

**SISTEMAS DE EQUAÇÕES NÃO-LINEARES E
PROBLEMAS DE EMPACOTAMENTO**

Jan Marcel Paiva Gentil

Orientador: Prof. Ernesto G. Birgin

Fomento: FAPESP (proc. 06/57633-1)

Departamento de Ciência da Computação

Instituto de Matemática e Estatística

Universidade de São Paulo

Fevereiro de 2008

Sumário

I	Parte técnica	1
1	Introdução	2
1.1	Abordagens anteriores	2
1.2	Motivações e objetivos	3
2	Metodologia	5
2.1	Modelo não-linear	5
2.1.1	Objeto circular	5
2.1.2	Objeto quadrado	6
2.1.3	Objeto em <i>strip</i>	6
2.1.4	Objeto retangular	6
2.2	Abordagem proposta	6
2.3	O algoritmo	7
2.4	Detecção de contatos	8
2.5	Formulação do sistema	9
2.6	Resolução do sistema	10
2.6.1	Método de Newton-Raphson	11
2.6.2	Decomposição QR	11
2.6.3	Decomposição de Cholesky	12
2.7	Pré e pós-processamento	13
2.7.1	Detecção de itens soltos	13
2.7.2	Iteração de item semi-fixado	15
2.7.3	Reescalamento da solução	16
3	Resultados	18
3.1	Experimentos numéricos	18
3.2	Análise e comentários	18
4	Conclusão	25
A	Imagens produzidas	26
A.1	Objeto circular	26
A.2	Objeto quadrado	29
A.3	Objeto em <i>strip</i>	32
A.4	Objeto retangular	36

A.4.1	Minimização de área	36
A.4.2	Minimização de perímetro	38
II	Parte subjetiva	42
4.5	A Iniciação Científica	43
4.6	Interação com o orientador	43
4.7	Aspectos do curso mais relevantes	44
4.8	Passos futuros de aprofundamento na área	44

Resumo

A classe de problemas em estudo é a de empacotamento de itens circulares, que consiste em encontrar uma disposição de um número fixo deles que minimize as dimensões do objeto que os contém. Neste trabalho, foram considerados objetos em forma de círculo, quadrado, *strip* e retângulo. Por objetivo o projeto teve o de encontrar, a partir de resultados obtidos por métodos de Otimização Contínua empregados em trabalhos anteriores, soluções mais precisas para tais problemas. Para isso, é proposta uma estratégia baseada na formulação de sistemas de equações não-lineares e a sua solução a partir do método de Newton-Raphson, cujos resultados mostraram-se bastante satisfatórios.

Palavras-chave: Empacotamento, sistemas de equações não-lineares, programação não-linear.

Parte I

Parte técnica

Capítulo 1

Introdução

Problemas de empacotamento ocorrem naturalmente em diversas situações da vida cotidiana, da disposição de produtos em uma embalagem ao carregamento de caixas em um caminhão. Por esse motivo, algoritmos capazes de resolvê-los eficientemente há muito despertam grande interesse não só matemático, mas também econômico.

Já é sabido, porém, que a busca por uma solução ótima global para um empacotamento é um problema *NP*-difícil, o que a torna muito custosa em termos de tempo de processamento. Sendo assim, comumente são empregados métodos heurísticos, que mesmo sendo em sua maioria razoavelmente rápidos, não garantem convergir a minimizadores globais. Dentre eles, interessam a este trabalho os que utilizam modelos não-lineares do problema, que podem ser então resolvidos por algoritmos de programação não-linear.

Serão discutidas a seguir algumas dessas estratégias de resolução publicadas na literatura, introduzindo-se daí a motivação para o desenvolvimento do trabalho. No capítulo seguinte, o algoritmo desenvolvido será exibido e seu funcionamento, bem como a teoria sobre a qual se fundamenta, serão detalhados. Por fim, os resultados numéricos obtidos serão sintetizados e comentados.

