



UM ENFOQUE PARA O PROBLEMA DE RECOBRIMENTO DE SÓLIDOS  
POR ESFERAS DE DIÂMETROS DIFERENTES

Renan Vicente Pinto

Dissertação de Mestrado 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 Mestre em Engenharia de Sistemas e Computação.

Orientador: Nelson Maculan Filho

Rio de Janeiro  
Março de 2011

UM ENFOQUE PARA O PROBLEMA DE RECOBRIMENTO DE SÓLIDOS  
POR ESFERAS DE DIÂMETROS DIFERENTES

Renan Vicente Pinto

DISSERTAÇÃO 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 MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

---

Prof. Nelson Maculan Filho, H.D.R.

---

Prof. Marcia Helena Costa Fampa, D.Sc.

---

Prof. Sérgio Barbosa Villas-Boas, Ph.D

RIO DE JANEIRO, RJ – BRASIL

MARÇO DE 2011

Vicente Pinto, Renan

Um Enfoque Para o Problema de Recobrimento de Sólidos por Esferas de Diâmetros Diferentes/Renan Vicente Pinto. – Rio de Janeiro: UFRJ/COPPE, 2011.

XI, 56 p.: il.; 29, 7cm.

Orientador: Nelson Maculan Filho

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2011.

Referências Bibliográficas: p. 51 – 56.

1. Recobrimento por esferas. 2. Programação matemática. I. Maculan Filho, Nelson. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

*Ao meu pai, Ricardo, à minha  
mãe, Deise, e aos meus afilhados  
Lucas e Luan, pois é por eles que  
eu vivo e é a eles que tudo devo.*

# Agradecimentos

Agradeço ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo suporte financeiro.

Agradeço a meu pai, Ricardo, e minha mãe, Deise, por estarem sempre do meu lado.

Agradeço aos meus sobrinhos e afilhados, Lucas e Luan, por tantas vezes me acordarem cedo para brincar e me permitirem trabalhar desde a manhã.

Agradeço à minha irmã, Diene, por treinar minha paciência durante esses dois longos anos.

Agradeço à Patricia Schroeder e Luciana Casacio pela força e incentivo contínuo.

Agradeço aos amigos de classe e laboratório, Jesus Ossian, Michael Souza, Vinicius Forte, Ana Flavia Uzeda, Viviane Köhler, Rogério Tostas e Vinicius Heineck, pela ajuda e apoio.

Agradeço, também, aos amigos de infância e familiares, Diego Mendes, Mônica Martins, Guilherme Martins, Heloíse Costa, Taísa Costa, Jéssica Carvalho, Amanda da Cunha, Marcelo Rodrigues, Marcos Antônio, Jorge Henrique, Nathália Napoleão, Renata Ribeiro, Rafael Ribeiro e Luiza Medeiros, que me proporcionaram ótimos momentos e nunca me deixaram esquecer das minhas responsabilidades.

Agradeço à Rosa, funcionária do LabOtim, e à Carolina, secretária do LAND, pelas boas companhias, pelas risadas e por toda a alegria passada nos arredores do Laboratório de Otimização (LabOtim).

Agradeço à Maria de Fátima Cruz Marques por toda a ajuda, em todos os âmbitos, sempre disponível e amável para me atender e me escutar.

Mas, agradeço, principalmente, ao meu orientador e grande amigo, professor Nelson Maculan, por ter me oferecido a oportunidade de ser orientado por uma pessoa tão especial e pelos valiosos ensinamentos durante esses dois anos do meu mestrado.

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

UM ENFOQUE PARA O PROBLEMA DE RECOBRIMENTO DE SÓLIDOS  
POR ESFERAS DE DIÂMETROS DIFERENTES

Renan Vicente Pinto

Março/2011

Orientador: Nelson Maculan Filho

Programa: Engenharia de Sistemas e Computação

Uma das modalidades mais efetivas de radiocirurgia estereotáxica é a radiocirurgia conhecida como *Gamma Knife*, direcionada a tratamentos de tumores, malformações vasculares e distúrbio de dor na cabeça. Feixes de 201 fontes radioativas de Cobalto-60 são irradiados e focados no tumor, afetando uma região aproximadamente esférica. Um tratamento típico consiste em cobrir o volume de tratamento com uma certa dose de radiação, minimizando o efeito nos tecidos saudáveis ao redor. Para muitos pacientes, o processo é bastante tedioso e demorado, além de depender altamente da experiência dos profissionais responsáveis. Neste trabalho, propomos uma formulação para o problema de programação matemática de cobertura de um sólido por um número finito de esferas de tamanho diferentes, visando a automatização do processo de planejamento do tratamento via *Gamma Knife*.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

AN APPROACH TO THE PROBLEM OF COVERING SOLIDS BY SPHERES  
OF DIFFERENT DIAMETERS

Renan Vicente Pinto

March/2011

Advisor: Nelson Maculan Filho

Department: Systems Engineering and Computer Science

One of the most effective modalities of stereotactic radiosurgery for treating brain tumors, vascular malformations and head pain disorders is the one known as Gamma Knife. Beams of 201 radioactive sources of Cobalt-60 are irradiated and focused on a point in the tumor, affecting a spherical region. A typical treatment consists of covering the treatment volume with a certain dose of radiation, minimizing the effect on healthy tissues around. For many patients, the process is tedious, time consuming, and highly dependent on experienced professionals. In order to automate the Gamma Knife treatment planning process, we propose a mathematical programming formulation for the problem of covering a solid by a finite number of spheres of different size.

# Sumário

<b>Lista de Figuras</b>	<b>x</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 O Problema de Recobrimento</b>	<b>4</b>
2.1 Definição . . . . .	4
2.2 Problemas Relacionados . . . . .	6
2.2.1 Problema de Recobrimento 2D . . . . .	6
2.2.2 Problema de Empacotamento . . . . .	8
2.2.3 O problema do Kissing Number . . . . .	12
<b>3 Otimização Global Mista</b>	<b>14</b>
3.1 Definição do Problema . . . . .	14
3.2 Metaheurísticas de Otimização Global . . . . .	16
3.2.1 Algoritmos Genéticos . . . . .	16
3.2.2 Simulated Annealing . . . . .	19
3.2.3 Busca Tabu . . . . .	21
3.2.4 Variable Neighborhood Search . . . . .	23
3.2.5 GRASP . . . . .	24
3.3 Spatial Branch-and-Bound . . . . .	27
<b>4 Modelo de Programação Não-Linear Mista</b>	<b>30</b>
4.1 Modelo Proposto . . . . .	31
4.2 Discretização . . . . .	35
4.3 Diminuindo a combinatória . . . . .	37
<b>5 Resultados e Discussões</b>	<b>38</b>
5.1 Heurística . . . . .	45
<b>6 Conclusões</b>	<b>49</b>
6.1 Trabalhos Futuros . . . . .	50





# Lista de Figuras

2.1	Recobrimento dos mapas do (a) Brasil, com 5 círculos; e (b) do estado do Rio de Janeiro, com 9 círculos. . . . .	7
2.2	(a) Arranjo de círculos em formato quadricular e (b) arranjo de círculos em formato hexagonal. . . . .	9
2.3	As configurações <i>hexagonal close packing</i> , à esquerda, e a <i>face-centered cubic close packing</i> , à direita. As letras indicam quais camadas são as mesmas. Há duas camadas ‘A’ na configuração hexagonal, onde todas as esferas estão na mesma posição. Todas as camadas na configuração cúbica são diferentes. Esta última pode ser convertida na primeira se transladarmos a camada ‘C’ de forma que a esfera desenhada ocupe a região pontilhada. . . . .	9
2.4	Empacotamento de círculos em regiões compactas do $\mathbb{R}^2$ . Nesses exemplos, todos os círculos possuem diâmetro 1,3 unidade de comprimento. (a) 13 círculos foram empacotados em um quadrado de lado 5 unidades de comprimento; (b) 10 círculos em um círculo maior de diâmetro 5. . . . .	10
2.5	(a) Solução do KNP em uma dimensão e (b) Solução do KNP em duas dimensões. A parte cinza indica a “esfera” central. . . . .	12
2.6	Solução do KNP em três dimensões. Essa configuração pode ser obtida arrumando as esferas de forma que seus pontos de tangência com a esfera central correspondam aos vértices de um icosaedro. . . . .	13
3.1	Pseudo-código do Algoritmo Genético. . . . .	17
3.2	Exemplo de cruzamento com apenas um ponto de corte, à esquerda, e com dois pontos de corte, à direita. . . . .	18
3.3	Pseudo-código do algoritmo <i>Simulated Annealing</i> . . . . .	20
3.4	Pseudo-código do algoritmo <i>Tabu Search</i> . . . . .	22
3.5	Pseudo-código do algoritmo <i>Variable Neighborhood Search</i> (VNS). . . . .	24
3.6	Pseudo-algoritmo da fase de construção do GRASP. . . . .	25
3.7	Pseudo-algoritmo da fase de busca local do GRASP. . . . .	26
3.8	Pseudo-algoritmo da metaheurística GRASP. . . . .	26

3.9	Pseudo-algoritmo do método <i>Spatial Branch-and-Bound</i> . . . . .	28
4.1	Representação dos parâmetros $\alpha$ , representando a máxima interpene- tração das esferas. . . . .	32
4.2	Representação da região de segurança, onde a região hachurada re- presenta o tumor ou, de forma geral, o volume a ser recoberto. . . . .	34
4.3	A região em cinza representa o volume a ser recoberto. Como a esfera não pode ultrapassar a região de segurança, o domínio da variável que representa o seu centro seria um conjunto parecido com a região em laranja. . . . .	35
5.1	Exemplo bidimensional mostrando as regiões que correspondem às grandezas utilizadas na descrição de soluções. Em (a), a região a ser recoberta e as bolas presentes no recobrimento. As regiões hachuradas representam os parâmetros (b) cov, (c) miscov e (d) overlap. . . . .	39
5.2	Solução encontrada pelo solver GUROBI para o recobrimento de um paralelepípedo de dimensões 14mm x 12mm x 10mm. . . . .	41
5.3	Solução encontrada pelo solver CPLEX para o recobrimento de um paralelepípedo de dimensões 14mm x 12mm x 10mm. . . . .	42
5.4	Solução encontrada pelo solver XPRESS para o recobrimento de um paralelepípedo de dimensões 14mm x 12mm x 10mm, sem as restrições de ordenação das esferas. . . . .	42
5.5	Solução encontrada pelo solver XPRESS para o recobrimento de um paralelepípedo de dimensões 14mm x 12mm x 10mm, com as res- trições de ordenação das esferas. . . . .	43
5.6	Solução encontrada pelo solver XPRESS para o recobrimento de um paralelepípedo de dimensões 14mm x 12mm x 10mm, com $\epsilon$ reduzido para 1,5mm. . . . .	43
5.7	Solução encontrada pelo solver XPRESS para o recobrimento de um paralelepípedo de dimensões 14mm x 12mm x 10mm, tendo sido dis- ponibilizadas apenas esferas de raio 2mm. . . . .	44
5.8	Solução encontrada pelo solver XPRESS para o recobrimento de um paralelepípedo de dimensões 14mm x 12mm x 10mm, com as res- trições de ordenação das esferas. . . . .	44
5.9	Solução encontrada pela heurística que desenvolvemos para o recobri- mento de um paralelepípedo de dimensões 14x12x10mm. . . . .	46
5.10	Resultado obtido com a heurística para um elipsóide de raios 10mm, 5.2mm e 5.2mm. . . . .	47
5.11	Resultado obtido com heurística para um elipsóide de raios 10mm, 5.2mm e 5.2mm, com o acréscimo de uma esfera. . . . .	47

# Capítulo 1

## Introdução

Problemas relacionados ao recobrimento de regiões de espaços euclidianos por figuras regulares têm sido objeto de estudo por longos anos. Em especial, são de bastante interesse em aplicações os problemas de recobrimento de regiões do  $\mathbb{R}^2$  por círculos, ou recobrimento de regiões do  $\mathbb{R}^3$  por esferas. Mas mesmo regiões em espaços de dimensões maiores têm sido assunto de pesquisa.

Analisaremos, neste trabalho, o problema de recobrimento de uma região tridimensional por esferas de diferentes tamanhos. Dados um conjunto de esferas e um polítopo, o objetivo é cobrir completamente a região utilizando o menor número possível de esferas.

Esse problema tem importantes aplicações em planejamentos de tratamentos de radiocirurgias estereotáxicas. Uma das modalidades mais efetivas de radiocirurgia estereotáxica é a radiocirurgia conhecida como *Gamma Knife* [1]. O *Gamma Knife* consiste em uma unidade de tratamento altamente especializada direcionada a tratamentos de tumores cerebrais, malformações vasculares e distúrbio de dor na cabeça. No interior da unidade de tratamento, feixes de 201 fontes radioativas de Cobalto-60 são irradiados e focados em um único ponto no espaço. Esse procedimento é conhecido como um *shot* de radiação. Como consequência, uma região aproximadamente esférica é afetada com uma alta dose de radiação. Um tratamento típico consiste de aplicações de *shots*, eventualmente de diferentes tamanhos, posicionados em diferentes pontos no tumor, cujo objetivo é cobrir o volume de tratamento com uma certa dose de radiação, minimizando o efeito nos tecidos saudáveis ao redor.

O início da radiocirurgia *Gamma Knife* se dá, após aplicação de anestesia local, com a fixação de um equipamento estereotáxico na cabeça do paciente, com o uso de parafusos. Esse equipamento estabelece um quadro de coordenadas através do qual o tumor será precisamente localizado. Imagens de ressonância magnética ou tomografias computadorizadas são usadas para determinar a posição do volume de tratamento em relação às coordenadas do equipamento. O próximo passo, então, é definir o plano de tratamento. Um neurocirurgião, um oncologista e um físico

trabalham juntos para desenvolver um bom plano de tratamento para o paciente.

Em alguns casos, o processo de planejamento do tratamento é relativamente fácil. Por exemplo, algumas lesões pequenas podem ser cobertas com apenas um *shot* de radiação e o planejamento estará completo em minutos. Infelizmente, esse processo pode se tornar muito mais complexo quando o volume do tumor é grande ou de formato irregular. Estes casos requerem o uso de mais *shots* de radiação.

No equipamento afixado à cabeça do paciente, podem-se anexar 4 capacetes que se diferenciam no tamanho. Esses capacetes são responsáveis pelos *shots*. Portanto, quatro tamanhos diferentes de *shots* estão disponíveis, afetando regiões aproximadamente esféricas com raios de 2mm, 4mm, 7mm e 9mm. Combinando múltiplos *shots* de radiação, um plano de tratamento pode ser realizado, de forma a tratar lesões de diferentes tamanhos e formas. Contudo, esse procedimento geralmente resulta em um recobrimento não-homogêneo, devido à sobreposição dos diferentes *shots*. Na prática, muitos casos são tratados com o uso de 1 a 15 *shots* de radiação [2, 3], embora um número maior que 10 seja considerado impraticável pelas rotinas clínicas de tratamento.

Toda vez que há uma mudança do tamanho do *shot*, deve-se retirar o capacete em uso e substituí-lo com o capacete de tamanho correto. Esse é um processo que depende bastante tempo, pois, devido ao alto peso dos capacetes, devem ser manipulados com o auxílio de máquinas. Além disso, após cada *shot*, deve-se ajustar a conexão entre o capacete e o equipamento afixado na cabeça do paciente, para que não haja erro de precisão na localização do *shot*. Para muitos pacientes, esse processo torna-se bastante tedioso e demorado. Além disso, a qualidade do plano de tratamento depende altamente da experiência e paciência dos profissionais responsáveis.

Por essas razões, há o desejo de automatizar o processo de plano de tratamento via *Gamma Knife*, objetivando encontrar um bom recobrimento do volume do tumor e acelerar o processo de planejamento do tratamento. Otimizar a quantidade de *shots*, a posição dos seus centros e cada um de seus tamanhos pode reduzir as não-homogeneidades e evitar que tecidos saudáveis sejam afetados pela irradiação, além de alcançar o recobrimento desejado.

A literatura existente sobre o uso de técnicas de otimização em radiocirurgias *Gamma Knife* abrange uma área maior do que apenas o problema de recobrimento de um volume por esferas, assunto desse trabalho. Estudos relacionados visam prover uma solução completa para o problema de planejamento, isto é, tentam encontrar não só as posições dos *shots* como também as dosagens, relacionadas ao tempo de irradiação. Em [2] há uma relação de abordagens existentes. Alguns modelos, em particular [4–6], utilizam as variáveis de decisão  $D_w(x_s, y_s, z_s, i, j, k)$ , representando a dose de irradiação administrada no ponto  $(i, j, k)$  do volume que representa o

tumor, afetado por um *shot* de raio  $w$  centrado em  $(x_s, y_s, z_s)$ . Essas formulações oferecem como resultado uma distribuição de dosagens dada pelos valores de  $D_w$ . O verdadeiro recobrimento por esferas é recuperado com o uso de simulação e de técnicas de estimação de parâmetros.

Na seção 2, apresentaremos alguns problemas bastante relacionados. São os problemas de empacotamento, utilizados por muitos autores nas formulações do problema de planejamento do tratamento, e o problema do *Kissing Number*. Será apresentado, também, o problema de recobrimento na sua versão bidimensional.

Os problema não-lineares inteiros mistos (MINLP), como se classificam os problemas de recobrimento por esferas, serão apresentados na seção 3, assim como algumas metaheurísticas de otimização global usadas na resolução dessa classe de problemas.

Na seção 4, propomos um novo modelo de programação não-linear inteira mista para o problema de recobrimento. Lembramos que o problema estudado nesse trabalho é o de estabelecer um bom recobrimento por esferas, não levando em consideração o problema das dosagens. Portanto, a meta não é obter uma solução completa para o problema de planejamento, ainda. Continuando a seção 4, apresentamos uma discretização que pode ser aplicada no modelo proposto e uma forma de reduzir a combinatória do problema.

Alguns resultados obtidos com as formulações apresentadas serão apresentados na seção 5. Discutiremos algumas das características inerentes às soluções do problema de recobrimento, como a dificuldade em comparar soluções distintas. Ainda nessa seção, descreveremos a heurística que desenvolvemos como uma ferramenta adicional de solução do problema.

