número r de subpartes
- 2 Conjunto $O = \emptyset$
- 3 Enquanto $O \neq \{s_1, \dots, s_p\}$ repete
 - Escolha $s_i \notin O$
 - Crie um sub-problema R_i composta de r partes s_{i_1}, \dots, s_{i_r} mais relacionadas com s_i
 - Otimize R_i
 - Se R_i melhora a solução, atualize S e partes correspondentes e faça $O \leftarrow \emptyset$, caso contrário, faça $O \leftarrow O \cup \{s_i\}$

O conjunto O armazena as partes semente geradoras dos subproblemas que não produziram melhora da solução. Quando contiver todas as partes da solução completa, então todos os subproblemas foram analisados sem sucesso e o processo para. Esta técnica é uma busca local, a estrutura da vizinhança de uma solução conterá todas as soluções S' que diferem da solução inicial S apenas por um subproblema R_i . O parâmetro r , definido pelo usuário, determina o tamanho de subproblemas que serão resolvidos. No caso em que $r = p$, tem-se que o subproblema é o problema original. Para resolver cada subproblema pode ser utilizado um método heurístico ou um método baseado em branch and bound, sendo neste caso o parâmetro r ajustado de modo que seja possível resolver os subproblemas pelo método exato em um tempo razoável.

Os pontos principais a serem definidos na implementação do POPMUSIC são

- A definição de parte de uma solução;
- A medida de relação entre as partes;
- O procedimento de escolha de uma parte em O ;
- O método de otimização a ser aplicado aos subproblemas.

Para o problema de Steiner Euclidiano, dados os p pontos fixos e os pontos de Steiner da solução inicial obtida pela heurística microcanônica, fizemos um particionamento destes pontos a partir da mediana das coordenadas dos pontos. Em \mathbb{R}^2 , são gerados 4 subproblemas, pois a divisão segundo a mediana em x gera 2 subconjuntos e depois a divisão de cada um destes subconjuntos em relação à mediana em y . Definimos como parte da solução o conjunto de pontos restrito ao subconjunto resultante do particionamento. Então, temos 4 partes, $\{s_1, s_2, s_3, s_4\}$. O parâmetro r foi escolhido como $r = 1$, ou seja, formamos um subproblema constituído de 2 partes. O método de otimização usado em cada subproblema é a heurística microcanônica implementada por Montenegro. A medida de relação entre 2 partes é a menor distância entre quaisquer pares de nós de cada uma delas. A escolha de uma parte semente no conjunto O foi aleatória. Definida a parte semente, analisamos qual das outras 3 partes é mais relacionada com esta parte semente. Formamos o subproblema e otimizamos. Resolvendo cada subproblema, encontramos a localização dos pontos de Steiner e calculamos a árvore geradora mínima, usando o algoritmo de Prim, do grafo inteiro. Foram realizados alguns testes em conjuntos de instâncias com no mínimo 80 pontos fixos em \mathbb{R}^2 e \mathbb{R}^3 , retirados do site de J.Beasley, comparando a heurística microcanônica aplicada ao problema inteiro, o algoritmo de Smith

(dados do site) e o uso da heurística com o POPMUSIC. As instâncias de 1 a 10 estão em dimensão 2 e 11 até 13 estão em dimensão 3. Observamos que o algoritmo BONMIN não consegue resolver instâncias tão grandes (em número de pontos fixos), por isso, aplicamos heurística neste novo conjunto de instâncias. Na Tabela 7.1 a seguir apresentamos uma comparação entre os valores da função objetivo obtidos pelo algoritmo exato (geosteiner), a heurística microcanônica e o framework POPMUSIC em conjunto com a heurística microcanônica. A primeira coluna é o número do problema, a segunda coluna mostra o número de pontos fixos do problema, a terceira coluna mostra o valor da função objetivo obtido pelo algoritmo geosteiner, a quarta coluna apresenta o valor objetivo obtido com a heurística microcanônica, a quinta coluna, os valores obtidos com o framework. A sexta coluna apresenta o gap* calculado pela expressão $\left(\frac{z^* - \text{valor ótimo (geosteiner)}}{\text{valor ótimo (geosteiner)}} * 100\right)$, onde z^* é o valor da solução pela heurística microcanônica. A última coluna apresenta o gap** calculado pela expressão $\left(\frac{z^{**} - \text{valor ótimo (geosteiner)}}{\text{valor ótimo (geosteiner)}} * 100\right)$, onde z^{**} é o valor da solução pelo framework usado com heurística microcanônica.