1.1 Abordagens anteriores

Em [18], trata-se do problema de empacotar itens circulares idênticos em um quadrado. Um modelo *max-min* para maximizar a menor distância entre os centros dos itens é apresentado com o objetivo de maximizar o seu diâmetro. Uma reformulação não-linear clássica do problema *max-min* é resolvida usando-se MINOS [21] e a linguagem de modelagem GAMS [11]. Emprega-se uma estratégia *multi-start* para aumentar a probabilidade de se encontrarem soluções globais. Problemas com até 30 itens são resolvidos, alcançando-se as melhores soluções já divulgadas na literatura e dando origem a algumas novas configurações.

Uma outra formulação em programação não-linear do mesmo problema é também introduzida em [22]. Nela, busca-se minimizar a função energia $\sum_{i \neq j} (\lambda/d_{ij}^2)^m$, em que d_{ij} representa a distância euclidiana entre os centros dos itens i e j , λ é um fator de escala e m é um inteiro positivo, sob a restrição de dispôr os itens dentro do objeto. Uma reformulação sem restrições deste problema é resolvida usando-se um algoritmo de busca linear que emprega direções do tipo gradiente no início e de Newton próximo à solução. Utiliza-se então a resposta obtida pelo algoritmo de otimização como ponto de partida para se resolver um sistema de equações não-lineares com o objetivo de melhorar a precisão da solução. Problemas com até 50 itens são tratados, encontrando-se algumas respostas

alternativas e aprimorando alguns resultados obtidos em [14,16].

Em [17], estuda-se o mesmo problema, modelado como um de otimização quadrática. Duas propriedades satisfeitas por pelo menos uma solução ótima são apresentadas e um eficiente algoritmo *branch-and-bound* é desenvolvido. Ele é usado para provar a ε -otimalidade de soluções para até 35 itens, 38 e 39 itens. Além disso, novas soluções para 32 e 37 itens são encontradas. Finalmente, os autores de [17] ressaltam que, no que diz respeito ao problema de maximizar o diâmetro dos itens (em vez de minimizar o tamanho do objeto), enquanto a maioria das abordagens são capazes de obter um limitante inferior, a deles provê um limitante superior da solução.

O problema de empacotar círculos idênticos em um círculo é discutido em [20]. A formulação não-linear é uma das mais naturais e maximiza o raio dos itens enquanto impõe que caibam dentro do objeto sem que se sobreponham. Duas versões equivalentes são apresentadas, uma usando coordenadas cartesianas e outra, polares. É testado um método heurístico chamado *Reformulation Descent*, que iterativamente alterna de um modelo para outro. Os autores afirmam que essas transições reduzem a probabilidade de o método parar em pontos estacionários indesejáveis. Experimentos numéricos mostram que ele é muito mais rápido que os clássicos de programação não-linear e que, para instâncias de até 100 itens, a melhor solução é encontrada em 40% dos casos, enquanto no resto deles o erro nunca excede 1%.

Já o problema de empacotar círculos de tamanhos distintos em *strips* é considerado em [25]. Um típico modelo não-linear é apresentado e profundamente estudado. Conforme assinalado pelos autores, algumas peculiaridades do modelo permitem concluir que soluções ótimas são obtidas em pontos extremos da região viável. Uma abordagem inteligente (baseada em multiplicadores de Lagrange) de passar de um minimizador local para outro melhor trocando-se de posição dois círculos distintos é elaborada. Os experimentos numéricos em [25] mostram que essa técnica, que pode ser encarada como uma forma de se evitarem minimizadores locais, juntamente com um eficiente método local [24] e uma estratégia *multi-start*, conduzem a ótimas soluções. Uma comparação com um algoritmo *branch-and-bound* usando instâncias com até 35 itens exhibe as vantagens da abordagem apresentada.