Finalmente, na seção 6, serão apresentadas algumas conclusões com respeito a este trabalho e o que pode ser feito nessa área em trabalhos futuros.

# Capítulo 2

## O Problema de Recobrimento

### 2.1 Definição

O problema de recobrimento de interesse no tratamento por *Gamma Knife* pode ser descrito matematicamente como:

(P) Dados um conjunto compacto  $T \subset \mathbb{R}^3$ , um conjunto finito  $R \subset \mathbb{R}_+$  de raios, um conjunto  $N$  indexando as esferas e uma função  $\rho : N \rightarrow R$ , encontrar um conjunto de esferas  $\{B(x(i), \rho(i)) \mid i \in N\}$  de cardinalidade mínima recobrindo cada ponto de  $T$ .

O problema (P), como definido acima, pode ser formulado como um problema de programação matemática, com o objetivo de encontrar os centros de cada esfera e associar a cada uma delas um respectivo raio, através da função  $\rho$ , de forma que a união dessas esferas forme uma cobertura do conjunto  $T$ . Um modelo de programação infinita não-linear inteira mista foi desenvolvido por Liberti *et al.* [7], com um número infinito não-enumerável de variáveis e restrições, como segue.

Seja  $M \geq \text{diam}(T)$ . Considere as seguintes variáveis:

- $x : N \rightarrow \mathbb{R}^3$ , que representa o centro da esfera;
- $y : N \rightarrow \{0, 1\}$ , que representa a presença ou não da esfera correspondente na solução;
- $u : N \times T \rightarrow \{0, 1\}$ , que assume valor 1 caso a dada esfera cubra o ponto especificado; e
- $w : N \times U \rightarrow \{0, 1\}$ , que assume valor 1 caso a dada esfera tenha o raio especificado.

O modelo não possui função objetivo e o problema é corretamente descrito pelas restrições (2.1)-(2.5):

$$\|x^i - p\|^2 \leq u_i(p) \sum_{j \in U} w_{ij} r_j^2 + (1 - u_i(p)) M^2, \quad \forall i \in N, \forall p \in T \quad (2.1)$$

$$\sum_{j \in U} w_{ij} = 1, \quad \forall i \in N \quad (2.2)$$

$$\sum_{i \in N} u_i(p) \geq 1, \quad \forall p \in T \quad (2.3)$$

$$\int_{p \in T} u_i(p) dp \geq \epsilon y_i, \quad \forall i \in N \quad (2.4)$$

$$\int_{p \in T} u_i(p) dp \leq \text{Vol}(T) y_i, \quad \forall i \in N \quad (2.5)$$

Para atacar o problema, aproximou-se, por discretização, a região  $T$ , obtendo um MINLP convexo. Foram utilizados os seguintes parâmetros:

- $V$ , um conjunto de índices dos pontos obtidos da discretização de  $T$ ; e
- $\{p_v \mid v \in V\}$ , os pontos da discretização.

A variável  $u$  foi modificada para atender à discretização:

- $u : N \times V \rightarrow \{0, 1\}$ , que assume valor 1 caso a esfera dada cubra o ponto especificado da discretização.

A formulação escreve-se como em (2.6)-(2.10):

$$\min \quad \sum_{i \in N} y_i \quad (2.6)$$

$$\text{s.a} \quad \|x_i - p_v\|^2 \leq u_{iv} \rho_i^2 + (1 - u_{iv}) M^2, \quad \forall i \in N, \forall v \in V \quad (2.7)$$

$$\sum_{i \in N} u_{iv} \geq 1, \quad \forall v \in V \quad (2.8)$$

$$\sum_{v \in V} u_{iv} \geq y_i, \quad \forall i \in N \quad (2.9)$$

$$\sum_{v \in V} u_{iv} \leq |V| y_i, \quad \forall i \in N \quad (2.10)$$

As restrições (2.7) dizem que, se a esfera  $i$  cobre o ponto  $p_v$ , então a distância euclidiana entre  $p_v$  e o centro  $x_i$  dessa esfera deve ser, no máximo, o raio  $\rho_i$  dessa esfera. As restrições (2.8) asseguram que cada ponto da discretização de  $T$  será coberto por, pelo menos, uma esfera. Se uma dada esfera foi selecionada para a solução, as restrições (2.9) asseguram que ela cobre um volume não nulo de  $T$ . Caso contrário, as restrições (2.10) forçam as variáveis  $u$  a assumirem valor zero.

Os autores, em [7], ainda utilizaram técnicas de linearização na restrição (2.7), obtendo duas diferentes formulações lineares inteiras mistas. Contudo, esse tipo de abordagem pode tornar a resolução do problema impraticável, devido à grande quantidade de variáveis do tipo 0-1 presentes na formulação em discretizações mais finas.



Outra abordagem que pode ser encontrada na literatura é a formulação como um problema de programação geométrica signomial. Utilizando essa técnica, Quirino *et al.* [8] apresentam uma metodologia de solução do problema de recobrimento de um elipsóide que não se baseia em técnicas de otimização discreta, tratando até mesmo os raios das esferas de forma contínua.

## 2.2 Problemas Relacionados

Nesta seção, abordaremos problemas bastante relacionados com o problema de recobrimento, assunto desse trabalho. Primeiro, falaremos sobre o problema de recobrimento em duas dimensões, em uma versão mais simples, com círculos de mesmo raio. O estudo de problemas de recobrimento em casos bidimensionais tem mostrado bons resultados e acredita-se que há a possibilidade de extensão de algumas idéias de algoritmos para o caso tridimensional.

Apresentaremos, também, os problemas de empacotamento, bastante presentes na literatura relacionada ao planejamento do tratamento por *Gamma Knife*. Problemas de empacotamento são estudados por diversos autores como uma outra forma de modelar o tratamento. Por fim, é apresentado o problema conhecido como *Kissing Number*.

### 2.2.1 Problema de Recobrimento 2D

Consideramos, nesta seção, um caso especial de recobrimento de domínios planos finitos por um dado número fixo de círculos iguais. Almeja-se, então, encontrar o menor raio e a posição desses círculos. Esse problema surge em um grande número de aplicações práticas, como, por exemplo, em localização e dimensionamento de centros de telecomunicação. Tal aplicação foi a motivação do estudo realizado por Xavier *et al.* [9]. Embora o problema de recobrimento do tratamento por *Gamma Knife* tenha como variável a quantidade de esferas, ao invés do raio, ambos os problemas estão bastante relacionados.

O problema bidimensional foi formulado, em [9], como um problema min-max-min, como segue. Seja  $q$  a quantidade de círculos a cobrir a região 2D. Seja  $X = \{x_i\}_{i=1,\dots,q}$  um conjunto dos centros dos círculos que devem cobrir a região  $S \subseteq \mathbb{R}^2$ . Dado um ponto em  $s \in S$ , calcula-se a distância de  $s$  ao centro do círculo mais próximo, dada por

$$d(s, X) = \min_{x_i \in X} \|s - x_i\|_2 .$$

A distância  $d(s, X)$  é uma medida de cobertura para um ponto  $s$  específico. De forma a qualificar o cobertura de toda a região  $S$  pelos  $q$  círculos, os autores de [9] tomam o maior valor dentre as distâncias  $d(s, X)$ , o que corresponde ao cobertura mais crítico de um ponto. Esse valor é denotado como  $D(X)$ , e é escrito como

$$D(X) = \max_{s \in S} d(s, X) .$$

Essas distâncias são calculadas a partir de uma configuração prévia dos círculos. Ou seja, para o cálculo das distâncias, os círculos já devem ter as posições dos seus centros definidas. O posicionamento ótimo dos centros deve acarretar em um cobertura de  $S$  de melhor qualidade. Para esse feito, minimiza-se o cobertura mais crítico. Denotando por  $X^*$  o posicionamento ótimo, temos:

$$X^* = \arg \min_{X \in \mathbb{R}^{2q}} D(X) .$$

Escrevendo de outra forma, temos:

$$X^* = \arg \min_{X \in \mathbb{R}^{2q}} \max_{s \in S} \min_{x_i \in X} \|s - x_i\|_2 .$$

Para resolver o problema, o conjunto  $S$  foi discretizado. Após algumas reformulações, com prova de que a formulação final possuía mesmo valor ótimo que o problema agora descrito, foi aplicada a técnica de suavização hiperbólica [10], para lidar com a não-diferenciabilidade da função empregada. Bons resultados numéricos foram obtidos. Alguns exemplos podem ser vistos na figura 2.1, retiradas de [9].

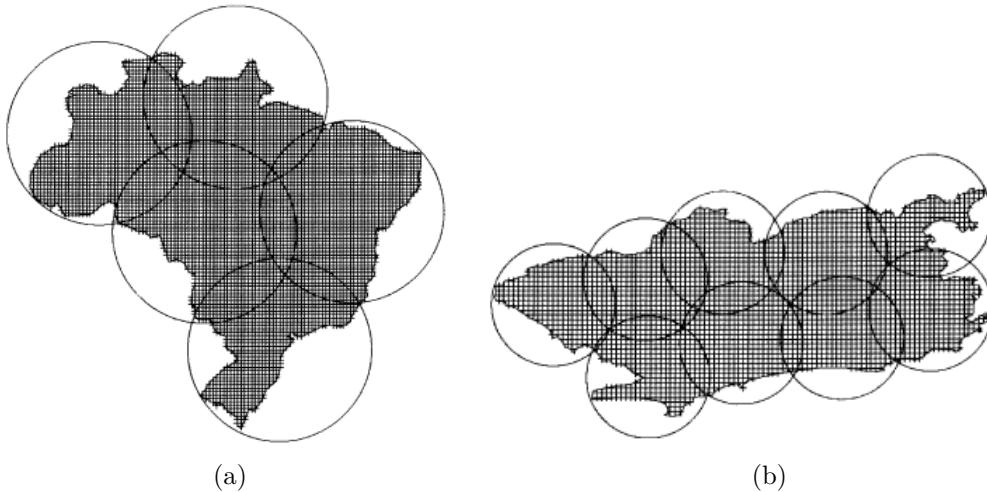


Figura 2.1: Recobrimento dos mapas do (a) Brasil, com 5 círculos; e (b) do estado do Rio de Janeiro, com 9 círculos.

## 2.2.2 Problema de Empacotamento

Os problemas de empacotamento, ou *Packing Problems*, são uma classe de problemas de otimização combinatória que envolvem empacotar certos objetos dentro de algum contêiner, da forma mais densa possível, sem sobreposição dos objetos.

Um problema de empacotamento bastante conhecido é o **problema da mochila**, também conhecido como *Knapsack Problem*. Dado um certo conjunto de objetos, cada um com seu peso e seu valor, o problema da mochila determina a quantidade de cada um dos objetos a ser incluída no contêiner, de forma que o peso total seja menor do que a capacidade máxima permitida da mochila e o valor total seja o maior possível. Matematicamente, formulamos o problema da mochila, em sua forma mais simples, da seguinte maneira:

$$\begin{aligned} &\text{maximizar} && \sum_{i=1}^n c_i x_i \\ &\text{sujeito a} && \sum_{i=1}^n a_i x_i \leq b \\ &&& x_i \geq 0 \text{ e inteiro, } i = 1, \dots, n. \end{aligned}$$

Acima,  $c$  é o vetor de custos, representando o valor de cada objeto,  $a$  é o vetor de pesos dos objetos e  $b$  é a capacidade do contêiner. O problema da mochila é NP-completo, embora possa ser resolvido em tempo pseudo-polinomial através de programação dinâmica [11].

A seguir, apresentaremos alguns dos problemas geométricos de empacotamento mais conhecidos, os empacotamentos de esferas.

### Empacotamento de Esferas

O problema geral de empacotamento de esferas determina qual a forma mais densa de empacotar esferas  $n$ -dimensionais em uma região do  $\mathbb{R}^n$ , sem sobreposição. As únicas interseções permitidas são os pontos de tangência. Por densidade, entende-se a proporção de espaço da região preenchido pelas esferas.

Em regiões não-limitadas, soluções desses tipos de problemas envolvem posicionar esferas em arranjos reticulados, ou **arranjos regulares**, onde os centros das esferas formam um padrão simétrico. Há ainda os arranjos que não formam um reticulado, chamados de irregulares, que podem ser periódicos, não-periódicos ou aleatórios. Os arranjos regulares são periódicos e os seus altos graus de simetria os tornam mais fáceis tanto de classificá-los quanto de medir suas densidades.

Em duas dimensões, há empacotamentos especiais de círculos [12], baseados em polígono regulares. Dois exemplos podem ser vistos na Figura 2.2.

O empacotamento de esferas com o arranjo hexagonal é sabido ser o mais denso

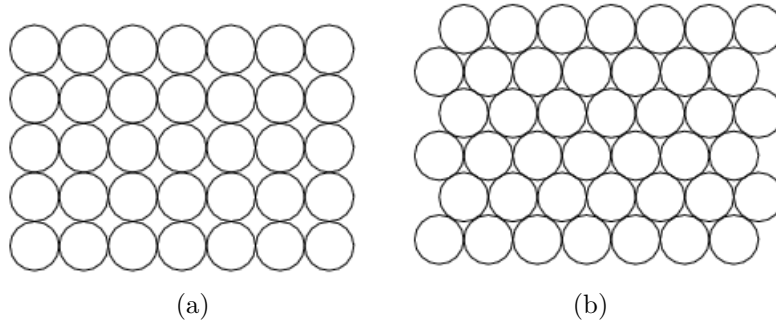


Figura 2.2: (a) Arranjo de círculos em formato quadricular e (b) arranjo de círculos em formato hexagonal.

no plano. Isso foi provado pelo matemático László Fejes Tóth [13, 14], em 1940. Essa maneira de dispor as esferas resulta em uma densidade de

$$\frac{1}{6}\pi\sqrt{3} \approx 0.9069.$$

No caso tridimensional, uma diposição aleatória das esferas apresenta um resultado experimental para a densidade de, aproximadamente, 0,65. Uma forma mais densa pode ser obtida da seguinte forma: primeiro, forma-se um arranjo em formato hexagonal em um plano qualquer, similar ao caso bidimensional; depois, colocam-se esferas nos “buracos” formados pelas esferas anteriores e, assim, sucessivamente. A cada passo, há duas possibilidades para o posicionamento das esferas, resultando em duas configurações, a cúbica e a hexagonal, conhecidas como *face-centered cubic close packing* e *hexagonal close packing*, respectivamente. Cada uma delas pode ser vista na Figura 2.3.

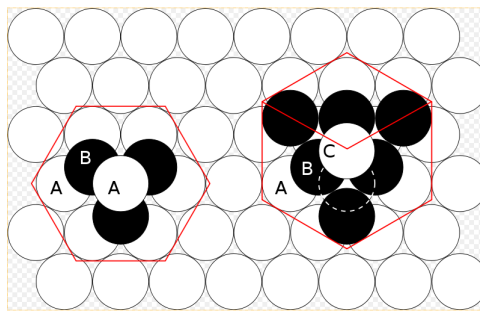


Figura 2.3: As configurações *hexagonal close packing*, à esquerda, e a *face-centered cubic close packing*, à direita. As letras indicam quais camadas são as mesmas. Há duas camadas ‘A’ na configuração hexagonal, onde todas as esferas estão na mesma posição. Todas as camadas na configuração cúbica são diferentes. Esta última pode ser convertida na primeira se transladarmos a camada ‘C’ de forma que a esfera desenhada ocupe a região pontilhada.

Ambas as configurações apresentam a mesma densidade média, com valor

$$\frac{\pi\sqrt{18}}{18} \approx 0.74048.$$

Essa é a melhor densidade que pode ser obtida em três dimensões, segundo a conjectura de Kepler. Ela diz que nenhum arranjo de esferas de tamanhos iguais preenchendo uma região tem densidade média maior do que a do *cubic close packing* e a do *hexagonal close packing*. Essa conjectura foi recentemente provada por Hales [15].

O que foi apresentado até agora trata de esferas de tamanhos iguais e não leva em consideração a região contenedora. Mas, em geral, há problemas de empacotamentos de esferas com diferentes diâmetros e em regiões de formatos definidos. Nas figuras 2.4a e 2.4b, obtidas em [16], podem ser vistos exemplos de empacotamentos de esferas iguais em quadrados e círculos. Esses problemas possuem aplicação em cortes de materiais, ou seja, quando têm-se, por exemplo, tecidos ou placas de metal em formatos retangulares ou circulares, e, a partir deles, deseja-se cortar círculos para determinado fim.

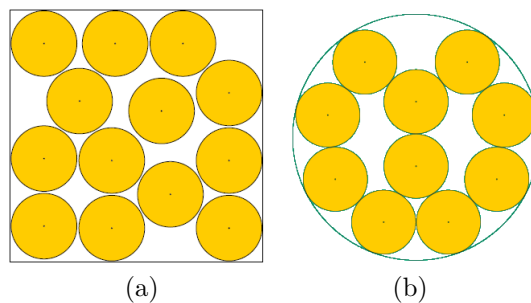


Figura 2.4: Empacotamento de círculos em regiões compactas do  $\mathbb{R}^2$ . Nesses exemplos, todos os círculos possuem diâmetro 1,3 unidade de comprimento. (a) 13 círculos foram empacotados em um quadrado de lado 5 unidades de comprimento; (b) 10 círculos em um círculo maior de diâmetro 5.

Um exemplo de aplicação onde as esferas possuem tamanhos diferentes é o próprio problema em estudo, relacionado ao planejamento do tratamento *Gamma Knife*, onde a região contenedora não possui forma definida, sendo representada pelo tumor [17, 18]. Considere os seguintes parâmetros:

- $K$ , a quantidade de diferentes raios ( $K = 4$ , na aplicação);
- $r_k$ ,  $k = 1, \dots, K$ , o raio correspondente;
- $L$ , a quantidade de esferas disponíveis para cada raio; e
- $P = \{(x, y, z) \in \mathbb{R}^3 \mid a_mx + b_my + c_mz \geq d_m, m = 1, \dots, M\}$ , para algum  $M > 0$ .