Inst	Nº Pontos Fixos	Algoritmo Exato (Geosteiner)	z^* Microcanônica (μO)	$z^{**}\mu O$ + POPMUSIC	gap*(%)	gap**(%)
1	80	6.257418	6.334703	6.332297	1.22	1.20
2	80	5.695396	5.763172	5.759893	1.18	1.13
3	90	6.056187	6.158267	6.139939	1.66	1.38
4	90	6.221351	6.303387	6.279729	1.30	0.94
5	90	6.460569	6.555474	6.540758	1.45	1.24
6	100	6.594812	6.698983	6.685550	1.56	1.36
7	100	6.531347	6.683620	6.665092	2.28	2.05
8	100	6.576977	6.660998	6.653345	1.26	1.16
9	100	6.674687	6.743929	6.742551	1.03	1.02
10	100	6.466368	6.601380	6.582626	2.05	1.80

Tabela 7.1: Tabela Comparativa

Observamos que há uma pequena melhora em algumas instâncias se comparamos a resolução do problema pela heurística Microcanônica em dimensão 2. É um caminho a ser abordado em dimensões maiores, uma vez que a aplicação deste framework consegue solução mais próxima da solução ótima conhecida. Fizemos alguns testes colocando o número de partes como 16 em algumas destas instâncias. Nesse caso, verificamos que a solução é uma árvore aproximada e a razão de Steiner piora, se aproximando de $\rho = 1$. Não comparamos o tempo pois esta informação não é fornecida no site OR-Library.

Capítulo 8

Problema de Steiner em Poliedros Regulares

O artigo de Smith apresenta uma tabela com o valor da razão de Steiner ρ para conjuntos de pontos que são vértices de poliedros regulares em \mathbb{R}^3 . A seguir analisaremos o desempenho do algoritmo híbrido do Bonmin para calcular o tamanho da árvore mínima de Steiner, através do modelo 2, para conjuntos de vértices de sólidos platônicos, cuja maior distância entre os pontos fixos, representada no modelo pela constante M , seja igual a 1, para minimizar os reflexos dessa constante nos cálculos. Como as restrições (3.8) e (3.9) do modelo 2 são não diferenciáveis devido à norma euclidiana e o BONMIN exige que as funções sejam diferenciáveis, foi introduzida uma constante $\lambda = 10^{-6}$ na função que representa a norma. Assim, as restrições (3.8) e (3.9) são escritas como

$$d_{ij} \geq \|a^i - x^j + \lambda\| - M(1 - y_{ij}), \quad [i, j] \in E_1$$

$$d_{ij} \geq \|x^i - x^j + \lambda\| - M(1 - y_{ij}), \quad [i, j] \in E_2$$

8.1 Tetraedro: $p = 4$

O seguinte conjunto de pontos define um tetraedro com aresta de tamanho $d = 1$ cuja maior distância entre os pontos é 1:

$$\begin{aligned}a_1 &= (0, 0, 0) \\a_2 &= (1, 0, 0) \\a_3 &= (0.5, \frac{\sqrt{3}}{2}, 0) \\a_4 &= (0.5, \frac{\sqrt{3}}{6}, \frac{\sqrt{6}}{3}).\end{aligned}$$

Há 3 soluções ótimas simétricas para esta instância que podem ser verificadas geometricamente. Aplicando o BONMIN obtivemos para o valor da função objetivo 2.43917. Denotaremos os pontos de Steiner por pontos vermelhos e os pontos fixos por pontos brancos.

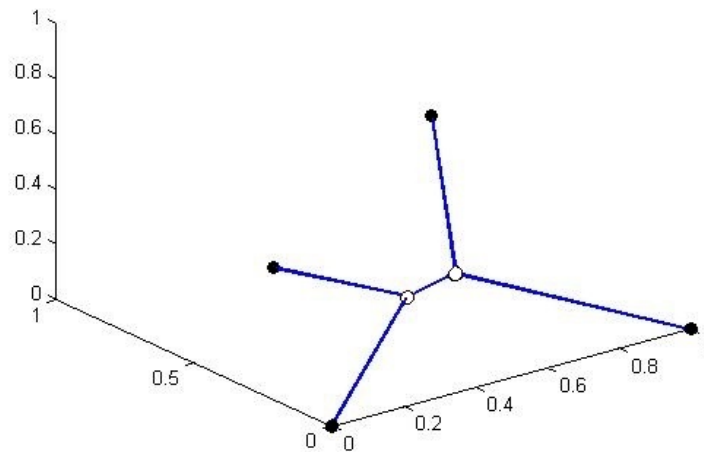


Figura 8.1: Árvore Mínima de Steiner Aproximada - Tetraedro

8.2 Octaedro: $p = 6$

O conjunto dos 6 pontos fixos descritos a seguir são as coordenadas dos vértices de um octaedro com aresta de tamanho $d = \frac{\sqrt{2}}{2}$. Estes vértices são o baricentro das

faces do cubo $[0, 1]^3$:

$$\begin{aligned}
 a_1 &= (0.5, 0.5, 0) \\
 a_2 &= (0.5, 0.5, 1) \\
 a_3 &= (0.5, 0, 0.5) \\
 a_4 &= (0.5, 1, 0.5) \\
 a_5 &= (0, 0.5, 0.5) \\
 a_6 &= (1, 0.5, 0.5).
 \end{aligned}$$