Modelos não-lineares têm sido também empregados com sucesso em problemas de, dado um número infinito de itens idênticos e um objeto, todos de dimensão fixada, empacotar o maior número possível de itens no objeto. Em [9], os autores lidam com itens circulares dentro de objetos retangulares e circulares. Em [7,8], problemas de empacotar retângulos ortogonais e livremente rotacionados no interior de regiões convexas arbitrárias são considerados, respectivamente. O Método de Sentinelas apresentado em [7,19] pode também ser utilizado para empacotar polígonos arbitrários distintos (com ângulos internos não menores que $\pi/2$) em regiões convexas arbitrárias. Em todos os casos, os modelos não-lineares são resolvidos usando-se ALGENCAN [2–4,6], um método de Lagrangeanos Aumentados para minimização de funções suaves com restrições gerais (gratuitamente disponível na página do Projeto TANGO [5]).

1.2 Motivações e objetivos

Todas as estratégias baseadas em modelos não-lineares já conhecidas têm um ponto em comum: constituem métodos iterativos, de convergência *linear*, que geram uma seqüência *infinita* de pontos, a cujo conjunto de pontos de acumulação, no melhor dos casos, pertence a solução buscada. Assim, na prática, faz-se necessária a utilização de um critério de parada capaz de medir a proximidade à resposta desejada. Ademais, os algoritmos empregados garantem convergir a minimizadores locais,

mas não a minimizadores globais.

Dessa forma, este projeto buscou desenvolver um método de convergência *quadrática* que aprimorasse a acurácia dos resultados obtidos por tais métodos, igualando as soluções ótimas já divulgadas [23] e provendo ainda respostas precisas para casos não reportados na literatura.

Capítulo 2

Metodologia

Neste capítulo, o trabalho desenvolvido será descrito em detalhes. Antes disso, é conveniente esclarecer alguns conceitos:

Item Cada um dos artigos empacotados, fixados como sendo círculos de raio unitário.

Objeto Contêiner de itens, cujo formatos estudados são o circular, o quadrado, o retangular e o *strip*, que consiste em um retângulo com uma de duas dimensões previamente fixada.

2.1 Modelo não-linear

Neste trabalho, a variante estudada dentre todas as apresentadas para problemas de empacotamento será a que busca minimizar as dimensões de um objeto que comporte em seu interior um dado número de itens circulares idênticos, sem que haja sobreposições. Dessa formulação obtém-se naturalmente o seguinte modelo não-linear:

$$\begin{array}{ll} \text{Minimizar} & \text{as dimensões do objeto} \\ \text{sujeito a} & \text{comportar os itens sem sobreposições} \end{array} \quad (2.1)$$

Sejam $c_i = (c_i^x, c_i^y) \in \mathbb{R}^2$ e $r_i \in \mathbb{R}$ o centro e o raio, respectivamente, do i -ésimo item considerado, para $i = 1, \dots, N$. Denotemos também por $d(\cdot, \cdot)$ a distância euclidiana entre dois pontos de \mathbb{R}^2 . As restrições que concernem a proibição de sobreposições entre os itens empacotados dependem apenas de sua forma e, por estar essa definida como circular ao longo de todo o decorrer do projeto, traduzem-se no seguinte conjunto de inequações:

$$d(c_i, c_j) \geq r_i + r_j, \quad \forall i \neq j \quad (2.2)$$

O restante das restrições associadas a essa modelagem cuidam da não-violação dos limites do objeto e, portanto, dependem diretamente de sua forma. Do mesmo modo, a função objetivo informalmente definida em (2.1) varia conforme o tipo de objeto. Vejamos que formas ambas podem assumir para aqueles considerados neste trabalho.