Em [17], Sutou e Dai apresentam um modelo de programação matemática cuja solução é um empacotamento de, no máximo,  $L$  esferas, das  $KL$  disponíveis, em um politopo  $P$ , como definido acima. São utilizadas as seguintes variáveis:

- $c_i = (x_i, y_i, z_i)$ ,  $i = 1, \dots, L$ , representando o centro da esfera  $i$ ; e
- $t_{ik} \in \{0, 1\}$ , assumindo valor 1 se a esfera  $i$  estiver associada ao raio  $k$ .

Dessa forma, o problema de otimização foi formulado como em (2.11)-(2.16):

$$\max \quad (4/3)\pi \sum_{i=1}^L \sum_{k=1}^K r_k^3 t_{ik} \quad (2.11)$$

$$\text{s. a } \|c_i - c_j\|^2 \geq \left( \sum_{k=1}^K r_k t_{ik} + \sum_{k=1}^K r_k t_{jk} \right)^2, \quad \forall i \leq j, i \neq j \quad (2.12)$$

$$\frac{|a_m x_i + b_m y_i + c_m z_i - d_m|}{\sqrt{(a_m^2 + b_m^2 + c_m^2)}} \geq \sum_{k=1}^K r_k t_{ik}, \quad \forall i, \forall m \quad (2.13)$$

$$a_m x_i + b_m y_i + c_m z_i - d_m \geq 0, \quad \forall i, \forall m \quad (2.14)$$

$$\sum_{k=1}^K t_{ik} \leq 1, \quad \forall i \quad (2.15)$$

$$t_{ik} \in \{0, 1\}, \quad \forall i, \forall k \quad (2.16)$$

As restrições (2.12) e (2.14) asseguram que as esferas não se interceptam e que seus centros pertencem ao politopo. As restrições (2.15) e (2.16) garantem que, no máximo, um raio será escolhido para cada esfera. Isto é, se  $t_{ik} = 1$ , a esfera  $i$ , com raio  $r_k$  é empacotada. Caso contrário, a esfera não é empacotada. Junto com as restrições (2.14) e (2.16), as restrições (2.13) exprimem que a distância entre o centro de uma esfera e a fronteira do politopo é, pelo menos, o raio dessa esfera.

Os autores ainda reformularam o problema baseado nas seguintes idéias:

- Seja  $e_m = \sqrt{a_m^2 + b_m^2 + c_m^2}$ . Por (2.14), restrição (2.13) pode ser escrita como

$$a_m x_i + b_m y_i + c_m z_i - d_m \geq e_m \sum_{k=1}^K r_k t_{ik}, \quad \forall i, m.$$

Como o lado direito é não-negativo, a equação acima implica em (2.14).

- As variáveis binárias  $t_{ik}$  serão substituídas pelas desigualdades

$$t_{ik}(t_{ik} - 1) \geq 0 \quad \text{e} \quad 0 \leq t_{ik} \leq 1.$$

Mas note que  $t_{ik} \leq 1$  é implicado pelas restrições (2.15).

Com isso, temos a reformulação como em (2.17) e (2.22):

$$\max \quad \sum_{i=1}^L \sum_{k=1}^K r_k^3 t_{ik} \quad (2.17)$$

$$\text{s. a} \quad \|c_i - c_j\|^2 \geq \left( \sum_{k=1}^K r_k t_{ik} + \sum_{k=1}^K r_k t_{jk} \right)^2, \quad \forall i \leq j, i \neq j \quad (2.18)$$

$$a_m x_i + b_m y_i + c_m z_i - d_m \geq e_m \sum_{k=1}^K r_k t_{ik}, \quad \forall i, \forall m \quad (2.19)$$

$$t_{ik}(t_{ik} - 1) \geq 0, \quad \forall i, \forall k \quad (2.20)$$

$$\sum_{k=1}^K t_{ik} \leq 1, \quad \forall i \quad (2.21)$$

$$t_{ik} \geq 0, \quad \forall i, \forall k \quad (2.22)$$

Note que a constante  $(4/3)\pi$  foi omitida na função objetivo e que as funções quadráticas nas restrições (2.18) não são nem convexas nem côncavas.

### 2.2.3 O problema do Kissing Number

Determinar o número máximo de hiperesferas unitárias que podem tocar uma hiperesfera central, também unitária, sem sobreposição, num espaço euclidiano  $n$ -dimensional, é um problema geométrico conhecido como *Kissing Number Problem* (KNP) [19]. As únicas interseções permitidas são os eventuais pontos de tangência.

O problema já foi resolvido para duas, três e quatro dimensões. Em uma ou em duas dimensões, as esferas são, na verdade, dois pontos e círculos, respectivamente. Em uma dimensão, é trivial. A solução do problema é 2, como pode ser visto na Figura 2.5a. Em duas dimensões, a solução compreende seis círculos ao redor do círculo central, conforme a Figura 2.5b.

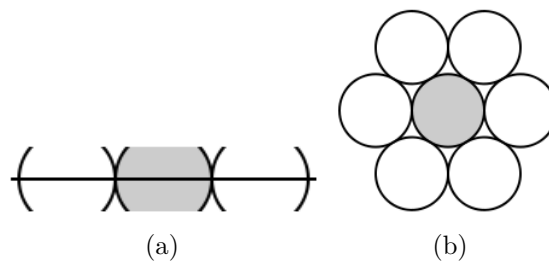


Figura 2.5: (a) Solução do KNP em uma dimensão e (b) Solução do KNP em duas dimensões. A parte cinza indica a “esfera” central.

A situação é bem mais complicada no  $\mathbb{R}^3$ . Newton acreditava, e estava certo, que o *kissing number* em três dimensões era 12, mas a prova completa foi elaborada bem mais tarde, por Leech [20]. Após colocar as 12 esferas ao redor da esfera central, ainda sobra uma quantidade significativa de espaço livre, mas não o suficiente para encaixar uma 13ª esfera. Uma configuração da solução pode ser vista na Figura 2.6.

Para o caso de quatro dimensões, Oleg Mousin provou, em 2003, que a solução

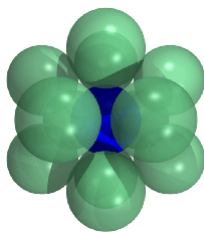


Figura 2.6: Solução do KNP em três dimensões. Essa configuração pode ser obtida arrumando as esferas de forma que seus pontos de tangência com a esfera central correspondam aos vértices de um icosaedro.

é formada por 24 hiperesferas [21, 22]. O *kissing number* em  $n$ -dimensões, com  $n > 4$ , é conhecido somente para  $n=8$  e  $n=24$ , com valores 240 e 196560 [23, 24], respectivamente.

Em [25], são apresentados dois modelos de programação matemática não-lineares e não-convexos para a solução do KNP. O primeiro deles consiste em uma variável sendo maximizada, chamada de *indicador de viabilidade*. Se o valor ótimo dessa variável  $\alpha$  é maior que um limite fixado, o arranjo de  $N$  esferas em  $D$  dimensões é aceito como viável. Assim, uma solução ótima para o KNP será o mínimo  $N$  tal que o problema com  $N + 1$  esferas seja inviável. Sejam  $\{x^i\}_{i=1,\dots,N}$  as posições dos centros das esferas ao redor da esfera central. Temos:

$$\max \quad \alpha \tag{2.23}$$

$$\text{s. a} \quad \|x^i\|^2 = 4, \quad \forall i \leq N \tag{2.24}$$

$$\|x^i - x^j\|^2 \geq 4\alpha, \quad \forall i < j \leq N \tag{2.25}$$

$$\alpha \geq 0 \tag{2.26}$$

$$x^i \in \mathbb{R}^D, \quad \forall i \leq N, \tag{2.27}$$

onde  $\|\cdot\|$  é a norma euclideana. As restrições 2.24 asseguram que os centros das esferas estarão a uma distância de 2 unidades do centro da esfera central, uma vez fixado o centro desta na origem. Tomando como 1 o valor do limite da variável  $\alpha$  que torna a solução viável, teremos que as restrições 2.25 impedem que as esferas se sobreponham.

O outro modelo apresentado se baseia na mesma idéia, com a diferença de que os centros das esferas são definidos por coordenadas esféricas, com relação ao centro da esfera central.



# Capítulo 3

## Otimização Global Mista

### 3.1 Definição do Problema

Uma grande variedade de problemas de otimização envolvem variáveis inteiras ou discretas, além de variáveis contínuas. São os chamados problemas **mistos** ou **combinatórios**. Problemas como esses surgem de diversas aplicações e são comumente não-lineares. Essas classes de problemas de otimização são conhecidas como **Problemas de Programação Não-Linear Inteira Mista**, ou MINLP (*Mixed Integer Nonlinear Problem*). A formulação geral de um problema do tipo MINLP pode ser escrita como:

$$\begin{array}{ll} \text{minimizar} & f(\mathbf{x}, \mathbf{y}) \\ \text{sujeito a} & \mathbf{h}(\mathbf{x}, \mathbf{y}) = 0 \\ & \mathbf{g}(\mathbf{x}, \mathbf{y}) \leq 0 \\ & \mathbf{x} \in X \\ & \mathbf{y} \in Y . \end{array}$$

Acima,  $\mathbf{x}$  representa um vetor de variáveis contínuas,  $\mathbf{x} \in \mathbb{R}^n$ , e  $\mathbf{y}$  um vetor de variáveis inteiras,  $\mathbf{y} \in \mathbb{Z}^m$ . O sistema de equações  $\mathbf{h}(x, y) = 0$  representa as **restrições de igualdade**, com  $\mathbf{h} \in \mathbb{R}^p$ , e o sistema de inequações  $\mathbf{g}(x, y) \leq 0$  representa as **restrições de desigualdade**, com  $\mathbf{g} \in \mathbb{R}^q$ . A função real  $f(x, y)$  é denominada **função objetivo**. Essa é a função a ser otimizada, a fim de se encontrar seu máximo ou mínimo local. Não nos importaremos com a direção de otimização, já que é possível transformar um problema de minimização em um problema de maximização, e vice-versa, bastando apenas trocar a função objetivo  $f$  por  $-f$ . Ainda,  $X$  é um subconjunto do  $\mathbb{R}^n$  e  $Y$  um subconjunto de  $\mathbb{Z}^m$ , ambos representando os limites das variáveis  $\mathbf{x}$  e  $\mathbf{y}$ , respectivamente.

Sem perda de generalidade, podemos assumir que  $\mathbf{y}$  é um vetor binário, desde

que  $Y$  represente uma caixa em  $\mathbb{Z}^m$ . Observe [26] que as variáveis inteiras  $\mathbf{y}$  com dados limites superiores e inferiores,

$$\mathbf{y}^L \leq \mathbf{y} \leq \mathbf{y}^U,$$

podem ser expressas através de variáveis 0-1, ou variáveis binárias, denotadas por  $\mathbf{z}$ , de acordo com a seguinte fórmula:

$$y = y^L + z_1 + 2z_2 + 4z_3 + \cdots + 2^{N-1}z_N,$$

onde  $N$  é o número mínimo de variáveis necessárias, dado por:

$$N = 1 + \left\lceil \frac{\log(y^U - y^L)}{\log 2} \right\rceil.$$

Podemos, então, reescrever a formulação geral de um problema do tipo MINLP da seguinte maneira:

$$\begin{aligned} & \textit{minimizar} && f(\mathbf{x}, \mathbf{y}) \\ & \textit{sujeito a} && \mathbf{h}(\mathbf{x}, \mathbf{y}) = 0 \\ & && \mathbf{g}(\mathbf{x}, \mathbf{y}) \leq 0 \\ & && \mathbf{x} \in X \subseteq \mathbb{R}^n \\ & && \mathbf{y} \in Y = \{0, 1\}^m \end{aligned}$$

Solucionar problemas de otimização não-lineares inteiros mistos da forma acima envolve enfrentar duas grandes dificuldades associadas à natureza do problema. Conforme o número de variáveis binárias aumenta, lidamos com um problema altamente combinatório, classificado como NP-completo [27]. Além disso, devido às não-linearidades, os problemas do tipo MINLP, mesmo após relaxadas as restrições de não-integralidade, são geralmente não-convexos, apresentando vários ótimos locais. Encontrar um ótimo global para MINLP não-convexos é uma tarefa NP-hard [28], visto que mesmo a otimização global de problemas não-lineares restritos pode ser NP-hard [29].

Apesar da complexidade desses tipos de problemas, progressos tanto teóricos quanto computacionais têm sido alcançados nessa área. Como resultado, diversos algoritmos foram propostos. No caso especial em que são convexos, há códigos bem conhecidos para solução de problemas inteiros mistos não-lineares, como SBB [30], MINLP\_BB [31], BONMIN [32] e FILMINT [33]. Para o caso geral, não-convexo, podemos citar o COUENNE [34]. Na próxima seção, apresentaremos algumas metaheurísticas que podem ser utilizadas para resolver problemas do tipo MINLP.

## 3.2 Metaheurísticas de Otimização Global

Funções de muitas variáveis apresentam uma enorme quantidade de máximos e mínimos locais. Encontrar um ótimo local é uma tarefa relativamente fácil com o emprego de métodos de otimização local. Porém, encontrar um máximo ou mínimo global de uma função multi-variada é muito mais desafiador.

Em se tratando de problemas reais, é bastante comum aplicações apresentarem modelos não-lineares, com muitos máximos e mínimos locais. Como exemplo, ocorre em áreas de biotecnologia, em modelagem científica, em análise de dados, em processos de controle, entre outros. O uso de métodos de otimização local, dependendo do ponto de partida da busca, resultará, muitas vezes, em ótimos locais. Com o intuito de encontrar um ótimo global, precisamos de um método de busca global.

Bastante progresso tem sido obtido em buscas por soluções comprovadamente globais, com o uso de programação dinâmica, planos de corte e *branch and cut*. Porém, muitos problemas de otimização combinatória difíceis, ou NP-hard, ainda não foram resolvidos e requerem boas metaheurísticas.

Nessa seção, apresentaremos algumas das mais importantes metaheurísticas de otimização global existentes, como os Algoritmos Genéticos, o *Simulated Annealing*, a Busca Tabu, o *Variable Neighborhood Search* (VNS) e o *Greedy Randomized Adaptive Search Procedure* (GRASP).

### 3.2.1 Algoritmos Genéticos

Os algoritmos genéticos fazem parte de uma família de métodos computacionais baseados na teoria biológica da evolução dos seres vivos. Segundo a teoria da evolução natural, proposta por Charles Darwin, a evolução de populações naturais por várias gerações ocorre de acordo com os princípios de seleção natural e sobrevivência dos mais aptos. A partir dela, implementa-se uma técnica de busca utilizada para resolver problemas de otimização.

Certos conceitos da biologia são empregados, como mutação, recombinação ou *crossover*, hereditariedade e seleção natural. Alguns termos também são utilizados, como:

- cromossomo: cada uma das soluções, ou indivíduos; e
- gene: cada elemento do cromossomo.

A evolução inicia-se a partir de um conjunto de soluções e é realizada através de gerações. Esse conjunto inicial é, geralmente criado aleatoriamente. Cada solução representa um indivíduo dentro de uma população. A representação de um indivíduo deve ser capaz de representar todo o conjunto de valores do espaço de busca e precisa

ter tamanho finito [35]. Tradicionalmente, as soluções são representadas por vetores binários.

O algoritmo termina quando um número máximo de gerações tiver sido produzida ou quando um nível satisfatório, com relação à função objetivo, tiver sido alcançado ou, ainda, com o tempo máximo de execução. Um pseudo-código para o algoritmo genético pode ser visto na Figura 3.1.

---

**Entrada:**  $f$ , função a ser maximizada  
**Entrada:**  $numCruzamentos$ , quantidade de cruzamentos  
**Entrada:**  $numMutacoes$ , quantidade de mutacoes  
**Dados:**  $cruzamento(\cdot, \cdot)$ , função de cruzamento  
**Dados:**  $mutacao(\cdot)$ , função de mutação  
**Saída:** Solução  $S$

**Inicializar** população  $P$  ;  
**enquanto** *critério de parada não satisfeito* **faça**

**Selecionar** subpopulação  $sP \subseteq P$  de pais;

**para**  $i = 1 : numCruzamentos$  **faça**

**Escolher**  $S_1, S_2 \in sP$  aleatoriamente ;  
         $filho \leftarrow cruzamento(S_1, S_2)$  ;  
        **se**  $f(S_1) \geq f(S_2)$  **então**  
             $S_{aux} \leftarrow S_1$  ;  
        **senão**  
             $S_{aux} \leftarrow S_2$  ;  
        **fim**  
        **se**  $f(S_{aux}) \geq f(filho)$  **então**  
            **Substituir**  $S_{aux}$  por  $filho$  em  $P$  ;  
        **fim**

**fim**

**para**  $i = 1 : numMutacoes$  **faça**

**Selecionar**  $S_j \in P$  ;  
         $S_j \leftarrow mutacao(S_j)$  ;  
    **fim**

**fim**  
 $S \leftarrow$  melhor elemento de  $P$  ;

---

Figura 3.1: Pseudo-código do Algoritmo Genético.

---

Escolhem-se, então, os indivíduos que permanecerão na população e os que serão utilizados para reprodução. Nesse processo de seleção, há estratégias para manter a diversidade e preservar boas soluções. Dentre elas, temos:

- seleção elitista: seleciona indivíduos com melhor aptidão, ou melhor valor da função objetivo;

- seleção por roleta: é a mais comum e atribui uma probabilidade de escolha a cada indivíduo, de acordo com sua aptidão na população. Assim, indivíduos com melhor valor de função objetivo são mais prováveis de serem escolhidos para a reprodução; e
- seleção por classificação: ordena a população e atribui uma probabilidade de escolha a cada indivíduo, de acordo com seu posicionamento.