Utilizando o Bonmin, o valor objetivo obtido para a função, ou seja, o tamanho da árvore obtido foi 2.868032. Foram analisados 1624 nós e o tempo de CPU foi aproximadamente 88 segundos. A razão de Steiner do octaedro em \mathbb{R}^3 é 0.811197 [30]. A árvore geradora mínima para este octaedro é formada por 5 arestas que conectam os pontos, portanto o tamanho da AGM é $\frac{5\sqrt{2}}{2} \cong 3.53553390$. Logo, estima-se que o tamanho da árvore mínima de Steiner seja

$$\rho = \frac{|\text{AMS}|}{|\text{AGM}|} \Rightarrow 0.811197 = \frac{|\text{AMS}|}{3.53553390} \Rightarrow |\text{AMS}| = 2.86801449$$

Como foi mencionado na construção do modelo 1 no artigo [8], o conjunto S de pontos de Steiner induzem um subgrafo, completo, e as variáveis binárias y_{ij} que satisfazem as restrições (3.3),(3.4) e (3.5) estão associadas a um subconjunto de árvores geradoras do grafo G . As restrições do tipo (3.5) garantem que os pontos de Steiner se conectam entre si formando uma árvore, não haverá ciclos entre os pontos de Steiner. Além disso, este tipo de restrição estabelece que o primeiro ponto de Steiner estará sempre conectado ao segundo ponto de Steiner. Por exemplo, quando há 4 pontos de Steiner, nomeados por $\{7, 8, 9, 10\}$, as restrições do tipo (3.5) são

$$\begin{cases}
 y_{7,8} & = 1 \\
 y_{7,9} + y_{8,9} & = 1 \\
 y_{7,10} + y_{8,10} + y_{9,10} & = 1
 \end{cases}$$

Uma solução viável do conjunto de restrições é $y_{7,8} = 1$, $y_{7,9} = 1$ e $y_{7,10} = 1$, as

demais variáveis iguais a zero. Outra solução é $y_{7,8} = 1$, $y_{8,9} = 1$ e $y_{7,10} = 1$. Na Figura 8.2 mostra-se os dois grafos representando estas soluções. Estes grafos não são isomorfos.

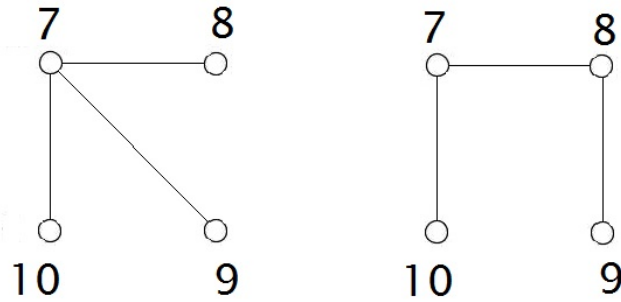


Figura 8.2: Soluções Não isomorfas

Verificamos que qualquer outra solução do conjunto de restrições irá gerar um grafo isomorfo a uma destas duas soluções mencionadas. Se considerarmos a solução $y_{7,8} = 1$, $y_{7,9} = 1$ e $y_{8,10} = 1$, o grafo, mostrado na Figura 8.3, será isomorfo ao grafo à direita da Figura 8.2.

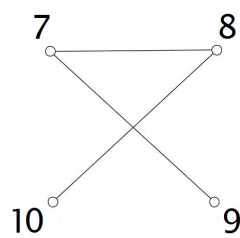


Figura 8.3: Solução

Consideramos que ao analisar todas as soluções possíveis, sem descartar subárvores isomorfas, o algoritmo pode levar considerável tempo de processamento, para chegar a uma solução que seria simétrica a outra que o algoritmo tivesse encontrado.

Foi feito um estudo de corte de soluções isomorfas a fim de diminuir o número de nós a serem analisados pelo algoritmo. A proposta inicial foi identificar estes isomorfismos entre soluções possíveis do conjunto de restrições (3.5).