2.1.1 Objeto circular

Minimizar as dimensões de um círculo equivale a minimizar o seu raio R . Além disso, todos os itens que estiverem completamente contidos em seu interior satisfazem à inequação $R \geq d(c_i, C) +$

$r_i, \forall i$, em que C denota o centro do objeto. Supondo, sem perda de generalidade, que ele está centrado na origem (o que pode ser conseguido por uma simples operação de translação), é possível expressar (2.1) como:

$$\begin{aligned} \text{Min.} \quad & R \\ \text{s. a} \quad & (c_i^x)^2 + (c_i^y)^2 \leq (R - r_i)^2, \forall i \\ & \text{restrições de sobreposição (2.2)} \end{aligned} \tag{2.3}$$

2.1.2 Objeto quadrado

Analogamente ao caso circular, a função objetivo de interesse é a que busca minimizar o lado L do quadrado. Supondo agora que a origem do sistema cartesiano é dada pelo vértice inferior esquerdo do quadrado, o modelo não-linear fica:

$$\begin{aligned} \text{Min.} \quad & L \\ \text{s. a} \quad & r_i \leq c_i^x \leq L - r_i, \forall i \\ & r_i \leq c_i^y \leq L - r_i, \forall i \\ & \text{restrições de sobreposição (2.2)} \end{aligned} \tag{2.4}$$

2.1.3 Objeto em *strip*

Por ter uma de suas dimensões (digamos, L) fixada, o *strip* só permite a minimização da outra, designada por W . Mantendo a suposição sobre a origem do sistema de coordenadas, vem:

$$\begin{aligned} \text{Min.} \quad & W \\ \text{s. a} \quad & r_i \leq c_i^x \leq L - r_i, \forall i \\ & r_i \leq c_i^y \leq W - r_i, \forall i \\ & \text{restrições de sobreposição (2.2)} \end{aligned} \tag{2.6}$$

2.1.4 Objeto retangular

A minimização das dimensões de um retângulo, por sua vez, pode ser encarada de duas formas distintas: (i) minimização da soma de sua base e de sua altura (também chamada de semi-perímetro) ou (ii) minimização de sua área. É interessante observar que essas duas formulações não são de forma alguma equivalentes, merecendo ambas ser objeto do estudo desenvolvido. Têm-se, portanto, dois modelos possíveis:

$$\begin{aligned} \text{Min.} \quad & L + W \text{ ou } L \times W \\ \text{s. a} \quad & r_i \leq c_i^x \leq L - r_i, \forall i \\ & r_i \leq c_i^y \leq W - r_i, \forall i \\ & \text{restrições de sobreposição (2.2)} \end{aligned} \tag{2.8}$$

2.2 Abordagem proposta

Não é difícil compreender que, em uma configuração ótima, muitos dos itens envolvidos são postos em contato entre si ou com as bordas do objeto (Figura 2.1), o que torna ativas muitas dessas

restrições. Assim, se conhecidas de antemão essas que acabariam por ser satisfeitas com igualdade, é então possível escrever um sistema de equações não-lineares cuja solução constitui uma resposta para o problema de empacotamento estudado. Dessa forma, pode-se obter uma solução de precisão ainda maior que a da fornecida pelos métodos de otimização.

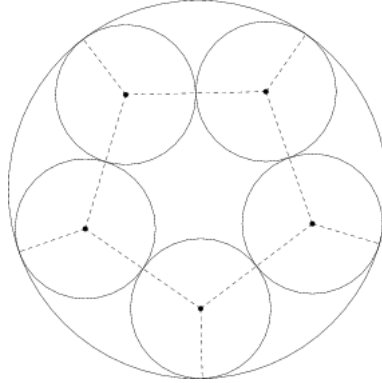


Figura 2.1: Configuração ótima para 5 itens

Cabe também mencionar que o número de restrições ativas na solução (que nada mais é que o número de contatos entre os itens somado ao número de itens que tocam as bordas do objeto) pertence a $O(N)$, enquanto o número total de restrições do modelo de programação não-linear é $N(N - 1)/2 + N$.

Para que a idéia apresentada possa ser viabilizada, é preciso que se conheçam as restrições ativas na configuração ótima buscada. Isso pode ser conseguido por meio da análise das soluções dadas por métodos de otimização não-linear, em especial pelo proposto em [10].