Uma vez selecionados os reprodutores, o próximo estágio é o cruzamento. O cruzamento, ou *crossover*, consiste na geração de novos indivíduos, a partir de dois existentes. Computacionalmente, há algumas maneiras de se implementar a troca de material genético. Algumas das formas mais comuns:

- cruzamento de um ponto: apenas um ponto de corte é utilizado, ou seja, informações presentes antes desse ponto se ligarão às informações presentes após esse ponto, como mostra a Figura 3.2;
- cruzamento multiponto: generalização do cruzamento de um ponto, aonde vários pontos de corte podem ser utilizados; e
- cruzamento uniforme: não utiliza pontos de cruzamento. Através de um parâmetro global, estabelece uma probabilidade a cada gene de ser trocado entre os pais.

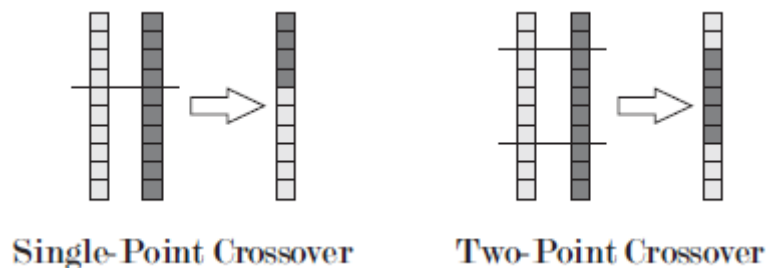


Figura 3.2: Exemplo de cruzamento com apenas um ponto de corte, à esquerda, e com dois pontos de corte, à direita.

Como resultado prático, cruzamento uniforme garante melhor desempenho que o cruzamento de dois pontos que, por sua vez, garante melhor desempenho que o cruzamento de um ponto [36].

Outra idéia presente é a mutação. A mutação é o fator que dá origem à diversidade. Esse processo é utilizado a fim de se evitar a convergência a um ótimo local. A mutação ocorre ao acaso e em baixa proporção, caso contrário, o algoritmo tenderia a um comportamento de busca randômica. No algoritmo, a mutação é representada por uma troca aleatória de um ou mais genes.

Os algoritmos genéticos podem ser utilizados tanto em problemas de otimização contínua como de otimização discreta. Utilizam regras de transição probabilísticas, e não determinísticas. Possuem a vantagem de não necessitarem de informações sobre derivadas de funções, utilizando-se apenas da função de aptidão, além de serem facilmente paralelizáveis.

### 3.2.2 Simulated Annealing

O algoritmo conhecido como *Simulated annealing* (SA) é uma metaheurística de otimização global inspirada na forma como os metais cristalizam, no processo de aquecimento, ou na forma como líquidos congelam. O aquecimento do metal é resfriado vagarosamente de forma a manter o sistema de derretimento em equilíbrio térmico, o que aumentará o tamanho dos seus cristais e reduzirá seus defeitos. Durante o resfriamento, os átomos do metal ficam mais ordenados. Se o resfriamento for prolongado além do normal, o sistema atingirá o estado limite  $T = 0K$ , o estado com menor energia possível. A temperatura inicial não deve ser muito baixa e o processo de resfriamento deve ser realizado suficientemente devagar para evitar que o sistema fique preso em um estado que represente um mínimo local de energia.

O método original era uma cópia exata do processo físico usado para simular um grupo de átomos em equilíbrio termodinâmico a uma dada temperatura. A idéia foi desenvolvida, de forma independente, por Scott Kirkpatrick, Daniel Gelatt e Mario Vecchi, em 1983 [37]. O método é uma adaptação do algoritmo de Metropolis-Hastings, um método de Monte Carlo para gerar amostras de estados de um sistema termodinâmico, inventado, junto com outros, por Nicholas Metropolis, em 1953 [38]. Um pseudo-código do algoritmo Simulated Annealing pode ser visto na Figura 3.3.

O objetivo é trazer o sistema de um estado inicial qualquer para o estado de menor energia possível. A cada iteração, o algoritmo considera um vizinho  $s'$  da solução atual  $s$  e, de acordo com uma certa distribuição de probabilidade, decide se realizará um movimento para o estado  $s'$  ou se manterá no estado  $s$ . Tipicamente, esse passo é repetido até que o sistema atinja um estado suficientemente bom para a aplicação ou até que alguma regra de parada estabelecida tenha sido alcançada.

Uma vez encontrado um vizinho  $s'$  de  $s$ , é testada a variação de energia  $\Delta E$  entre os estados, calculado pelo valor da função objetivo, como abaixo.

$$\Delta E = f(s') - f(s)$$

A seguir, calcula-se a probabilidade de transição para o novo estado encontrado. Essa probabilidade é especificada por uma função  $P(e', e, T)$  que depende tanto das energias dos dois estados,  $e' = f(s')$  e  $e = f(s)$ , quanto de um parâmetro  $T$  que representa a temperatura do sistema. Em geral, utiliza-se para  $P$  uma função com

---

**Entrada:**  $f$ , função objetivo a ser minimizada  
**Entrada:**  $mutacao(\cdot)$ , função de mutação  
**Dados:**  $T$ , temperatura do sistema  
**Dados:**  $t$ , índice de tempo  
**Dados:**  $\Delta E$ , potencial de energia  
**Saída:** Solução  $x^*$

**Criar** elemento inicial  $x_{new}$  ;  
 $x_{cur} \leftarrow x_{new}$  ;  
 $x^* \leftarrow x_{new}$  ;  
 $t \leftarrow 0$  ;

**enquanto** critério de parada não satisfeito **faça**  
 $\Delta E \leftarrow f(x_{new}) - f(x_{cur})$  ;  
**se**  $\Delta E \leq 0$  **então**  
 $x_{cur} \leftarrow x_{new}$  ;  
**se**  $f(x_{cur}) < f(x^*)$  **então**  
 $x^{star} \leftarrow x_{cur}$  ;  
**fim**  
**senão**  
 $T \leftarrow obterTemperatura(t)$  ;  
**Gerar** número aleatório  $numAleat$  ;  
**se**  $numAleat < e^{-\frac{\Delta E}{k_B T}}$  **então**  
 $x_{cur} \leftarrow x_{new}$  ;  
**fim**  
**fim**  
 $x_{new} \leftarrow mutacao(x_{cur})$  ;  
 $t \leftarrow t + 1$  ;  
**fim**  
**Retornar**  $x^*$  ;

Figura 3.3: Pseudo-código do algoritmo *Simulated Annealing*.

---

as seguintes características:

- $P > 0$  quando  $e' > e$ ;
- quando  $T \rightarrow 0$ , devemos ter  $P \rightarrow 0$ , se  $e' > e$ , e tendendo a um valor positivo, quando  $e' < e$ .

O primeiro item especifica que o sistema pode mover-se para o novo estado mesmo quando ele é uma solução pior que a atual, já que esse novo estado apresenta uma energia maior que a do estado atual. Essa característica é responsável por evitar que o método resulte em um ótimo local. Pelo segundo item, para valores suficientemente baixos de temperatura, indicando proximidade do equilíbrio, o sistema tenderá a mover-se apenas para estados que apresentam menor energia ou, em termos de função objetivo, para soluções melhores. Em particular, quando  $T$  se aproxima de

zero, o algoritmo se comporta de maneira similar a um método de descida, devido à probabilidade quase nula de aceitar um movimento que piore o valor da função objetivo.

Geralmente, usa-se, para o cálculo da probabilidade de transição, a função  $P$  definida abaixo, que utiliza o fator de Boltzmann, onde  $T$  é a temperatura medida em Kelvin e  $k_B = 1.380650424 \times 10^{-23} \text{ JK}^{-1}$  é a constante de Boltzmann:

$$P(e', e, T) = \begin{cases} 1 & \text{se } e' < e \\ e^{-\frac{\Delta E}{k_B T}} & \text{se } e' \geq e \end{cases} .$$

Definidas essas propriedades do algoritmo, a temperatura  $T$  tem uma grande importância no processo de controle de evolução dos estados do sistema. Inicialmente,  $T$  assume um valor elevado. A temperatura é, então, gradativamente diminuída, a uma certa quantidade de iterações, por um fator de resfriamento  $\alpha$ :

$$T_n = \alpha \cdot T_{n-1}, \quad 0 < \alpha < 1 .$$

Essa não é a única forma de se controlar a temperatura. Há, na verdade, uma grande variedade de métodos para determinar esse controle. Inclusive, já foram utilizados algoritmos genéticos para esse fim [39].

O procedimento termina quando a temperatura alcança um valor suficientemente próximo de zero e não há melhoria na solução, ou seja, quando o sistema atinge a estabilidade. Algumas variações do algoritmo incluem um reaquecimento, elevando novamente o valor de  $T$  e reiniciando o algoritmo sobre a melhor solução encontrada.

Algoritmos que utilizam essas idéias possuem prova de convergência assintótica [40], o que significa que eles irão, de fato, encontrar o ótimo global, se a execução for longa o suficiente. Esse resultado teórico, contudo, não é tão satisfatório, visto que o tempo necessário para garantir uma probabilidade de sucesso geralmente excede o tempo de uma busca completa no espaço solução [41]. Existem variações, denominadas *Simulated Quenching*, que aceleram o processo, mas abrem mão da garantia de convergência.

### 3.2.3 Busca Tabu

O algoritmo conhecido como Busca Tabu, ou *Tabu Search*, foi proposto por Glover [42, 43], na década de 1970, e é uma técnica muito semelhante à do *Simulated Annealing*. É um método de otimização voltado para buscas locais, que trabalha com apenas um candidato a solução por vez, e que se diferencia dos demais pelo uso de uma estrutura de memória.

A cada iteração, um conjunto de pontos na vizinhança da solução atual é visitado. O melhor vizinho, em se tratando de valor da função objetivo, é escolhido para



continuar o algoritmo. A escolha é realizada sem considerar o valor da solução corrente, ou seja, o vizinho é escolhido mesmo que não seja tão bom quanto a solução atual. Isso permite ao algoritmo escapar de ótimos locais. Porém, essa estratégia pode fazer com que o algoritmo cicle, encontrando seguidas vezes o mesmo ótimo local.

Como uma forma de contornar esse problema, utiliza-se uma lista com as últimas soluções visitadas pelo algoritmo, a **lista tabu**. É uma lista de tamanho fixo, cujo conteúdo são os movimentos proibidos pelo algoritmo. Toma-se uso, ainda, de uma **função de aspiração**. Ela define se o movimento a uma nova solução será realizado ou não, mesmo estando presente na lista tabu. Um pseudo-código do algoritmo da Busca Tabu pode ser visto na Figura 3.4.

---

**Entrada:**  $f$ , função objetivo a ser minimizada  
**Entrada:**  $mutacao(\cdot)$ , função de mutação  
**Entrada:**  $deletarElem(\cdot)$ , função para eliminar um elemento da lista tabu  
**Entrada:**  $adicionar(\cdot, \cdot)$ , função para adicionar um elemento na lista tabu  
**Entrada:**  $n$ , tamanho máximo da lista tabu  
**Entrada:**  $buscaLocal(x, listaTabu)$ , função que realiza uma busca na vizinhança de  $x$ , respeitando os movimentos da listaTabu e a função de aspiração  
**Dados:**  $X_{tabu}$ , a lista tabu  
**Saída:** Solução  $x^*$

**Criar** elemento inicial  $x_{new}$  ;  
 $x^* \leftarrow x_{new}$  ;  
 $X_{tabu} \leftarrow \emptyset$  ;

**enquanto** *critério de parada não satisfeito faça*  
    **se**  $buscaLocal(x_{new}, X_{tabu})$  **então**  
        **se**  $f(x_{new}) < f(x^*)$  **então**  
             $x^* \leftarrow x_{new}$  ;  
        **fim**  
        **se**  $|X_{tabu}| \geq n$  **então**  
             $X_{tabu} \leftarrow deletarElem(X_{tabu})$  ;  
        **fim**  
         $X_{tabu} \leftarrow adicionar(X_{tabu}, x_{new})$  ;  
    **fim**  
     $x_{new} \leftarrow mutacao(x^*)$  ;  
**fim**  
**Retornar**  $x^*$  ;

Figura 3.4: Pseudo-código do algoritmo *Tabu Search*.

---

A memória usada na Busca tabu, ou a lista tabu, pode ser tanto **explícita** como **atributiva** [42]. Memórias explícitas armazenam informações completas das soluções e, possivelmente, vizinhos altamente interessantes, mas ainda não visitados.

Já as memórias atributivas armazenam informações sobre os atributos das soluções que são modificados ao mover-se de uma solução a outra.

Dois importantes componentes da Busca Tabu são as estratégias de intensificação e diversificação [44, 45]. As estratégias de intensificação, em geral, têm por objetivo concentrar a busca em algumas regiões consideradas promissoras e fazem uso de memória explícita, enquanto que as estratégias de diversificação, em geral, têm por objetivo pesquisar regiões ainda pouco exploradas no espaço de soluções e fazem uso de memória atributiva. Alguns métodos de Busca Tabu também incluem listas tabus dinâmicas [43, 46, 47]. Essas listas têm o seu tamanho atualizado de acordo com o progresso da pesquisa.

### 3.2.4 Variable Neighborhood Search

A metaheurística conhecida como *Variable Neighborhood Search* (VNS), proposta por Hansen and Mladenovic [48–50], é um método de busca local que difere dos demais por basear-se em um modelo de **vizinhança dinâmica**. Ao contrário dos outros métodos de busca local, VNS permite mudanças na estrutura da vizinhança ao longo da busca.

O método explora diferentes vizinhanças, de tamanho crescente, da melhor solução encontrada. A cada iteração, há a geração de um vizinho da solução atual, uma busca local em torno desse vizinho e, caso seja encontrada uma solução melhor, um pulo para a nova solução.

Seja  $N_k$ ,  $k = 1, \dots, k_{max}$ , um conjunto pré-definido de estruturas de vizinhança e  $N_k(x)$  o conjunto dos vizinhos de  $x$  na vizinhança definida por  $N_k$ . Primeiro, o algoritmo encontra um vizinho  $x' \in N_k(x)$  da solução atual  $x$ . Depois, realiza uma busca local a partir de  $x'$ , resultando na solução  $x''$ . Finalmente, há o movimento do ponto  $x$  para o ponto  $x''$ , no caso deste ser uma solução melhor do que aquele. Caso contrário, aumenta-se o tamanho da vizinhança, com o uso de  $N_{k+1}$ , e repete-se os passos descritos até que algum critério de parada seja alcançado. Um pseudo-algoritmo para o VNS pode ser visto na Figura 3.5.

O algoritmo termina quando não há melhora na solução, depois de utilizadas todas as vizinhanças. A solução final é um ótimo local com respeito a todas as  $k_{max}$  vizinhanças.

Uma variação dessa metaheurística é o método *Variable Neighborhood Descent* (VND). Ele muda a estrutura da vizinhança toda vez que encontra um ótimo local. Há, ainda, uma outra variação, denominada *General Variable Neighborhood Search* (GVNS) [51, 52], que aplica duas séries de vizinhanças, não necessariamente diferentes, uma para a fase de encontrar um vizinho da solução atual e outra para a fase de busca local sobre esse vizinho. O GVNS tem obtido os melhores resultados

---

**Entrada:**  $f$ , função objetivo a ser maximizada  
**Entrada:**  $buscaLocal(\cdot)$ , função de busca local  
**Entrada:**  $maxIter$ , máximo de iterações do algoritmo  
**Entrada:**  $k_{max}$ , máximo de vizinhanças  
**Dados:**  $k$ , índice da vizinhança atual  
**Dados:**  $N_k(x)$ ,  $k$ -ésima vizinhança de  $x$   
**Saída:** Solução  $x^*$

```

para  $i = 1 : maxIter$  faça
     $k \leftarrow 1$  ;
    Gerar aleatoriamente elemento inicial  $x$  ;
    enquanto  $k \leq k_{max}$  faça
        Selecionar aleatoriamente  $x' \in N_k(x)$  ;
         $x'' \leftarrow buscaLocal(x')$  ;
        se  $f(x'') > f(x)$  então
             $x \leftarrow x''$  ;
             $k \leftarrow 1$  ;
        senão
             $k \leftarrow k + 1$  ;
    fim
fim
 $x^* \leftarrow x$  ;
Retornar  $x^*$  ;

```

Figura 3.5: Pseudo-código do algoritmo *Variable Neighborhood Search* (VNS).

---

nas aplicações da metaheurística VNS.

### 3.2.5 GRASP

A metaheurística conhecida como GRASP (*Greedy Randomized Adaptive Search Procedure*), desenvolvida por Feo e Resende [53], é um método iterativo, onde cada iteração consiste de duas fases: **construção** e **busca local**. Esses passos são repetidos muitas vezes, caracterizando-a como uma metaheurística *multi-start*.

Soluções viáveis são construídas a partir da combinação de métodos semi-gulosos com aleatoriedade. Se a solução não é viável, um procedimento para tentar recuperar a viabilidade é aplicado. Caso não haja sucesso, a solução é descartada e uma nova é gerada. Uma vez criada a solução, sua vizinhança é explorada, na fase de busca local, até que se encontre um ótimo local. O melhor ótimo local encontrado, dentre todas as iterações, é retornado como a solução da metaheurística.

Na fase de construção do GRASP, cada elemento formador de uma solução é analisado. Forma-se, então, uma lista somente com os elementos que podem ser incorporados à solução parcial, sob construção, sem que haja violação de sua via-

bilidade. De acordo com o valor de uma função gulosa pré-estabelecida, que varia conforme a aplicação, cada elemento é avaliado e qualificado. Uma outra lista, chamada **RCL** (*Restricted Candidate List*), é formada com os melhores elementos candidatos, ou seja, com os elementos cuja inclusão na solução parcial resultem no menor custo incremental (minimização). O próximo elemento que, de fato, será usado para formar a solução é escolhido aleatoriamente dentre os presentes na RCL. Um pseudo-algoritmo da fase de construção pode ser visto na Figura 3.6.