Fixando uma configuração onde $y_{7,9} = 0$ e $y_{7,10} = 0$, observa-se uma redução do número de nós analisados de 1600 para 500 nós, o número de iterações do algoritmo reduziu de 66000 para aproximadamente 16000, enquanto o tempo de processamento reduziu de 88 segundos para 21 segundos. Foi encontrado o mesmo valor para a função objetivo. Observou-se também uma redução fazendo a mesma configuração para um hexágono.

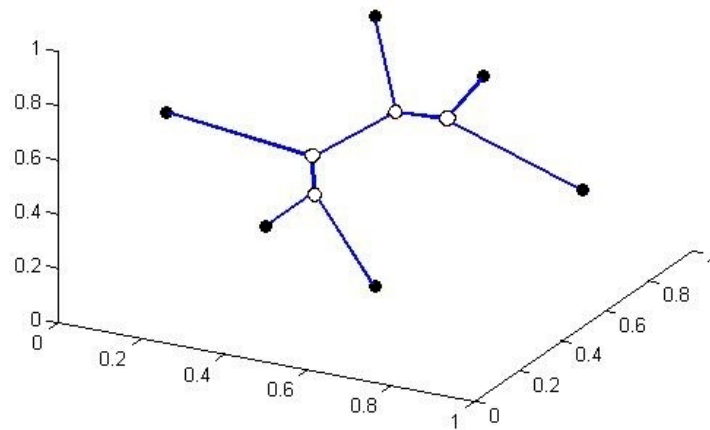


Figura 8.4: Árvore Mínima de Steiner Aproximada - Octaedro

8.3 Cubo: $p = 8$

Considere o seguinte conjunto de pontos fixos cujas coordenadas são os vértices de um cubo de aresta $d = \frac{\sqrt{3}}{3} \cong 0.57735$, cuja maior diagonal vale 1:

$$\begin{aligned}a_1 &= (0, 0, 0) \\a_2 &= (0, 0, d) \\a_3 &= (0, d, 0) \\a_4 &= (0, d, d) \\a_5 &= (d, 0, 0) \\a_6 &= (d, 0, d) \\a_7 &= (d, d, 0) \\a_8 &= (d, d, d)\end{aligned}$$

Utilizando o Bonmin, o valor da função objetivo, ou seja, o tamanho da árvore obtido foi 3.5773746. Foram analisados 1908 nós, sendo o processo interrompido por tempo máximo de busca. A razão de Steiner para a instância do cubo é 0.885165 [30]. É fácil ver, geometricamente, que a árvore geradora mínima é composta pelas 7 arestas que conectam os pontos. O tamanho da árvore geradora mínima para este cubo é $7\frac{\sqrt{3}}{3} \cong 4.04145189$, logo

$$\rho = \frac{|\text{AMS}|}{|\text{AGM}|} \Rightarrow 0.885165 = \frac{|\text{AMS}|}{4.04145189} \Rightarrow |\text{AMS}| = 3.57735175719.$$

O valor resultante é bem próximo. Procurou-se eliminar soluções simétricas nas subárvores que satisfazem a restrição (2). Como são 6 pontos de Steiner, o número de árvores geradoras desses 6 pontos, soluções do sistema das restrições do tipo (3.5) é $5! = 120$. Fixando uma configuração $y_{9,10} = 1, y_{9,11} = y_{9,12} = y_{9,13} = y_{9,14} = 0$, obteve-se a mesma resposta. Observa-se que a dificuldade de enumerar as soluções para encontrar as simetrias aumenta à medida que se aumenta a quantidade de pontos.

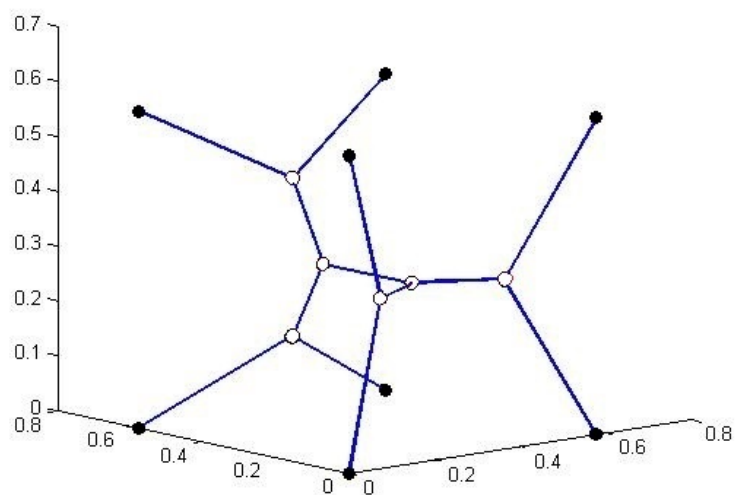


Figura 8.5: Árvore Mínima de Steiner Aproximada - Cubo

8.4 Icosaedro: $p = 12$

Smith encontrou como solução para um icosaedro regular, cuja aresta tem tamanho 2, o valor 18.5529. A razão de Steiner encontrada foi 0.843315. Testando o algoritmo para o icosaedro cuja aresta mede 0.53573 e os vértices estão descritos a seguir, obteve-se 4.949177 após 8 horas de processamento:

$$\begin{aligned}
a_1 &= (0, 0.26287, 0.42533) \\
a_2 &= (0, -0.26287, 0.42533) \\
a_3 &= (0, 0.26287, -0.42533) \\
a_4 &= (0, -0.26287, -0.42533) \\
a_5 &= (0.26287, 0.42533, 0) \\
a_6 &= (-0.26287, 0.42533, 0) \\
a_7 &= (0.26287, -0.42533, 0) \\
a_8 &= (-0.26287, -0.42533, 0) \\
a_9 &= (0.42533, 0, 0.26287) \\
a_{10} &= (0.42533, 0, -0.26287) \\
a_{11} &= (-0.42533, 0, 0.26287) \\
a_{12} &= (-0.42533, 0, -0.26287)
\end{aligned}$$

Em um segundo teste, que durou aproximadamente 24 horas, a melhor solução encontrada foi 4.9287804, que equivale a 18.40024 em relação ao icosaedro do artigo de Smith. Obtivemos, portanto, uma melhor solução do que a conhecida na literatura usando o algoritmo híbrido do BONMIN.

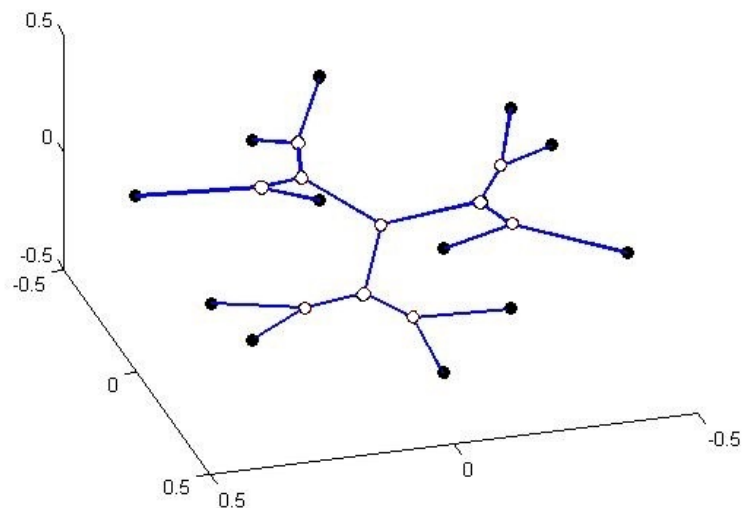


Figura 8.6: Árvore Mínima de Steiner Aproximada - Icosaedro

8.5 Dodecaedro: $p = 20$

O dodecaedro cujos 20 vértices estão descritos a seguir, tem 1 como maior distância entre seus pontos, a aresta tem comprimento 0.35682, enquanto que a árvore geradora mínima tem comprimento 6.7796:

$$\begin{aligned} a_1 &= (0.28868, & 0.28868, & 0.28868) \\ a_2 &= (0.28868, & 0.28868, & -0.28868) \\ a_3 &= (0.28868, & -0.28868, & 0.28868) \\ a_4 &= (0.28868, & -0.28868, & 0.28868) \\ a_5 &= (-0.28868, & 0.28868, & 0.28868) \\ a_6 &= (-0.28868, & 0.28868, & -0.28868) \\ a_7 &= (-0.28868, & -0.28868, & 0.28868) \\ a_8 &= (-0.28868, & -0.28868, & -0.28868) \\ a_9 &= (0.17841, & 0.46709, & 0) \\ a_{10} &= (-0.17841, & 0.46709, & 0) \\ a_{11} &= (0.17841, & -0.46709, & 0) \\ a_{12} &= (-0.17841, & -0.46709, & 0) \\ a_{13} &= (0.46709, & 0, & 0.17841) \\ a_{14} &= (-0.46709, & 0, & 0.17841) \\ a_{15} &= (0.46709, & 0, & -0.17841) \\ a_{16} &= (-0.46709, & 0, & -0.17841) \\ a_{17} &= (0, & 0.17841, & 0.46709) \\ a_{18} &= (0, & -0.17841, & 0.46709) \\ a_{19} &= (0, & 0.17841, & -0.46709) \\ a_{20} &= (0, & -0.17841, & -0.46709). \end{aligned}$$