Nesse artigo, apresentam-se modelos não-lineares duas vezes diferenciáveis para empacotamento de itens circulares em diversos tipos de objetos, que então são resolvidos pelo método ALGENCAN, um moderno algoritmo de Lagrangeanos Aumentados para minimização suave e restrita ([2–4, 6]). As soluções obtidas se aproximam das ótimas publicadas em [23] com precisão de 10^{-6} , que, apesar de aquém do desejado pelos autores do trabalho, é o bastante para que se estime a disposição dos itens no empacotamento ótimo.

2.3 O algoritmo

Dispondo então de um método capaz de produzir razoáveis aproximações da solução do problema de empacotamento de interesse, é possível, a partir de sua análise, a detecção das restrições ativas e a formulação de um sistema de equações não-lineares satisfazível pela configuração ótima buscada. A estrutura geral do método produzido é a seguinte:

Algoritmo 1 Algoritmo em pseudo-código

Entrada: N := número de itens no empacotamento

Entrada: R := raio dos itens no empacotamento

```
1: while o tempo já consumido for menor que  $T$  do
2:   Execute ALGENCAN e obtenha uma solução  $x_0$  {aproximada}
3:   Realize o pré-processamento da solução
4:   Detecte os contatos estabelecidos na solução
5:   Formule o sistema de equações não-lineares correspondente
6:   repeat
7:     Aplique o método de Newton usando a solução  $x_{k-1}$ 
8:     Resolva o novo sistema linear e obtenha um novo  $x_k$ 
9:   until o número de iterações tenha ultrapassado  $K$ 
10:  Realize o pós-processamento da nova solução
11:  Compare com a melhor já obtida e atualize-a
12: end while
13: return  $x^*$  {melhor solução encontrada}
```

Por se tratar de um método iterativo, é necessário estabelecer limites de tempo à sua execução. Na implementação feita do algoritmo acima, o valor de K foi fixado em 1000 (número de iterações mais que suficiente para garantir a convergência total do método de Newton) e o de T variou conforme o tamanho do problema a ser resolvido, de apenas alguns segundos até muitas horas.

Não pertence ao escopo deste texto o detalhamento do passo (2.1), que está muito bem definido por seus autores em [10]. Em vez disso, será exibida nas próximas seções uma descrição minuciosa dos demais passos, iniciando-se pela detecção de contatos em uma dada configuração.

2.4 Detecção de contatos

Dois itens i e j estão em contato se a distância entre seus centros for igual à soma de seus raios, ou seja, $d(c_i, c_j) = r_i + r_j$. Supondo que todos os raios são idênticos, vale que $d(c_i, c_j) = 2r$. Todavia, por serem as operações de ponto flutuante suscetíveis a erros de truncamento e de representação na máquina, é prudente adotar um $\varepsilon_1 \in \mathbb{R}_+$ que represente o afastamento mínimo entre as bordas de dois itens necessário para que eles não sejam considerados em contato. Dessa forma, a verificação de contato entre itens do empacotamento fica:

$$\text{Item } i \text{ faz contato com item } j \iff \begin{array}{rcl} d(c_i, c_j) & \leq & 2r + \varepsilon_1 \\ d(c_i, c_j) - 2r & \leq & \varepsilon_1 \end{array} \quad (2.10)$$

Já a detecção de contato entre um item i e o objeto está intimamente relacionada à forma desse. De modo geral, isso acontece quando as restrições de não-violação nos modelos apresentados na Seção 2.1 forem satisfeitas com igualdade. Contudo, pelas razões já discutidas, é introduzido um valor $\varepsilon_2 \in \mathbb{R}_+$, que denota o afastamento mínimo entre as bordas de um item e do objeto necessário para que eles não sejam considerados em contato. Para os objetos estudados, decorrem as seguintes

verificações:

$$\text{Item } i \text{ faz contato com círculo} \iff R - ((c_i^x)^2 + (c_i^y)^2) - r \leq \varepsilon_2 \quad (2.11)$$