---

**Entrada:**  $\alpha$ , parâmetro da LCR  
**Saída:** Solução  $S$

$S \leftarrow \emptyset$  ;  
**Inicializar** o conjunto de candidatos  $C$  ;  
**Calcular** o custo incremental  $c(e)$  para todo  $e \in C$  ;

**enquanto**  $C \neq \emptyset$  **faça**  
     $c_{min} \leftarrow \min\{c(e) \mid e \in C\}$  ;  
     $c_{max} \leftarrow \max\{c(e) \mid e \in C\}$  ;  
    **Construir** a lista de candidatos restrita:  
         $LCR \leftarrow \{e \in C \mid c(e) \leq c_{min} + \alpha(c_{max} - c_{min})\}$  ;  
    **Selecionar** aleatoriamente  $s \in LCR$  ;  
    Incorporar  $s$  na solução:  $S \leftarrow S \cup \{s\}$  ;  
    **Atualizar** o conjunto de candidatos  $C$  ;  
    **Recalcular** o custo incremental  $c(e)$  para todo  $e \in C$  ;  
**fim**

**Retornar**  $S$  ;

Figura 3.6: Pseudo-algoritmo da fase de construção do GRASP.

---

O parâmetro  $\alpha \in [0, 1]$ , na Figura 3.6, é o responsável por controlar a quantidade de gula e aleatoriedade na formação da lista RCL. Quando  $\alpha = 0$ , temos uma escolha puramente gulosa, enquanto que, para  $\alpha = 1$ , temos uma escolha totalmente aleatória. Valores de  $\alpha$  no intervalo  $(0, 1)$  mesclam as duas estratégias no algoritmo.

Há resultados [54], porém, que mostram que usar um valor fixo para  $\alpha$  pode impedir soluções boas de serem encontradas, um problema que seria contornado caso outro valor de  $\alpha$  tivesse sido utilizado. Uma alternativa foi proposta por Prais e Ribeiro [54] e denominada *Reactive GRASP*. Eles propuseram que o parâmetro  $\alpha$  fosse modificado automaticamente de acordo com a qualidade das soluções obtidas.

A fase de busca local tem o objetivo de melhorar a solução criada no estágio de construção, de forma iterativa, substituindo sucessivamente a solução atual pela melhor solução encontrada na sua vizinhança. A busca termina quando nenhuma solução melhor é encontrada na vizinhança da solução atual. Um pseudo-algoritmo da fase de busca local pode ser visto na Figura 3.7. A eficácia da busca local depende tanto da estrutura da vizinhança quanto da estratégia de busca, da velocidade de

cálculo dos vizinhos, do ponto inicial e, ainda, do próprio procedimento de construção.

---

**Entrada:**  $S$ , solução criada na fase de construção  
**Entrada:**  $f$ , função a ser minimizada  
**Entrada:**  $N$ , estrutura de vizinhança  
**Saída:** Solução  $S$  melhorada

**enquanto**  $S$  não é um ótimo local **faça**  
    **Encontrar**  $S' \in N(S)$  com  $f(S') < f(S)$  ;  
     $S \leftarrow S'$  ;  
**fim**

**Retornar**  $S$  ;

Figura 3.7: Pseudo-algoritmo da fase de busca local do GRASP.

---

O GRASP também utiliza estratégias de intensificação e diversificação [55], conforme apresentado na seção sobre a Busca Tabu. Na Figura 3.8, encontra-se um pseudo-algoritmo para o GRASP.

---

**Entrada:**  $f$ , função a ser minimizada  
**Entrada:**  $maxIter$ , máximo de iterações do algoritmo  
**Entrada:**  $\alpha$ , argumento do procedimento de construção  
**Entrada:**  $construcao(\cdot)$ , procedimento de construção da solução  
**Entrada:**  $buscaLocal(\cdot)$ , procedimento de busca local  
**Entrada:**  $reparar(\cdot)$ , procedimento para tornar solução viável  
**Saída:** Solução  $S^*$

$f^* \leftarrow \infty$  ;  
**para**  $k=1:maxIter$  **faça**  
     $S \leftarrow construcao(\alpha)$  ;  
    **se**  $S$  não é viável **então**  
         $S \leftarrow reparar(S)$  ;  
    **fim**  
     $S \leftarrow buscaLocal(S)$  ;  
    **se**  $f(S) < f^*$  **então**  
         $S^* \leftarrow S$  ;  
         $f^* \leftarrow f(S)$  ;  
    **fim**

**fim**

**Retornar**  $S^*$  ;

Figura 3.8: Pseudo-algoritmo da metaheurística GRASP.

---

### 3.3 Spatial Branch-and-Bound

O algoritmo conhecido como *Spatial Branch-and-Bound* (sBB) [34], para problemas inteiros mistos não-lineares, tem bastantes semelhanças com o *Branch-and-Bound* comum, para problemas inteiros mistos lineares, mas apresenta detalhes adicionais. Uma diferença principal entre ambos os métodos é que, no sBB, todas as funções não-lineares são simbolicamente e recursivamente decompostas através de operadores simples até alcançarmos funções simples. Exemplos de operadores simples são: soma, produto, quociente, exponenciação, potência, logaritmo, seno, cosseno e valor absoluto. Com essa técnica, obtém-se uma reformulação com a característica de que todos os ótimos locais e globais do problema original são mapeados em ótimos locais e globais da reformulação [56, 57]. Essa decomposição é, geralmente, representada por uma coleção de grafos acíclicos direcionados e, dependendo de como é implementada, pode resultar em grafos com componentes conexas maiores e em menor quantidade, o que consecute em melhores resultados em procedimentos de aperto de limites.

O método sBB cria, ainda, uma hierarquia de nós, representada por uma árvore binária, conhecida com *árvore sBB*. Na raiz da árvore encontra-se o problema original. O restante da árvore é criado a partir de particionamentos dos problema. Os componentes essenciais de qualquer algoritmo sBB são:

- uma forma de calcular um **limite inferior** de cada problema da árvore sBB;
- uma maneira de calcular um **limite superior** ou uma **solução viável** de cada problema da árvore sBB;
- uma **técnica de *branching*** para particionar os problemas; e
- uma forma de **apertar os limites** para reduzir o espaço de soluções dos problemas, sem eliminar qualquer solução ótima.

O último item é importante por ter como consequência uma redução no tempo de execução do algoritmo. Um procedimento de aperto dos limites não é essencial para a convergência do método sBB, mas uma implementação eficiente pode aumentar drasticamente a performance do algoritmo. Um pseudo-código do algoritmo sBB pode ser visto na Figura 3.9.

Todos os problemas da árvore sBB sofrem uma relaxação linear. Como todas as soluções viáveis do problema original são também viáveis no problema relaxado, uma solução ótima da relaxação linear fornece um limite inferior para o valor ótimo do problema original. A qualidade da técnica de linearização aumenta quando os limites das variáveis são mais apertados. Dessa forma, o método sBB casa bem com

---

**Entrada:** Problema  $P$

**Entrada:**  $apertarLimites(\cdot)$ , procedimento para aperto de limites de um problema

**Saída:** Solução ótima de  $P$

**Inicializar** conjunto  $L$  de subproblemas:  $L \leftarrow \{P\}$  ;

**Inicializar** limite superior:  $z^u \leftarrow \infty$  ;

**enquanto**  $L \neq \emptyset$  **faça**

**Escolher**  $P_k \in L$  ;

$L \leftarrow L \setminus \{P_k\}$  ;

$apertarLimites(P_k)$  ;

**se**  $P_k$  *é viável* **então**

**Gerar**  $LP_k$ , relaxação linear de  $P_k$  ;

**repita**

**Solucionar**  $LP_k$  ;

$\bar{z}^k \leftarrow$  valor ótimo de  $LP_k$  ;

**Refinar** a linearização de  $LP_k$  ;

**até**  $\bar{z}^k$  *não apresentar melhora suficiente*;

$\hat{z}^k \leftarrow$  valor de um ótimo local de  $CP_k$  ;

$z^u \leftarrow \min\{z^u, \hat{z}^k\}$  ;

**se**  $\bar{z}^k \leq z^u - \epsilon$  **então**

**Escolher** variável  $x_i$  ;

**Escolher** um ponto de *branching*  $x_i^b$  ;

**Criar** subproblemas:

$P_{k-}$ , com  $x_i \leq x_i^b$  ;

$P_{k+}$ , com  $x_i \geq x_i^b$  ;

$L \leftarrow L \cup \{P_{k-}, P_{k+}\}$  ;

**fim**

**fim**

**fim**

**Retornar**  $z^u$  ;

Figura 3.9: Pseudo-algoritmo do método *Spatial Branch-and-Bound*.

---

problemas do tipo MINLP, já que cada partição do problema reduz o domínio de uma variável, melhorando a linearização e gerando um melhor limite inferior.

O processo de aperto dos limites das variáveis é responsável por reduzir o intervalo  $[x_i^L, x_i^U]$ , com  $x_i^L \leq x_i \leq x_i^U$ , onde  $x_i$  seria uma variável do problema, com garantia de não-alteração do valor ótimo do problema. Como resultado, temos uma redução no espaço solução e uma melhora na linearização. Um dos métodos de se apertar os limites das variáveis é conhecido como *Optimality-Based Bounds Tightening* (OBBT) [58]. O OBBT consome bastante tempo computacional e, portanto, é utilizado somente no nó raiz da árvore sBB ou até uma certa profundidade. Outro método, mais rápido, é o *Feasibility-Based Bounds Tightening* (FBBT) [58–60],

podendo ser aplicado em todos os nós da árvore.

Uma estratégia de *branching* eficaz visa minimizar o tamanho da árvore sBB e pode afetar fortemente a performance do algoritmo. Um dado problema da árvore sBB pode ser particionado com os seguintes objetivos:

- melhorar o limite inferior dos subproblemas resultantes;
- criar subproblemas de dificuldade similar, para manter a árvore sBB balanceada; ou
- eliminar a maior porção possível do conjunto solução.

Esses três objetivos são conflitantes. Pôr mais ênfase em um ou outro leva a diferentes técnicas de *branching*. A escolha geralmente é feita empiricamente, através de testes que avaliam sua performance.

Por fim, um limite superior é obtido quando, em um dado nó da árvore, a solução do problema relaxado é viável ao problema original. Quando a solução ótima da relaxação não é viável do problema original, tenta-se encontrar uma solução viável deste aplicando um *solver* não-linear a um problema modificado, onde as variáveis inteiras são fixadas a valores inteiros.



## Capítulo 4

# Modelo de Programação Não-Linear Mista

Muitos autores atacam o problema de planejamento do tratamento da radiocirurgia *Gamma Knife* sob o ponto de vista de um problema de empacotamento, não permitindo que as esferas se sobreponham. Dessa forma, o problema é descrito como um modelo matemático de empacotamento de esferas em regiões tridimensionais, almejando uma densidade de empacotamento superior a 0,9. Como já observado anteriormente, o empacotamento de esferas idênticas em um espaço euclidiano tridimensional ocupa no máximo 74% desse espaço. Como o tratamento em questão disponibiliza quatro diferentes tamanhos de esferas, uma densidade mais alta é possível de ser obtida, de acordo com os requerimentos do tratamento.

Contudo, desejamos, nesse trabalho, não apenas um recobrimento parcial, mas uma cobertura total da região. Para esse feito, a restrição de não-sobreposição das esferas é relaxada, sendo permitido que esferas distintas possuam um volume de interseção. Porém, essa situação deve ser evitada, já que a sobreposição das esferas indica uma aplicação dobrada de radiação na região compreendida pela interseção, enquanto que, idealmente, o tumor deva ser recoberto de forma homogênea. Temos, então, um problema que recai na classe de problemas de recobrimento, com o objetivo de utilizar o menor número possível de esferas.

Seguindo esse mesmo caminho, Liberti *et al.* [7] propuseram um modelo MINLP de programação matemática, cuja relaxação contínua é não-convexa. A partir dessa formulação, foi feita uma discretização do sólido a ser recoberto e, com o emprego de técnicas de linearização de restrições que utilizam variáveis 0-1, conseguiram aproximar o modelo por uma formulação de programação linear 0-1. Soluções interessantes foram obtidas com o uso do CPLEX. No entanto, esse enfoque pode se tornar inviável em discretizações mais finas, devido à enorme quantidade de variáveis.

Assim sendo, na seção a seguir, apresentamos um novo modelo para o problema, através, também, de um MINLP não-convexo.

## 4.1 Modelo Proposto

Seja  $S$  um conjunto de esferas. O **problema clássico de recobrimento ótimo** consiste em cobrir um sólido  $T \subseteq \mathbb{R}^3$  com as esferas pertencentes a  $S$ , mas utilizando o menor número possível dessas esferas. De outra forma, devemos posicionar as esferas de modo que cada ponto  $p \in T$  pertença, também, a alguma das esferas. Propomos, nesta seção, uma descrição formal do problema de recobrimento de um sólido tridimensional por esferas de diferentes diâmetros, em termos de uma formulação de programação matemática.

Seja  $S$  um conjunto formado por  $n$  esferas,  $S = \{S_1, \dots, S_n\}$ . Assim, para cada  $i \in \{1, \dots, n\}$ , seja  $S_i$  uma esfera centrada em  $x^i \in \mathbb{R}^3$  e com raio  $r_i > 0$ . Assumiremos que um recobrimento de  $T$  por esferas de  $S$  existe. Uma solução para o problema clássico de recobrimento ótimo pode ser definida por variáveis  $x$  e  $y$ , onde:

- $x \in T^n$ , e
- $y \in \{0, 1\}^n$ , sendo  $y_i = \begin{cases} 1, & \text{se } \mu(T \setminus \bigcup_{j \neq i} S_j) > 0 \\ 0, & \text{caso contrário} \end{cases}$ ,  $\forall i \in \{1, \dots, n\}$ .

Acima,  $\mu$  é a medida de Lebesgue em  $\mathbb{R}^3$ . A variável  $x = \{x^1, \dots, x^n\}$  representa um vetor contendo o centro de cada esfera e a variável binária  $y = \{y_1, \dots, y_n\}$  determina quais esferas do conjunto  $S$  estão presentes na solução. O termo

$$V_{S_i} = T \setminus \bigcup_{j \neq i} S_j$$

representa o que sobra do volume  $T$ , após subtraídas as regiões pertencentes a cada uma das esferas, com exceção da esfera  $S_i$ . Logo, se  $\mu(V_{S_i}) > 0$ , então a esfera  $S_i$  tem contribuição positiva no recobrimento e, portanto, deve participar da solução, resultando em  $y_i = 1$ .

Para apresentar a formulação, consideremos, primeiro, os parâmetros

$$\alpha_{ij}, \quad 1 \leq i < j \leq n, \quad e$$

$$c_i, \quad 1 \leq i \leq n.$$

O parâmetro  $\alpha_{ij}$  mede o quanto as esferas  $S_i$  e  $S_j$  estão permitidas de se sobrepor, conforme a Figura 4.1. Se  $\alpha_{ij} = 0$ , então  $|S_i \cap S_j| \leq 1$  ou, em palavras, as esferas não podem se sobrepor, podendo apenas tangenciar uma à outra. Devemos ter

$$\alpha_{ij} < \min\{r_i, r_j\}, \quad \forall 1 \leq i < j \leq n.$$

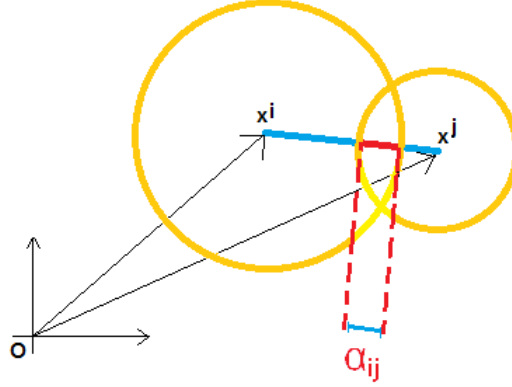


Figura 4.1: Representação dos parâmetros  $\alpha$ , representando a máxima interpenetração das esferas.

Podemos, ainda, escrever

$$\alpha_{ij} = \rho \min\{r_i, r_j\}, \quad 0 \leq \rho < 1.$$

Tomemos, ainda,

$$d_{ij} = r_i + r_j - \alpha_{ij},$$

medindo a distância entre os centros das esferas  $S_i$  e  $S_j$ , caso elas possuam interseção. Caso contrário,  $d_{ij}$  assume o valor  $r_i + r_j$ .

O parâmetro  $c_i$  mede a contribuição da esfera  $S_i$  para o recobrimento. Propomos, então, a seguinte formulação:

$$\max \quad \sum_{i=1}^n c_i y_i \tag{4.1}$$

$$\text{s. a} \quad \|x^i - x^j\|_2^2 \geq d_{ij}^2 (y_i + y_j - 1), \quad \forall i < j \leq n \tag{4.2}$$

$$x \in T^n \tag{4.3}$$

$$y \in \{0, 1\}^n \tag{4.4}$$

que define o *problema de recobrimento  $\alpha$ -ótimo* de  $T$ .

As variáveis de decisão são os  $x^i \in \mathbb{R}^3$ , centros das esferas, e os  $y_i \in \{0, 1\}$ , que determinam se cada esfera está ou não presente na cobertura. Importante ressaltar que, embora o objetivo seja o de minimizar a quantidade de esferas na solução, a direção de otimização é a maximização. No nosso modelo, estamos maximizando o volume das esferas no recobrimento, priorizando as maiores e, assim, diminuindo a quantidade de esferas na solução. Note, também, que, devido à região de segurança e às restrições quanto às interseções entre as esferas, definidas pelos parâmetros  $\alpha$ , talvez não seja possível recobrir completamente o conjunto  $T$ .

A partir dessa formulação, temos o seguinte lema, que define a existência de um conjunto de valores para os parâmetros, com os quais podemos retomar a solução do problema clássico:

**Lema.** Existem  $\{\alpha_{ij} \geq 0\}_{1 \leq i < j \leq n}$  e  $\{c_i \geq 0\}_{1 \leq i \leq n}$  para os quais uma solução ótima de (4.1)-(4.4) é uma solução ótima para o problema clássico de recobrimento ótimo.