Ao rodar o algoritmo híbrido do BONMIN para o dodecaedro, encontrou-se o valor 8.96839373 para a função objetivo, após interrupção por tempo, introduzindo-se o tamanho da árvore geradora mínima como limitante superior. O aspecto combinatorio do problema de Steiner torna o problema extremamente difícil de resolver

em dimensões maiores. A solução da Figura 8.7 foi obtida através do algoritmo microcanônico, o valor ótimo da função objetivo é 6.617614.

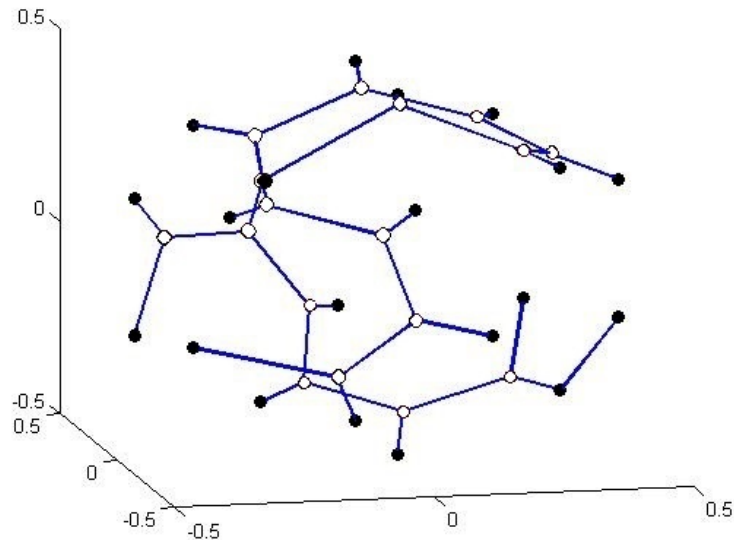


Figura 8.7: Árvore Mínima de Steiner Aproximada - Dodecaedro

8.6 Análise de Redução do Big-M do Modelo 2

Com o objetivo de investigar se há uma melhora de tempo de processamento ou do valor da função objetivo, rodamos para um mesmo sólido, cujas coordenadas estão descritas na seção anterior, o algoritmo BONMIN com o Big-M sendo a maior distância entre os pontos fixos e a seguir, com o Big-M sendo a $(n - 1)$ -ésima maior distância entre os pontos, conforme o Teorema da seção 3.1.2 no capítulo 3, que enunciaremos a seguir para facilitar a leitura.

Teorema: Seja o Problema Euclidiano de Steiner com n nós terminais. Suponha que S é uma árvore de Steiner ótima para o problema, com topologia cheia (degenerada ou não). Considere $d_1 \geq d_2 \geq d_3 \geq \dots \geq d_{\frac{n(n-1)}{2}}$ as distâncias Euclidianas colocadas em ordem não crescente, entre todos os pares de nós terminais. Então o tamanho de todas as arestas da árvore S tem comprimento menor ou igual a d_{n-1} .

Para o octaedro houve melhora no tempo de processamento, enquanto que, para o cubo não houve alterações. É possível que a distribuição dos pontos tenha influência

Sólido	M	Tempo(seg)	z^*
cubo	1	600	3.5773746
	$\frac{\sqrt{6}}{3}$	600	3.57737
octaedro	1	88	2.868032
	$\frac{\sqrt{2}}{2}$	52	2.868067

nos processos numéricos. No cubo, as distâncias em ordem não crescente são $d_1 = d_2 = d_3 = d_4 = 1$, $d_5 = \dots = d_{14} = \frac{\sqrt{6}}{3} \simeq 0.8164$, portanto $d_7 = 0.8164$. No octaedro, as distâncias em ordem não crescente são $d_1 = 1$, $d_2 = d_3 = d_4 = d_5 = \frac{\sqrt{2}}{2} \simeq 0.707106$.

8.7 Análise da Mudança de Escala das instâncias

Foi realizada uma comparação entre 4 cubos, reduzindo o comprimento da aresta com o objetivo de verificar se haveria redução no tempo de processamento em relação ao algoritmo híbrido do BONMIN. A tabela a seguir mostra o tamanho da aresta, o tempo (em minutos) de execução e o valor da função objetivo. O Big-M considerado foi, conforme o teorema acima, a $(n - 1)$ -ésima maior distância.