$$\text{Item } i \text{ faz contato com quadrado} \iff \begin{cases} c_i^x - r \leq \varepsilon_2 \text{ ou} \\ L - c_i^x - r \leq \varepsilon_2 \text{ ou} \\ c_i^y - r \leq \varepsilon_2 \text{ ou} \\ L - c_i^y - r \leq \varepsilon_2 \end{cases} \quad (2.12)$$

$$\text{Item } i \text{ faz contato com } \textit{strip} \iff \begin{cases} c_i^x - r \leq \varepsilon_2 \text{ ou} \\ L - c_i^x - r \leq \varepsilon_2 \text{ ou} \\ c_i^y - r \leq \varepsilon_2 \text{ ou} \\ W - c_i^y - r \leq \varepsilon_2 \end{cases} \quad (2.13)$$

$$\text{Item } i \text{ faz contato com retângulo} \iff \begin{cases} c_i^x - r \leq \varepsilon_2 \text{ ou} \\ L - c_i^x - r \leq \varepsilon_2 \text{ ou} \\ c_i^y - r \leq \varepsilon_2 \text{ ou} \\ W - c_i^y - r \leq \varepsilon_2 \end{cases} \quad (2.14)$$

A valoração de ε_1 e ε_2 é definida empiricamente, já que depende diretamente da máquina utilizada e também da precisão da resposta fornecida pelo método de programação não-linear. Na implementação desenvolvida, foram constatados bons resultados com a definição de ambos iguais a 10^{-4} , número que se aproxima da raiz quarta do *epsilon* da máquina disponível.

Uma vez construídos os conjuntos α de pares de itens em contato entre si e β de itens em contato com as bordas do objeto, é possível escrever um sistema de equações não-lineares a partir das restrições satisfeitas com a igualdade pela configuração ótima buscada x^* .

2.5 Formulação do sistema

Sejam n o número de variáveis e m o número de equações do sistema buscado $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$. O valor de n é precisamente $2N + 1$ para círculos, quadrados e *strips* e $2N + 2$ para retângulos (coordenadas x e y do centro dos N itens mais as dimensões variáveis do objeto), enquanto m é da ordem de $O(N)$ (número de contatos detectados no passo anterior).

A cada par $(i, j) \in \alpha$ corresponde a restrição ativa $d(c_i, c_j) = 2r$, ou, equivalentemente, $d(c_i, c_j)^2 = (2r)^2$. Isso se reflete no sistema buscado F pela seguinte equação:

$$\begin{aligned} f_\alpha(\cdot) &= d(c_i, c_j)^2 - (2r)^2 \\ &= (c_i^x - c_j^x)^2 + (c_i^y - c_j^y)^2 - 4r^2 \end{aligned} \quad (2.15)$$

Já a restrição ativa associada a cada elemento $i \in \beta$ varia de acordo com o objeto tratado e

pode assumir as seguintes formas:

$$\text{Em círculos, } f_{\beta}(\cdot) = (R - r)^2 - ((c_i^x)^2 + (c_i^y)^2) \quad (2.16)$$

$$\text{Em quadrados, } f_{\beta}(\cdot) = \begin{cases} c_i^x - r & \text{ou} \\ L - c_i^x - r & \text{ou} \\ c_i^y - r & \text{ou} \\ L - c_i^y - r \end{cases} \quad (2.17)$$

$$\text{Em strips, } f_{\beta}(\cdot) = \begin{cases} c_i^x - r & \text{ou} \\ L - c_i^x - r & \text{ou} \\ c_i^y - r & \text{ou} \\ W - c_i^y - r \end{cases} \quad (2.18)$$

$$\text{Em retângulos, } f_{\beta}(\cdot) = \begin{cases} c_i^x - r & \text{ou} \\ L - c_i^x - r & \text{ou} \\ c_i^y - r & \text{ou} \\ W - c_i^y - r \end{cases} \quad (2.19)$$

É importante chamar a atenção para o fato de que o sistema assim construído, ainda que sobredeterminado (em geral, $m \gg n$), não deixa de ser compatível, desde que, é claro, os valores de ε_1 e ε_2 tenham sido apropriadamente escolhidos. Isso porque muitas das equações que o compõem são redundantes, isto é, são meras combinações lineares das demais.