*Demonstração:* Seja  $(x^*, y^*)$  uma solução ótima do problema clássico de recobrimento ótimo. Para cada par de esferas da solução, calculemos a distância  $dist_{ij}$  entre seus centros:

$$dist_{ij} = \|(x^i)^* - (x^j)^*\|.$$

Se  $dist_{ij} \geq r_i + r_j$ , tome  $\alpha_{ij} = 0$ . Caso contrário, tome

$$\alpha_{ij} = r_i + r_j - dist_{ij}.$$

Tome  $c_i = y_i^*$ , para todo  $1 \leq i \leq n$ . Então, afirmamos que  $(x^*, y^*)$  é uma solução ótima do problema de recobrimento  $\alpha$ -ótimo com os parâmetros  $\{\alpha_{ij}\}_{1 \leq i < j \leq n}$  e  $\{c_i\}_{1 \leq i \leq n}$  como acabamos de definir.

De fato, utilizando os parâmetros agora escolhidos, a solução ótima do problema clássico também será viável em (4.1)-(4.4). Basta mostrar, então, que ela terá o melhor valor da função objetivo. De fato, como os  $y_i^*$  assumem valores 0 ou 1, para todo  $i$ , e estamos usando  $c_i = y_i^*$ , temos que

$$\sum_{i=1}^n c_i y_i = \sum_{i|y_i^*=1} y_i.$$

Podemos observar que esse somatório atingirá seu valor máximo se tomarmos  $y_i = 1$  quando  $y_i^* = 1$ , ou seja, quando dermos às variáveis  $y_i$  os valores  $y_i^*$ . Portanto,  $(x^*, y^*)$  é uma solução ótima de (4.1)-(4.4) com os parâmetros tais como definidos. ■

Tanto o problema clássico quanto os problemas de recobrimento  $\alpha$ -ótimos não podem ser resolvidos explicitamente, a menos que a restrição 4.3 torne-se explícita. Temos casos simples quando  $T$  pode ser descrito através de equações matemáticas, como um paralelepípedo ou um elipsóide. Para um paralelepípedo, considere as medidas de seus lados como  $\{b_k > 0\}_{k \in \{1,2,3\}}$ . Assim,

$$T_{paral} = \{ (x_1, x_2, x_3) \mid 0 \leq x_k \leq b_k, 1 \leq k \leq 3 \}.$$

No caso do elipsóide, sejam  $\{b_k > 0\}_{k \in \{1,2,3\}}$  os seus raios. Dessa forma,

$$T_{elip} = \left\{ (x_1, x_2, x_3) \mid \sum_{k=1}^3 \left( \frac{x_k}{b_k} \right)^2 \leq 1 \right\}.$$

Como medida de segurança, em se tratando da aplicação no tratamento por *Gamma Knife*, aonde o volume  $T$  representa um tumor cerebral, costumam-se usar técnicas que impeçam que grande parte do volume das esferas ocupe a parte externa de  $T$ . Isso se dá pelo fato do tumor estar cercado por tecidos saudáveis. Não é aconselhável que tecidos ao redor da região-alvo recebam radiação. Uma forma de evitar que a solução contenha esferas posicionadas de forma a danificar tecidos saudáveis é restringir a ocupação da região externa a  $T$ , através de um novo parâmetro  $\epsilon$ , como pode ser visto na Figura 4.2. Nos referiremos a essa região como *região de segurança*.

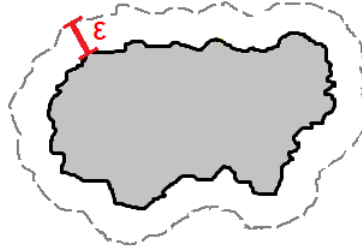


Figura 4.2: Representação da região de segurança, onde a região hachurada representa o tumor ou, de forma geral, o volume a ser recoberto.

Teríamos, como regiões de segurança para o paralelepípedo e para o elipsóide, por exemplo:

$$R_{paral} = \left\{ (x_1, x_2, x_3) \mid -\epsilon \leq x_k \leq b_k + \epsilon, 1 \leq k \leq 3 \right\} \quad e$$

$$R_{elip} = \left\{ (x_1, x_2, x_3) \mid \sum_{k=1}^3 \left( \frac{x_k}{b_k + \epsilon} \right)^2 \leq 1 \right\}.$$

Devemos observar que as variáveis que representam os centros das esferas não podem assumir qualquer valor na região de segurança. Por exemplo, se o centro de uma esfera dista da borda de região de segurança um valor menor que o raio dessa esfera, teríamos volume positivo na região externa a  $T$ . Portanto, precisamos restringir o domínio dessas variáveis, de acordo com os raios das esferas. Veja a Figura 4.3 para melhor entendimento.

Dessa forma, em um paralelepípedo ou em um elipsóide, por exemplo, teremos

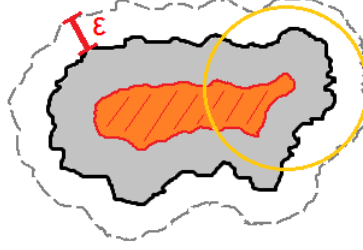


Figura 4.3: A região em cinza representa o volume a ser recoberto. Como a esfera não pode ultrapassar a região de segurança, o domínio da variável que representa o seu centro seria um conjunto parecido com a região em laranja.

como domínio da variável  $x_i$ :

$$D(x_i)_{paral} = \{ (x_1, x_2, x_3) \mid r_i - \epsilon \leq x_k \leq b_k + \epsilon - r_i, 1 \leq k \leq 3 \} \quad e$$

$$D(x_i)_{elip} = \left\{ (x_1, x_2, x_3) \mid \sum_{k=1}^3 \left( \frac{x_k}{b_k + \epsilon - r_i} \right)^2 \leq 1 \right\}.$$

## 4.2 Discretização

Quando o volume  $T$  é mais geral, de forma que a restrição 4.3 não possa ser feita explícita, é necessário o uso de técnicas de discretização. Na prática, em particular no problema do tratamento por *Gamma Knife*, como o volume é obtido através de imagens de ressonância magnética ou de tomografias computadorizadas, o volume já é um objeto discretizado, pela natureza das imagens. Porém, mesmo quando o volume é tal que a restrição 4.3 possa ser explicitada, talvez haja vantagem em discretizar  $T$ , de forma que o problema deixe de ser um MINLP e torne-se um problema apenas com variáveis binárias. Fazemos isso da forma como segue.

Seja  $x_k^i$  a  $k$ -ésima coordenada do centro da esfera  $i$ . Seja  $[a_k^i, b_k^i]$  o intervalo de variação da variável  $x_k^i$ , com  $a_k^i$  e  $b_k^i$  conhecidos, ou seja,

$$a_k^i \leq x_k^i \leq b_k^i.$$

Podemos discretizar cada um dos intervalos de definição das variáveis com a quantidade de pontos desejada. Não é necessário termos algo semelhante a um quadriculado e nem mesmo é necessário que cada par de pontos adjacentes tenham a mesma distância. Através de variáveis binárias  $\lambda$ , podemos escrever:

$$x_k^i = w_{k,1}^i \lambda_{k,1}^i + \cdots + w_{k,L_{ik}}^i \lambda_{k,L_{ik}}^i,$$

onde

- $L_{ik}$  é a quantidade de pontos usados na discretização do intervalo  $[a_k^i, b_k^i]$ ;
- $w_{k,j}^i$  são escolhidos como possíveis valores de assunção da variável;
- $a_k^i \leq w_{k,1}^i \leq \dots \leq w_{k,L_{ik}}^i \leq b_k^i$  ;
- $\lambda_{k,j}^i \in \{0, 1\}$ ,  $\forall j \in \{1, \dots, L_{ik}\}$  .

Dessa forma, forçamos os centros das esferas a assumirem valores discretos, em seus respectivos domínios. Mas, para efeito de unicidade e para utilizarmos como vantagem mais à frente, no cálculo do quadrado da variável, adicionaremos a restrição

$$\sum_{j=1}^{L_{ik}} \lambda_{k,j}^i = 1 \quad \forall i, k .$$

Como exemplo, suponha a variável  $x$  pertencente ao intervalo  $[1, 10]$  e queremos discretizar seu domínio com 10 pontos equispaçados. Teríamos:

$$x = \lambda_1 + 2\lambda_2 + \dots + 10\lambda_{10} .$$

Essa forma de discretizar as variáveis que representam o centro das esferas apresenta uma vantagem no momento do cálculo dos seus quadrados. No lado esquerdo da restrição 4.2, aparecem os valores  $(x_k^i)^2$ , que podem facilmente ser escritos como:

$$(x_k^i)^2 = (w_{k,1}^i)^2 \lambda_{k,1}^i + \dots + (w_{k,L_{ik}}^i)^2 \lambda_{k,L_{ik}}^i ,$$

já que apenas um dos  $\lambda_k^i$  assumirá valor 1.

Ao reescrevermos o lado esquerdo da restrição 4.2,

$$\|x^i - x^j\|_2^2 = \sum_{k=1}^3 \left[ (x_k^i)^2 - 2x_k^i x_k^j + (x_k^j)^2 \right] ,$$

nos deparamos com outro termo não-linear. Ao substituírmos pelas expressões de discretização, temos:

$$x_k^i x_k^j = \sum_{p=1}^{L_{ik}} \sum_{q=1}^{L_{jk}} w_{k,p}^i w_{k,q}^j \lambda_{k,p}^i \lambda_{k,q}^j .$$

Essa substituição mantém a não-linearidade, apresentando ainda produtos de variáveis. Aplicaremos, então, uma técnica de linearização envolvendo variáveis binárias. Fazemos

$$\delta_{kpq}^{ij} = \lambda_{k,p}^i \lambda_{k,q}^j ,$$

com

- $\delta_{kpq}^{ij} \geq 0$  ;
- $\delta_{kpq}^{ij} \leq \lambda_{k,p}^i$  ;
- $\delta_{kpq}^{ij} \leq \lambda_{k,q}^j$  ; e
- $\delta_{kpq}^{ij} \geq \lambda_{k,p}^i + \lambda_{k,q}^j - 1$  .

### 4.3 Diminuindo a combinatória

Com o objetivo de tentar reduzir ainda mais a enumeração ou a combinatória referente à escolha das esferas que pertencerão à solução, acrescentamos mais um conjunto de restrições, cuja idéia é explicada a seguir.

Queremos evitar que, durante a execução do algoritmo de solução do problema, esferas de um dado raio  $r$ , empregadas na solução, sejam trocadas por outras esferas de raio  $r$  ainda não utilizadas. Teoricamente, essa troca seria inútil, já que todas as esferas de raio  $r$  são idênticas.

Para tal feito, considerados conjuntos de esferas de mesmo raio, estabeleceremos, de certa forma, uma ordenação das esferas. Para melhor entendimento, suponha que esteja disponível o seguinte conjunto de esferas para realizar o recobrimento:

$$S = \{S_1^{r_1}, \dots, S_{k_1}^{r_1}, S_1^{r_2}, \dots, S_{k_2}^{r_2}, \dots\},$$

onde temos  $k_1$  esferas de raio  $r_1$ ,  $k_2$  esferas de raio  $r_2$  e assim por diante. Na nossa aplicação, temos apenas quatro diferentes valores para os raios: 2mm, 4mm, 7mm e 9mm. Adicionaremos, então, o seguinte conjunto de restrições:

$$y_i \geq y_{i+1}, \quad \forall i \notin \{k_1, k_2, k_3, k_4\}. \quad (4.5)$$

Com essas restrições, ao empregarmos esferas de mesmo raio, estaremos forçando o uso das esferas de menores índices na solução.

No capítulo seguinte, serão mostrados alguns resultados de testes realizados com a utilização de ambas as formulações apresentadas neste capítulo.



# Capítulo 5

## Resultados e Discussões

Nesta seção, serão apresentados resultados obtidos para o problema de recobrimento em questão, ao serem empregadas ambas as formulações apresentadas na seção 4. Para tentar solucionar os problemas, contamos com o auxílio de diversos solvers, como XPRESS, CPLEX, GUROBI e COUENNE. E utilizamos um programa de visualização que desenvolvemos para melhor analisar as soluções encontradas.

O problema de recobrimento em estudo apresenta uma certa dificuldade quanto à qualificação das soluções. Ao analisarmos duas soluções distintas de uma mesma instância do problema, não é evidente dizer qual solução é a melhor. Na verdade, não há, ainda, um critério de comparação entre soluções. Como exemplo, dadas duas soluções com um mesmo número de esferas no recobrimento, uma das soluções pode apresentar um recobrimento melhor que a outra, mas com as esferas ocupando um volume maior na parte externa da região-alvo, ou mesmo ter um volume muito grande de interseções entre esferas. Não é trivial julgar qual solução é mais favorável. Assim, introduziremos três grandezas, inspiradas em [61], cuja finalidade é apenas a de conhecer melhor as características da solução. Seja  $T$  o volume a ser coberto pelas esferas e  $U$  o volume da união das esferas. Calcularemos:

- ‘**cov**’ : Percentual do volume de  $T$  coberto pelas esferas;
- ‘**overlap**’ : Percentual do volume de  $T$  coberto por mais de uma esfera; e
- ‘**miscov**’ : Percentual do volume de  $U$  localizado na parte externa de  $T$ .

Essas grandezas podem ser melhor entendidas com o auxílio da Figura 5.1.

Para o cálculo dessas grandezas, podemos utilizar ou uma malha fina ou um algoritmo probabilístico conhecido como método de Monte Carlo [62]. O método, voltado ao cálculo do volume de uma região dada, consiste na geração aleatória de pontos e no cálculo da proporção de pontos gerados pertencentes à região. O método pode ser estendido para o cálculo do volume da união de vários objetos. Nesse trabalho, os objetos são esferas de diferentes raios.

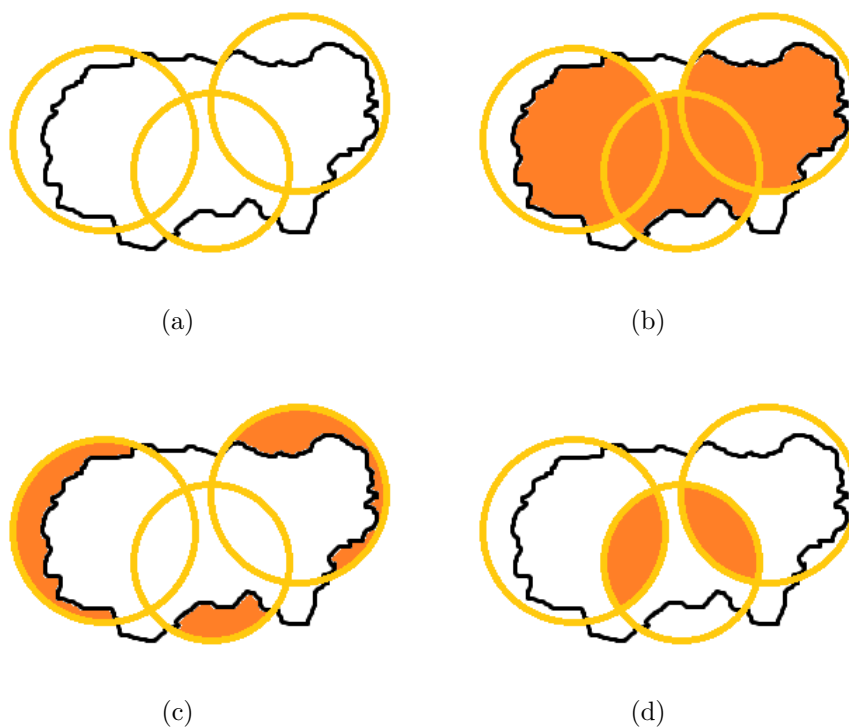


Figura 5.1: Exemplo bidimensional mostrando as regiões que correspondem às grandezas utilizadas na descrição de soluções. Em (a), a região a ser recoberta e as bolas presentes no recobrimento. As regiões hachuradas representam os parâmetros (b) cov, (c) miscov e (d) overlap.

Para o cálculo do percentual do volume de  $T$  coberto pelas esferas, representado por ‘cov’, geramos pontos em  $T$  e calculamos a proporção de pontos que pertencem a pelo menos uma das esferas. Esse mesmo algoritmo pode ser aproveitado para o cálculo do volume das interseções das esferas no interior de  $T$ , representado por ‘overlap’, verificando apenas se o ponto pertence a mais de uma esfera.

Para calcular o volume percentual de  $U$  externo a  $T$ , que chamamos de ‘miscov’, primeiro aplicamos o método de Monte Carlo a cada uma das esferas, a fim de obter o volume que cada esfera ocupa externamente a  $T$ . O volume interno da união pode ser calculado como segue:

$$Vol_{interno} = cov * Vol(T) .$$

Fazemos

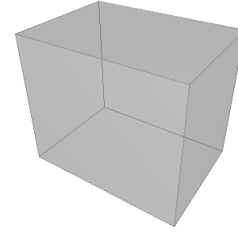
$$Vol_{Total} = Vol_{Interno} + Vol_{Externo} \quad e,$$

por fim,

$$miscov = \frac{Vol_{externo}}{Vol_{Total}} .$$

É importante notar que não há, para esse problema de recobrimento, um repositório de instâncias com o qual se possa trabalhar e realizar testes. Portanto, inicialmente, nos concentramos na realização de testes em um paralelepípedo de dimensões 14mm, 12mm e 10mm.

Utilizamos, ainda, para a região de segurança, um valor de  $\epsilon = 2$ . E, devido às dimensões do paralelepípedo, disponibilizamos para o recobrimento 20 esferas, de raios 2mm e 4mm.



Para os parâmetros do modelo, consideramos os seguintes valores:

$$c_i = r_i^3 \quad \text{e}$$

$$\alpha_{ij} = \min\{r_i, r_j\}/2.$$

Os  $c_i$  são os custos na função objetivo e representam o peso de cada esfera no recobrimento. Como o objetivo é minimizar a quantidade de esferas na solução, maior peso é dedicado às esferas de maior volume, pois recobrem um volume maior. Dessa forma, atribuímos aos custos os valores dos volumes das esferas, com omissão da constante  $(4/3)\pi$ .

Embora não haja uma definição de um bom critério para medir o quanto as esferas podem se interceptar, um volume grande de interseção, medido como ‘overlap’, não é desejado, do ponto de visto prático. Com relação ao planejamento do tratamento por *Gamma Knife*, interseções entre esferas representam quantidades de radiação dobradas na região. Portanto, apenas para efeito de testes, consideramos, para os parâmetros  $\alpha_{ij}$ , os valores correspondentes a metade do menor raio, dentre os raios envolvidos,  $r_i$  e  $r_j$ .