Aresta	M	Tempo(min)	z^*
1	$\sqrt{2}$	200	6.1961
$\frac{\sqrt{2}}{2}$	1	165	4.38138
0.5	$\frac{\sqrt{2}}{2}$	93	3.09810
0.25	$\frac{\sqrt{2}}{4}$	105	1.54909

Houve redução no tempo de processamento em relação aos cubos de aresta 1, de aresta $\frac{\sqrt{2}}{2}$ e de aresta medindo 0.5. Estes resultados indicam que escalonar o problema em \mathbb{R}^3 melhora o tempo de processamento. A topologia da solução do PSE não se altera quando escalamos o cubo.

Capítulo 9

Conclusões

Neste trabalho, apresentamos o Problema Euclidiano de Steiner (PSE) e os dois modelos matemáticos existentes na literatura. Foram apresentados os algoritmos para problemas de programação não linear inteiro misto desenvolvidos para resolver esta classe de problemas, a qual o PSE pertence. Foi feito um estudo comparativo entre estes algoritmos e o algoritmo exato conhecido para o PSE. Em seguida, com o objetivo de obter uma melhora da solução ou do tempo de processamento para instâncias em dimensões maior ou igual a três, apresentamos uma heurística conhecida por microcanônica e sugerimos a aplicação de um framework chamado POPMUSIC em conjunto.

Utilizando um dos modelos matemáticos, foi realizada a análise de instâncias específicas do \mathbb{R}^3 , no caso, os sólidos platônicos, para os quais a solução exata é conhecida. Aplicamos o pacote BONMIN para os sólidos platônicos e verificamos que, à medida que o número de pontos aumenta, se torna mais custoso encontrar a solução exata, como era esperado

Na busca por uma melhora do tempo de processamento, foi realizado um estudo sobre a simetria das soluções de um conjunto de restrições do PSE. A análise do isomorfismo destas soluções mostrou-se um caminho promissor para diminuir o número de iterações e o tempo.

A partir de um teorema sobre a limitação das arestas de uma árvore mínima de Steiner, foi feito um estudo comparativo, para ver a influência da mudança da

constante Big-M. Ainda foi feita uma análise sobre o escalonamento das dimensões dos sólidos. Obtivemos uma melhora para a instância do icosaedro.

Estudos que podem ser feitos a partir deste trabalho:

- Definição mais adequada dos parâmetros do framework POPMUSIC, por exemplo, em relação à solução inicial, ao particionamento do problema, à forma de reconstruir a solução.
- Busca por simetria nas soluções que satisfazem o subgrafo completo induzido pelos pontos de Steiner, a fim de eliminar soluções simétricas e melhorar o tempo de processamento do algoritmo.
- Estudo de desigualdades que possam ser acrescentadas ao modelo matemático.
- Estudo detalhado sobre a influência na redução da constante Big-M no desempenho do algoritmo.
- Estudo detalhado sobre a influência do escalonamento no desempenho do algoritmo em obter uma solução para o PSE.
- Elaboração de métodos híbridos mais eficientes para o PSE.

Referências Bibliográficas

- [1] BEASLEY, J. E. “OR-Library: Distributing test problems by eletronic mail”. <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>, June 1990.
- [2] GAREY, M. R., GRAHAM, R. L., JOHNSON, D. S. “The Complexity of Computing Steiner Minimal Trees”, *SIAM J. Appl. Math.*, v. 32, pp. 835–859, 1977.
- [3] BONAMI, P., CONN, A., LEE, J., et al. “COmputational INfrastructure for Operations Research - open source for the operations research community”. <http://www.coin-or.org/>, February 2012.
- [4] HWANG, F. K., RICHARDS, D. S., WINTER, P. “The Steiner Tree Problem”. In: *Annals of Discrete Mathematics*, v. 53, North-Holland, Amsterdam, 1992.
- [5] WINTER, P. “An algorithm for the Steiner problem in the Euclidean plane”, *Nteworks*, v. 15, pp. 323–345, 1985.
- [6] WINTER, P., ZACHARIASEN, M. “Euclidean Steiner Minimum Trees: An Improved Exact Algorithm”, *Networks*, v. 30, pp. 149–166, 1997.
- [7] WINTER, P., ZACHARIASEN, M. “Large Euclidean Steiner Minimum Trees in an Hour”. 1996.
- [8] MACULAN, N., MICHELON, P., XAVIER, A. E. “The Euclidean Steiner Tree Problem in \mathbb{R}^n : A mathematical programming formulation”, *Annals of Operations Research*, v. 96, pp. 209–220, 2000.
- [9] FAMPA, M., MACULAN, N. “Using a conic formulation for finding Steiner minimal trees”, *Numerical Algorithms*, v. 35, pp. 315–330, 2004.
- [10] RAVADA, S., SHERMAN, A. T. “Experimental Evaluation of a Partitioning Algorithm for the Steiner Tree Problem in \mathbb{R}^2 and \mathbb{R}^3 ”, *Networks*, v. 24, pp. 409–415, 1994.