Por fim, para o modelo discretizado, utilizamos uma malha quadricular onde a distância entre pontos adjacentes foi tomada como 1mm.

## COUENNE

O COUENNE (Convex Over and Under ENvelopes for Nonlinear Estimation) [34, 63] é um solver que aplica o algoritmo conhecido como *spatial Branch-and-Bound* para solucionar problemas não-lineares, mistos e não-convexos. Utilizamos esse pacote com o modelo contínuo e, mesmo após 5 dias de execução, o solver não foi capaz de encontrar uma solução viável para o problema.

## GUROBI

O GUROBI [64] é uma biblioteca de alta qualidade para programação matemática, voltada a problemas lineares, quadráticos, lineares mistos e quadráticos mistos, apresentando rápida performance.

Nas figuras 5.2a e 5.2b, temos a solução obtida com o uso do GUROBI, com o modelo discretizado. O solver atingiu um dos critérios de parada definido como tempo máximo de execução de 5 horas. Foi encontrada uma solução com um valor de gap de 12,5% com o ótimo da relaxação linear.

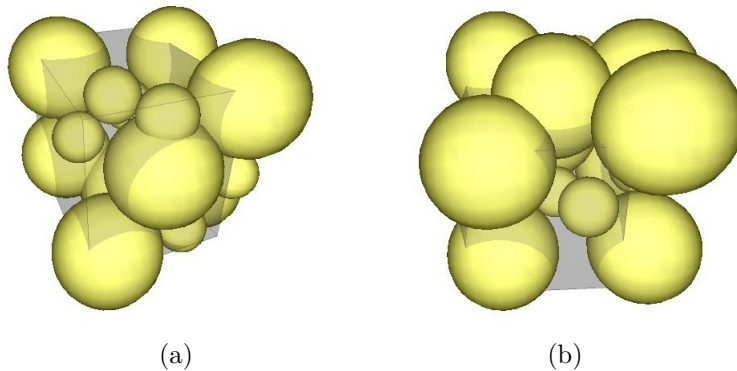


Figura 5.2: Solução encontrada pelo solver GUROBI para o recobrimento de um paralelepípedo de dimensões 14mm x 12mm x 10mm.

Para essa configuração das esferas, temos:	cov	=	59.27%
	miscov	=	58.41%
	overlap	=	3.19%

## CPLEX

O CPLEX [65] é um solver de programação matemática de alta-performance para programação linear, programação mista e programação quadrática.

Podemos ver os resultados do uso do CPLEX para solucionar a mesma instância do problema discretizado nas figuras 5.3a, 5.3b e 5.3c. Foi encontrada uma solução com um valor de gap de 2,27% com o valor ótimo da relaxação linear.

Essa solução também foi obtida após 5 horas de execução do algoritmo, tempo pré-estabelecido como um dos critérios de parada.

Para essa configuração das esferas, temos:	cov	=	77.17%
	miscov	=	50.88%
	overlap	=	11.19%

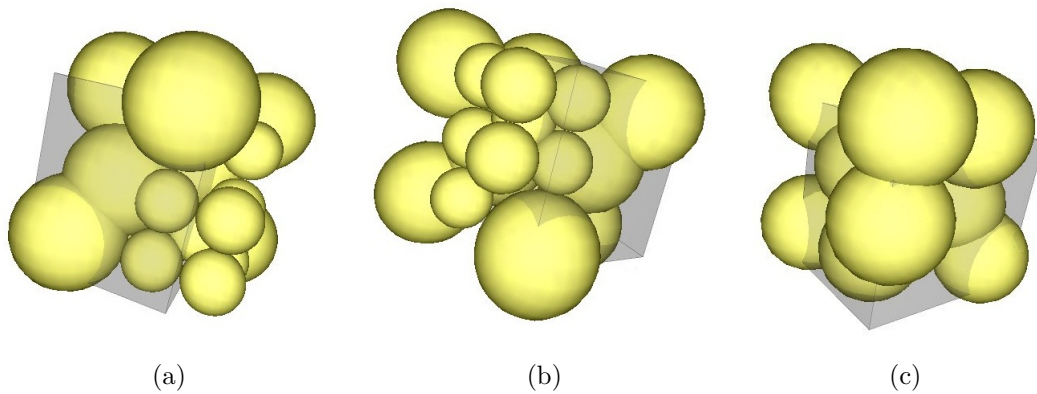


Figura 5.3: Solução encontrada pelo solver CPLEX para o recobrimento de um paralelepípedo de dimensões 14mm x 12mm x 10mm.

### XPRESS

Por fim, utilizamos ainda mais um solver para a realização dos testes, o XPRESS [66]. Testamos primeiro, o modelo sem as restrições (4.5), com o volume  $T$  original, ou seja, sem aproximação por discretização. Obtivemos o resultado apresentado nas figuras 5.4a e 5.4b, em um tempo de execução inferior a 1 segundo.

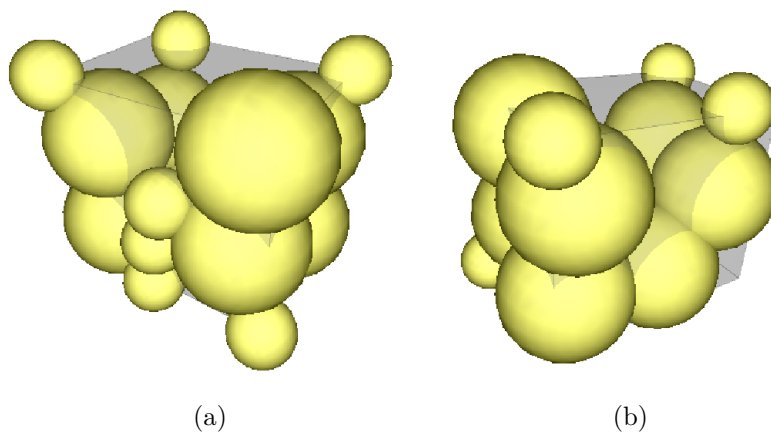


Figura 5.4: Solução encontrada pelo solver XPRESS para o recobrimento de um paralelepípedo de dimensões 14mm x 12mm x 10mm, sem as restrições de ordenação das esferas.

	cov	= 81.43%
Para essa configuração das esferas, temos:	miscov	= 32.13%
	overlap	= 10.44%

Testamos, então, com as restrições (4.5) e percebemos uma melhora na solução, em relação ao percentual de recobrimento. Essa nova solução pode ser vista nas figuras 5.5a e 5.5b, também encontrada em um tempo de execução inferior a 1 segundo.

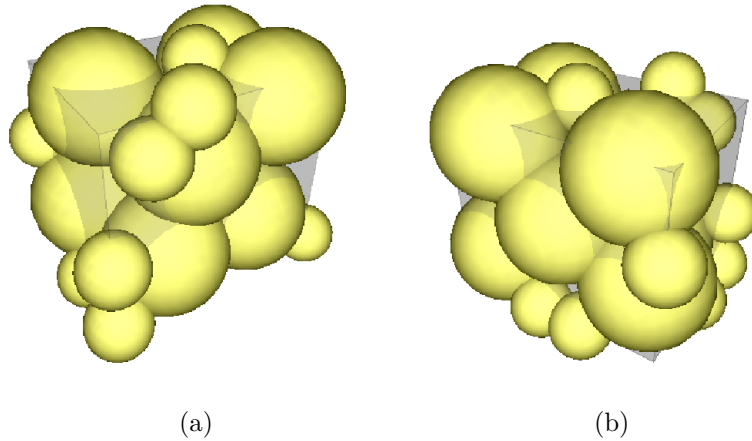


Figura 5.5: Solução encontrada pelo solver XPRESS para o recobrimento de um paralelepípedo de dimensões 14mm x 12mm x 10mm, com as restrições de ordenação das esferas.

	cov	=	96.65%
Para essa configuração das esferas, temos:	miscov	=	30.34%
	overlap	=	16.43%

Embora o modelo tenha apresentado essa solução com um bom nível de recobrimento, percebemos, através de testes adicionais, que ele possui uma característica indesejável. O modelo é bastante sensível aos parâmetros que o definem, como a quantidade de esferas disponíveis para serem utilizadas e o  $\epsilon$  que delimita a região de segurança. Esse efeito pode ser observado nos resultados dos dois testes a seguir.

Com a intenção de reduzir o volume ocupado pelas esferas na região externa a  $T$ , realizamos novo teste com  $\epsilon = 1,5mm$ . O resultado pode ser visto nas figuras 5.6a e 5.6b.

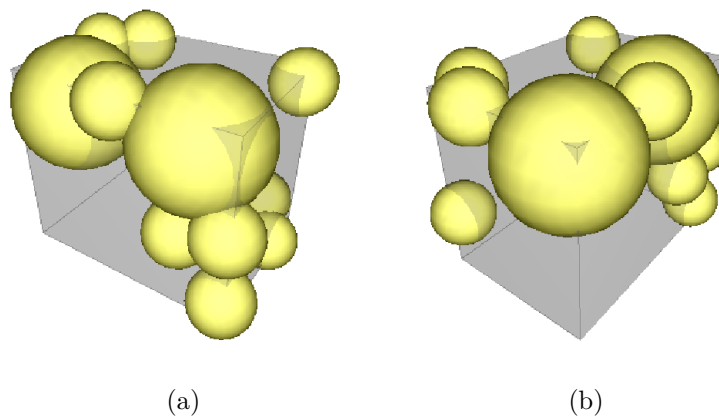


Figura 5.6: Solução encontrada pelo solver XPRESS para o recobrimento de um paralelepípedo de dimensões 14mm x 12mm x 10mm, com  $\epsilon$  reduzido para 1,5mm.

Para essa configuração das esferas, temos:

$$\begin{aligned} \text{cov} &= 36.26\% \\ \text{miscov} &= 31.15\% \\ \text{overlap} &= 1.48\% \end{aligned}$$

Solução ainda pior foi encontrada ao, retomada a instância original, disponibilizarmos apenas esferas de raio 2mm para o recobrimento. O resultado pode ser visto na Figura 5.7.

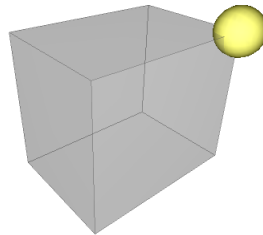


Figura 5.7: Solução encontrada pelo solver XPRESS para o recobrimento de um paralelepípedo de dimensões 14mm x 12mm x 10mm, tendo sido disponibilizadas apenas esferas de raio 2mm.

Para essa configuração das esferas, temos:

$$\begin{aligned} \text{cov} &= 0.27\% \\ \text{miscov} &= 86.98\% \\ \text{overlap} &= 0\% \end{aligned}$$

Resolvemos, então, analisar os resultados com o modelo discretizado, novamente com a instância em sua forma original. O resultado é o mostrado nas figuras 5.8a e 5.8b. Essa foi a solução encontrada após 20 horas de execução do algoritmo, tempo pré-estabelecido como um dos critérios de parada.

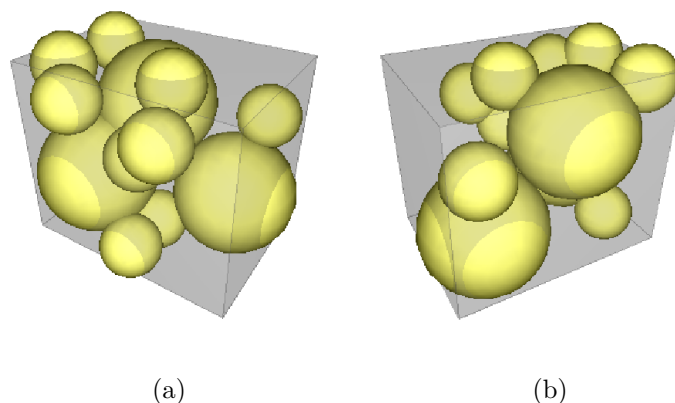


Figura 5.8: Solução encontrada pelo solver XPRESS para o recobrimento de um paralelepípedo de dimensões 14mm x 12mm x 10mm, com as restrições de ordenação das esferas.

	cov	=	54.45%
Para essa configuração das esferas, temos:	miscov	=	11.31%
	overlap	=	2.2%

O resultado ainda não é satisfatório, com relação ao percentual de recobrimento das esferas. Os quatro solvers utilizados não encontraram soluções boas sempre ou até mesmo não encontravam qualquer solução viável. A partir dessa observação, desenvolvemos uma heurística, a fim de obter bons recobrimentos em tempo razoável de execução.

## 5.1 Heurística

Da necessidade de obtenção de soluções que recobrissem o volume com um bom percentual de recobrimento em um tempo razoável, desenvolvemos uma heurística, na tentativa de cortar ao máximo a árvore de enumeração de soluções, com respeito às variáveis inteiras do problema.

Suponha que sejam conhecidos, a priori, a quantidade de esferas na solução e seus respectivos raios. A única tarefa seria encontrar a melhor posição para os centros das esferas, de acordo com as restrições do modelo. Inicialmente, os valores para os centros são gerados de forma totalmente aleatória, segundo uma distribuição normal. Depois, esfera por esfera, é realizada uma busca na direção de cada um dos eixos coordenados, a fim de obter uma nova posição para o seu centro que melhore o recobrimento, respeitando a região de segurança e o nível de interseção com outras esferas, de acordo com os valores dos parâmetros  $\alpha$ . Mas ambos esses procedimentos de construção da solução e de melhora são realizados a partir de um conjunto de esferas dado. A maneira usada para determinar a quantidade de esferas na solução e cada um de seus raios é descrita a seguir.

Como o objetivo do problema é encontrar um recobrimento utilizando o menor número possível de esferas, a idéia é começar com apenas uma esfera na solução e testar todos os raios. Estabelecemos um nível mínimo de recobrimento necessário, denotado  $minCov$ . Tomamos esse valor como 0,9. Caso os testes não alcancem esse percentual de recobrimento, adicionamos mais uma esfera na solução e testamos todas as combinações de raios com essa nova quantidade de esferas. O algoritmo termina quando o nível mínimo de recobrimento é alcançado e a solução é viável. Porém, testar todas as combinações possíveis de raios, para as diversas diferentes quantidades de esferas, é impraticável, pois forçaria a execução do algoritmo por um tempo muito longo. Alguns cortes na enumeração foram realizados.

O primeiro conjunto de cortes é relacionado à região de segurança e leva em consideração o diâmetro do conjunto  $T$ . Não serão utilizadas esferas de diâmetros



maiores que  $\text{diam}(T) + 2\epsilon$ . Se algum dos valores possíveis para os raios for superior a  $\text{diam}(T)/2 + \epsilon$ , esses raios não serão utilizados. Logo, qualquer combinação que envolva esferas com esses raios será cortada da árvore de enumeração.

Seja  $S_{rmax}$  a esfera de maior raio sendo utilizada e seja  $rmax$  esse maior raio. A quantidade inicial de esferas, denotada por  $N_0$  é, na verdade, calculada com relação ao volume da região  $T$  a ser recoberta:

$$N_0 = \text{minCov} * \frac{\text{Vol}(T)}{\text{Vol}(S_{rmax})} .$$

Ou seja, não será possível recobrir a região  $T$  se, mesmo somente com esferas do maior raio, utilizarmos uma quantidade menor que  $N_0$ , já que a soma de seus volumes é inferior a  $\text{mincov} * \text{Vol}(T)$ , que representa o menor volume recoberto aceitável.

Por fim, cada vez que uma nova esfera é adicionada na solução, a primeira combinação a ser otimizada é aquela onde todas as esferas possuem raio  $rmax$ . Tomemos o caso em que esse conjunto de esferas consegue atingir o nível mínimo de recobrimento  $\text{minCov}$ , mas não são viáveis por não atenderem aos valores máximos de interseção permitidos, definidos pelos parâmetros  $\alpha$  do modelo. Quando isso acontece, sabemos que não adianta utilizar mais que essa quantidade de esferas com raio  $rmax$ . Ou seja, ao aumentarmos ainda mais a quantidade de esferas na solução, não permitiremos que sejam utilizadas mais esferas de raio  $rmax$  do que a quantidade presente na situação descrita.

Dessa forma, a combinatória é bastante reduzida e permite o uso da heurística por apresentar resultados bons em tempos de execução aceitáveis, como pode ser comprovado a seguir.

## Resultados

Aplicamos a heurística na mesma instância que utilizamos com os solvers comerciais. O resultado pode ser visto nas figuras 5.9a, 5.9b e 5.9c.

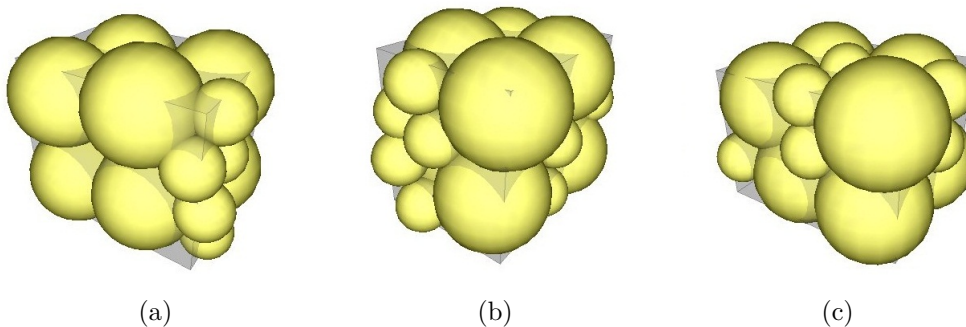


Figura 5.9: Solução encontrada pela heurística que desenvolvemos para o recobrimento de um paralelepípedo de dimensões 14x12x10mm.

Essa solução apresenta um bom nível de recobrimento, respeitando os valores permitidos para interseção entre esferas e para a região de segurança.

	cov	=	95.19%
Para essa solução, temos:	miscov	=	34.55%
	overlap	=	13.74%

Aproveitamos a heurística para realizar testes de recobrimentos em elipsóides. Utilizamos um elipsóide de raios 10mm, 5.2mm e 5.2mm. O resultado pode ser visto na figuras 5.10a e 5.10b. Foram utilizadas uma esfera de raio 7mm e uma de 4mm.

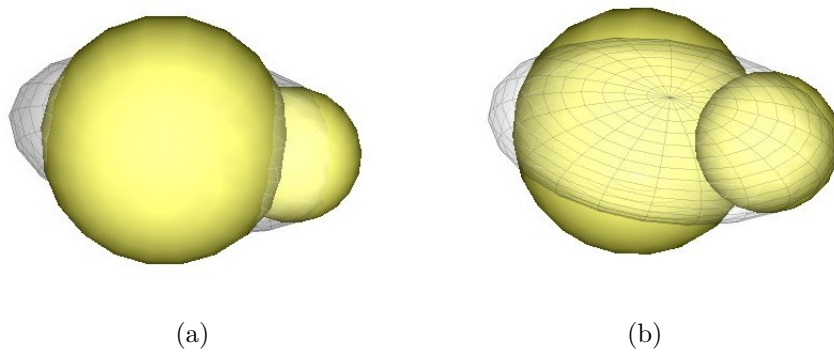


Figura 5.10: Resultado obtido com a heurística para um elipsóide de raios 10mm, 5.2mm e 5.2mm.

	cov	=	95,44%
Para essa configuração das esferas, temos:	miscov	=	85,47%
	overlap	=	6,34%

Ao utilizarmos três esferas, temos o resultado mostrado na figura 5.11a. A solução agora apresenta 2 esferas de raio 4mm e uma de 7mm.

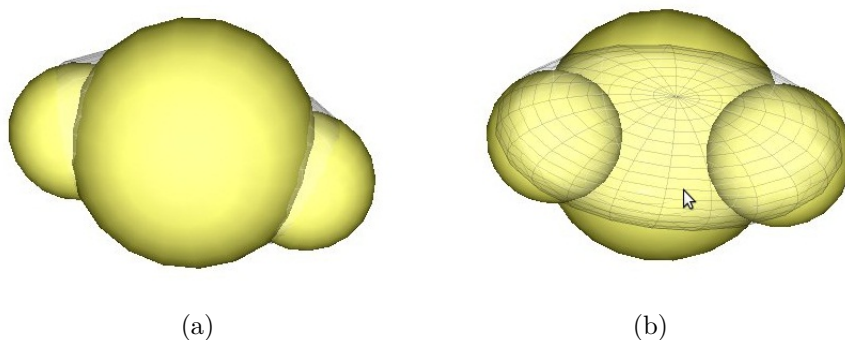


Figura 5.11: Resultado obtido com heurística para um elipsóide de raios 10mm, 5.2mm e 5.2mm, com o acréscimo de uma esfera.

Para essa configuração das esferas, temos:

$$\begin{aligned} \text{cov} &= 98,67\% \\ \text{miscov} &= 86,89\% \\ \text{overlap} &= 14,88\% \end{aligned}$$

Como pode ser observado, a solução com 3 esferas apresenta um nível de recobrimento maior. Porém, essa última solução também tem um valor maior de ‘miscov’ e ‘overlap’. Além de, obviamente, apresentar um número maior de esferas na solução. Esse é um exemplo da dificuldade encontrada ao tentar definir critérios de comparação e qualificação de soluções.

# Capítulo 6

## Conclusões

Esse trabalho teve como objetivo o estudo do problema de recobrimento associado ao problema de planejamento do tratamento conhecido como *Gamma Knife*. Esse problema é modelado com um problema de programação matemática não-linear misto (MINLP). Essa classe de problemas apresenta, em geral, duas principais dificuldades: a integralidade de um subconjunto das variáveis e a presença de função objetivo e restrições não-convexas e, portanto, não-lineares.

Muitos *solvers* exatos para MINLP, atualmente, utilizam técnicas de *branch-and-bound* para calcular um limite do valor ótimo através de relaxação linear do problema original. Alguns casos especiais de MINLP admitem *solvers* mais eficientes. Exemplos são os problemas lineares mistos (MILP), onde a função objetivo é linear e as restrições são funções afim, e MINLPs convexos, onde a relaxação contínua é um problema não-linear convexo. O problema de recobrimento estudado, porém, é um MINLP que não pode ser classificado em nenhum desses casos especiais. Encontrar uma relaxação e um limite inferior para o ótimo global desse problema não é uma tarefa fácil, já que, mesmo relaxando as restrições de integralidade, ainda obtemos um problema não-linear não-convexo.

Observamos que os resultados obtidos pelo XPRESS, pelo CPLEX, pelo GURROBI e pelo COUENNE não são satisfatórios. Apresentaram grande dificuldade em encontrar uma solução viável ao problema. Mesmo as soluções encontradas apresentam muito tempo de processamento, muitas esferas na cobertura e baixos percentuais de recobrimento.

Pela dificuldade dos solvers existentes na obtenção de boas soluções, por não incorporarem características do problema, faz-se necessário o desenvolvimento de heurísticas capazes de encontrar, em tempo razoável, soluções viáveis aos problemas, com bons percentuais de recobrimento. A idéia mais natural é a construção de um método de *branch-and-bound* específico ao problema, com bons critérios de corte, de forma a reduzir de forma considerável a árvore de enumeração.

## 6.1 Trabalhos Futuros

Como continuação ao estudo do problema de recobrimento associado ao tratamento por *Gamma Knife*, pretendemos explorar e incluir nos modelos variáveis que representem as dosagens ou duração de aplicações dos *shots*. Essas idéias já aparecem bastante na literatura, mas, em sua maioria, como problemas de empacotamento.

Outra proposta de trabalho futuro e, talvez, a mais interessante, é considerar regiões a serem cobertas de formatos mais irregulares, aproximando-se mais do problema de planejamento do tratamento por *Gamma Knife*. Neste trabalho, utilizamos apenas paralelepípedos e elipsóides como representação do tumor. Em pesquisas futuras, procuraremos utilizar imagens reais provenientes de tomografias computadorizadas e ressonâncias magnéticas. Essas imagens são de natureza discreta e podem ser utilizadas sem a necessidade de modificações nos modelos propostos.

Pretendemos, ainda, estudar o problema de detecção de cavidades em um sistema de esferas com sobreposição [67] e uma forma de calcular o volume recoberto de forma exata [68].

# Referências Bibliográficas

- [1] GANZ, J. C. “Gamma knife surgery”, *Springer-Verlag Wien, Austria*, 1997.
- [2] JITPRAPAIKULSARN, S. “An Optimization-based treatment planner for gamma knife radiosurgery”, *Ph.D. Thesis, Case Western Reserve University*, 2005.
- [3] LIM, J. “Optimization in radiation treatment planning”, *Ph.D. Thesis, University of Wisconsin-Madison*, 2002.
- [4] FERRIS, M., SHEPARD, D. “Optimization of gamma knife radiosurgery”, *Discrete Mathematical Problems with Medical Applications, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, v. 55, pp. 27–44, 2000.
- [5] FERRIS, M., VOELKER, M. “Neuro-dynamic programming for radiation treatment planning”, *Annals of Operations Research*, v. 119, pp. 247–260, 2003.
- [6] FERRIS, M., LIM, J., SHEPARD, D. “An optimization approach for the radiosurgery treatment planning”, *SIAM Journal on Optimization*, v. 13, n. 3, pp. 921–937, 2003.
- [7] LIBERTI, L., MACULAN, N., ZHANG, Y. “Optimal configuration of gamma ray machine radiosurgery units: the sphere covering subproblem”, *Optimization Letters*, v. 3, n. 1, pp. 109–121, 2009.
- [8] DO NASCIMENTO, R. Q., MACAMBIRA, A. F. U. S., DOS ANJOS FORMIGA CABRAL, L., et al. “The Ellipsoid Covering problem: an Continuous Approach”, *2nd International Conference on Engineering Optimization*, 2010.
- [9] XAVIER, A. E., DE OLIVEIRA, A. A. F. “Optimal Covering of Plane Domains by Circles Via Hyperbolic Smoothing”, *Journal of Global Optimization*, v. 31, pp. 493–504, 2005.

- [10] XAVIER, A. E. “Penalização Hiperbólica: Um Novo Método para Resolução de Problemas de Otimização”, *M.Sc. Thesis – COPPE – UFRJ – Rio de Janeiro*, 1982.
- [11] ANDONOV, R., POIRRIEZ, V., RAJOPADHYE, S. “Unbounded Knapsack Problem : dynamic programming revisited”, *European Journal of Operational Research*, v. 123, n. 2, pp. 168–181, 2000.
- [12] WILLIAMS, R. “Circle Packings, Plane Tessellations, and Networks”, *The Geometrical Foundation of Natural Structure: A Source Book of Design*. New York: Dover, pp. 34–47, 1979.
- [13] TÓTH, G. F. “New results in the theory of packing and covering”, in: *P.M. Gruber, J.M. Wills (Eds.), Convexity and Its Applications*, pp. 318–359, 1983.
- [14] TÓTH, G. F. “Densest packings of typical convex sets are not lattice like”, *Discrete and Computational Geometry*, v. 14, pp. 1–8, 1995.
- [15] HALES, T. C. “A proof of the Kepler conjecture”, *Annals of Mathematics. Second Series*, v. 162, n. 3, pp. 1065–1185, 2005.
- [16] SPECHT, E. <http://www.packomania.com/>.
- [17] SUTOU, A., DAI, Y. “Global Optimization Approach to Unequal Sphere Packing Problems in 3D”, *Journal of Optimization Theory and Applications*, v. 114, n. 3, pp. 671–694, 2002.
- [18] WANG, J. “Packing of Unequal Spheres and Automated Radiosurgical Treatment Planning”, *Journal of Combinatorial Optimization*, v. 3, pp. 453–463, 1999.
- [19] MACULAN, N., MICHELON, P., SMITH, J. M. “Bounds on the kissing numbers in  $R^n$ : mathematical programming formulations”, *Technical Report, Industrial Engineering and Operations Research Dept., University of Massachusetts, Amherst, USA*, 1995.
- [20] LEECH, J. “The problem of the thirteen spheres”, *Math. Gaz.*, v. 40, pp. 22–23, 1956.
- [21] MUSIN, O. R. “The problem of the twenty-five spheres”, *Russian Mathematical Surveys*, v. 58, pp. 794–795, 2003.

- [22] PFENDER, F., ZIEGLER, G. M. “Kissing numbers, sphere packings, and some unexpected proofs”, *Notices of the American Mathematical Society*, pp. 873–883, 2004.
- [23] LEVENSHTAIN, V. I. “On bounds for packings in n-dimensional Euclidean space”, *Doklady Akademii Nauk SSSR*, v. 245, n. 6, pp. 1299–1303, 1979.
- [24] ODLYZKO, A. M., SLOANE, N. J. A. “New bounds on the number of unit spheres that can touch a unit sphere in n dimensions”, *Journal of Combinatorial Theory, Series A*, v. 26, n. 2, pp. 210–214, 1979.
- [25] KUCHERENKO, S., BELOTTI, P., LIBERTI, L., et al. “New formulations for the Kissing Number Problem”, *Discrete Applied Mathematics*, v. 155, pp. 1837–1841, 2007.
- [26] FLOUDAS, C. A. “Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications”, *Oxford University Press*, 1995.
- [27] NEMHAUSER, G. L., WOLSEY, L. A. “Integer and Combinatorial Optimization”, *J. Wiley, New York*, 1988.
- [28] MURTY, K. G., KABADI, S. N. “Some NP-complete problems in quadratic and nonlinear programming”, *Mathematical Programming*, v. 39, pp. 117–129, 1987.
- [29] PARDALOS, P. M., SCHNITGER, G. “Checking local optimality in constrained quadratic programming is NP-hard”, *Operational Research Letter*, v. 7, 1988.
- [30] CORP., G. D. “SBB”, <http://www.gams.com/dd/docs/solvers/sbb.pdf>.
- [31] LEYFFER, S. “User manual for MINLP\_BB”, *Technical report, University of Dundee, UK*, 1999.
- [32] BONAMI, P., BIEGLER, L., CONN, A., et al. “An algorithmic framework for convex mixed integer nonlinear programs”, *Discrete Optimization*, v. 5, pp. 186–204, 2008.
- [33] ABHISHEK, K., LEYFFER, S., LINDEROTH, J. T. “FILMINT: An outer-approximation-based solver for nonlinear mixed integer programs”, *Preprint ANL/MCS-P1374-0906*, 2006.
- [34] BELOTTI, P., LEE, J., LIBERTI, L., et al. “Branching and Bounds Tightening Techniques for Non-Convex MINLP”, *IBM Research Report*, 2008.



- [35] GOLDBERG, D. E. “Genetic Algorithms in Search, Optimization and Machine Learning”, 1989.
- [36] FOGEL, D. B. “Evolutionary Computation: Towards a New Philosophy of Machine Intelligence”, *New York: IEEE Press*, p. 140, 2000.
- [37] KIRKPATRICK, S., GELATT, C. D., VECCHI, M. P. “Optimization by Simulated Annealing”, *Science*, v. 220, pp. 671–680, 1983.
- [38] METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., et al. “Equations of State Calculations by Fast Computing Machines”, *The Journal of Chemical Physics*, v. 21, n. 6, pp. 1087–1092, 1953.
- [39] MIKI, M., HIROYASU, T., WAKO, J. “Adaptive temperature schedule determined by genetic algorithm for parallel simulated annealing”, *2003 Congress on Evolutionary Computation*, 2003.
- [40] GRANVILLE, V., KRIVANEK, M., RASSON, J. P. “Simulated annealing: A proof of convergence”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 16, n. 6, pp. 652–656, 1994.
- [41] INGBER, L. “Simulated annealing: Practice versus theory”, *Mathematical and Computer Modelling*, v. 18, n. 11, pp. 29–57, 1993.
- [42] GLOVER, F. “Tabu Search - Part I”, *ORSA Journal on Computing*, v. 1, n. 3, pp. 190–206, 1989.
- [43] GLOVER, F. “Tabu Search - Part II”, *ORSA Journal on Computing*, v. 2, n. 1, pp. 4–32, 1990.
- [44] GLOVER, F., LAGUNA, M. “Tabu Search”, *Kluwer Academic Publishers*, 1997.
- [45] HEDAR, A. R., FUKUSHIMA, M. “Tabu search directed by direct search methods for nonlinear global optimization”, *European Journal of Operational Research*, v. 170, n. 2, pp. 329–349, 2006.
- [46] A.SCHAEFER. “Tabu search techniques for large high-school timetabling problems”, *Proceedings of the 13th National Conference on Artificial Intelligence*, v. 13, pp. 363–368, 1996.
- [47] DAMMEYER, F., VOB, S. “Dynamic tabu list management using the reverse elimination method”, *Annals of Operations Research*, v. 41, n. 2, pp. 29–46, 1993.

- [48] HANSEN, P., MLADENOVIC, N. “Variable neighborhood search”, *Computers & Operations Research*, v. 24, n. 11, pp. 1097–1100, 1997.
- [49] HANSEN, P., MLADENOVIC, N. “Variable neighborhood search: principles and applications”, *European Journal of Operational Research*, v. 130, pp. 449–467, 2001.
- [50] HANSEN, P., MLADENOVIC, N. “Developments of variable neighborhood search”, *Essays and Surveys in Metaheuristics*, pp. 415–439, 2001.
- [51] DRAZIC, M., LAVOR, C., MACULAN, N., et al. “A continuous variable neighborhood search heuristic for finding the three-dimensional structure of a molecule”, *European Journal of Operational Research*, v. 185, n. 3, pp. 1265–1273, 2008.
- [52] MLADENOVIC, N., DRAZIC, M., KOVACEVIC-VUJCIC, V., et al. “General variable neighborhood search for the continuous optimization”, *European Journal of Operational Research*, v. 191, pp. 753–770, 2008.
- [53] FEO, T. A., RESENDE, M. G. “Greedy randomized adaptive search procedures”, *Journal of Global Optimization*, v. 6, pp. 109–133, 1995.
- [54] PRAIS, M., RIBEIRO, C. “Parameter variation in GRASP procedures”, *Investigación Operativa*, v. 9, pp. 1–20, 2000.
- [55] RESENDE, M. G. C., RIBEIRO, C. C. “GRASP”, *AT&T Labs Research Technical Report*, 2008.
- [56] LIBERTI, L. “Reformulation techniques in mathematical programming”, *Thèse d’Habilitation à Diriger des Recherches*, 2007.
- [57] LIBERTI, L. “Reformulations in mathematical programming: Definitions”, *Proceedings of the 7th Cologne-Twente Workshop on Graphs and Combinatorial Optimization*, 2008.
- [58] LIBERTI, L. “Writing global optimization software”, *Global Optimization: from Theory to Implementation*, pp. 211–262, 2008.
- [59] CARRIZOSA, E., HANSEN, P., MESSINE, F. “Improving interval analysis bounds by translations”, *Journal of Global Optimization*, v. 29, n. 2, pp. 157–172, 2004.
- [60] MESSINE, F. “Deterministic global optimization using interval constraint propagation techniques”, *RAIRO-RO*, v. 38, n. 4, pp. 277–294, 2004.

- [61] WEI, Z., LONG, Y., YE, Y. “A Sphere Packing Model for the Optimal Treatment Plan”, .
- [62] SPIRAKIS, P. G. “The volume of the union of many spheres and point inclusion problems”, *Lecture Notes in Computer Science*, v. 182, pp. 328–338, 1985.
- [63] <http://www.coin-or.org/Couenne/>, .
- [64] [www.gurobi.com](http://www.gurobi.com), .
- [65] <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>, .
- [66] <http://www.fico.com/en/Products/DMTools/Pages/FICO-Xpress-Optimization-Suite.aspx>, .
- [67] BUSA, J., HAYRYAN, E., HAYRYAN, S., et al. “Detecting cavities in a system of overlapping spheres using enveloping triangulation”, *Proceedings of Algoritmy 2009*, pp. 392–401, 2009.
- [68] AVIS, D., BHATTACHARYA, B. K., IMAI, H. “Computing the volume of the union of spheres”, *The Visual Computer*, v. 3, pp. 323–328, 1988.