- [11] KALPAKIS, K., SHERMAN, A. T. “Probabilistic Analysis of an Enhanced Partitioning Algorithm for the Steiner Tree Problem in \mathbb{R}^d ”, *Networks*, v. 24, pp. 147–159, 1994.
- [12] SMITH, J. M., WEISS, R., PATEL, M. “An $O(N^2)$ heuristic for Steiner minimal trees in E^3 ”, *Networks*, v. 25, pp. 273–289, 1995.
- [13] CHAPEAU-BLONDEAU, F., JANEZ, F., FERRIER, J. “A Dynamic Adaptive Relaxation Scheme Applied to the Euclidean Steiner Minimal Tree Problem”, *SIAM Journal of Optimization*, v. 7, pp. 1037–1053, 1997.
- [14] MONTENEGRO, F. M. T. *Heurísticas para o Problema de Steiner Euclidiano em \mathbb{R}^n* . Tese de Doutorado, COPPE - Universidade Federal do Rio de Janeiro, 2001.
- [15] FORTE, V. L. *Algoritmos de Otimização Aplicados ao Problema de Steiner em N Dimensões*. Tese de Mestrado, COPPE - Universidade Federal do Rio de Janeiro, 2010.
- [16] MONTENEGRO, F., MACULAN, N. “A Genetic Algorithm for the Euclidean Steiner Problem in \mathbb{R}^n ”. In: *Proc X. Cong. Ibero-Latinoamericano Inv. Oper. CLAIO*, 2000.
- [17] LEE, S., GROSSMANN, I. E. “New Algorithms for Nonlinear Generalized Disjunctive Programming”, *Computers and Chemical Engineering*, v. 24, pp. 2125–2141, 2000.
- [18] GUPTA, O. K., RAVINDRAN, V. “Branch and Bound Experiments in Convex Nonlinear Integer Programming”, *Management Science*, v. 31, pp. 1533–1546, 1985.
- [19] QUESADA, I., GROSSMANN, I. E. “An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems”, *Computers and Chemical Engineering*, v. 16, pp. 937–947, 1992.
- [20] LI, D., SUN, X. “Nonlinear Integer Programming”. v. 84, Springer-Verlag, 2006.
- [21] BONAMI, P., BIEGLER, L. T., CONN, A., et al. “An Algorithm Framework for Convex Mixed Integer Nonlinear Programs”, *Discrete Optimization*, v. 5, pp. 186 – 204, 2008.
- [22] FOURER, R., GAY, D., KERNIGHAN, B. *AMPL A modeling Language for Mathematical Programming*. Brooks/Cole Cengage Learning, 2003.
- [23] BONAMI, P., LEE, J., 2007. BONMIN Users’ Manual.

- [24] MONTENEGRO, F., MACULAN, N., TORREÃO, J. “Microcanonical Optimization Applied to the Euclidean Steiner Problem in \mathbb{R}^n ”. In: *MIC 2001 - 4th Metaheuristics International Conference*, v. 2, pp. 515–519, 2001.
- [25] 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*, v. 68, pp. 0567021–0567025, 2003.
- [26] TAILLARD, E. D. *Programmation a memoire adaptative et algorithmes pseudo gloutons: nouvelles perspectives pour les metaheuristiques*. Tese de Doctorado, University of Versailles, França, 1998. Habilitation thesis.
- [27] TAILLARD, E. D. *Heuristic methods for large centroid clustering problems*. Relatório técnico, IDSIA, Lugano, Suíça, 1996.
- [28] TAILLARD, E., VOSS, S. “POPMUSIC: Partial Optimization Metaheuristic Under Special Intensification Conditions”. pp. 613–629, C. Ribeiro, P. Hansen, *Essays and Surveys in Metaheuristics*, Kluwer Academic Publishers, 2001.
- [29] TAILLARD, E. D., ALVIM, A. C. F. “POPMUSIC for the Point Feature Label Placement”, *European Journal of Operational Research*, v. 192/2, pp. 396–413, 2009.
- [30] SMITH, W. D. “How to find Steiner Minimal Trees in Euclidean d-Space”, *Algorithmica*, v. 7, pp. 137–177, 1992.