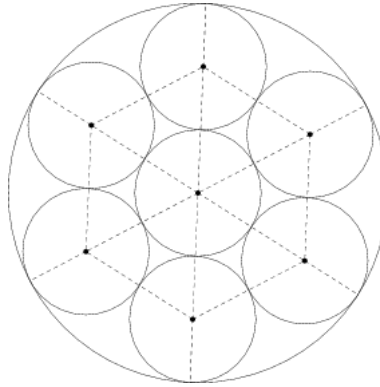


Figura 2.2: Configuração ótima para 7 itens, com $n = 15$ e $m = 18$

A Figura 2.2 constitui um bom exemplo, bastando observar que, mesmo se suprimidos alguns dos contatos (representados por linhas tracejadas) estabelecidos pelo item central, o posicionamento fixado para os demais não permite outra que não essa disposição.

2.6 Resolução do sistema

Estando a obtenção da solução vinculada à resolução do sistema obtido e visando conferir ao método convergência quadrática, a estratégia adotada para essa finalidade é o método de Newton-Raphson, exposto na próxima subseção.

2.6.1 Método de Newton-Raphson

O método de Newton é tido como um eficiente algoritmo para se determinarem aproximações para as raízes de uma função em \mathbb{R} . A idéia do método é a seguinte: partindo-se de uma estimativa inicial razoavelmente próxima à raiz verdadeira, a função é substituída por sua melhor aproximação local e então é computado o zero de tal aproximação. O valor obtido será uma estimativa ainda melhor para a raiz da função e o método pode daí ser iterado.

Seja $f: [a, b] \rightarrow \mathbb{R}$ uma função derivável definida no intervalo $[a, b]$ e com valores em \mathbb{R} . Suponha conhecida uma estimativa x_k de sua raiz. Sabendo-se que a derivada em um ponto nada mais é que a inclinação da reta tangente à função naquele ponto, é possível obter uma melhor aproximação x_{k+1} :

$$\begin{aligned} f'(x_k) &= \frac{f(x_k) - 0}{x_k - x_{k+1}} \\ &= \frac{0 - f(x_k)}{x_{k+1} - x_k} \\ x_{k+1} &= x_k - \frac{f(x_k)}{f'(x_k)} \end{aligned}$$

Iniciando-se o processo com um valor x_0 suficientemente próximo ao zero real da função, o método irá convergir para ele.

Esse procedimento pode ser facilmente aplicado à resolução de um sistema de m equações não-lineares, que é equivalente a encontrar a raiz de uma função $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$ continuamente diferenciável. Basta, na formulação dada acima, substituir o conceito de derivada pelo seu análogo multidimensional, a matriz jacobiana $J_F \in \mathbb{R}^{m \times n}$:

$$\begin{aligned} f(x_k) + f'(x_k)(x_{k+1} - x_k) &= 0 \\ F(x_k) + J_F(x_k)(x_{k+1} - x_k) &= 0 \\ J_F(x_k)(x_{k+1} - x_k) &= -F(x_k) \end{aligned} \tag{2.20}$$

Recai-se então em um sistema *linear* sobredeterminado da forma $Ax = b$, com $A \in \mathbb{R}^{m \times n}$, para a incógnita $x_{k+1} - x_k$.

Uma análise cuidadosa [12] permite também concluir que, para os casos em que a multiplicidade da raiz procurada é um, a seqüência $\{x_k\}$ gerada converge quadraticamente a ela (supondo x_0 suficientemente próximo).

São duas as formas de se computar a solução desse novo sistema:

- decomposição QR aplicada à matriz $A = J_F(x_k)$;
- decomposição de Cholesky aplicada às equações normais.

Ambas serão tratadas nas subseções seguintes.

2.6.2 Decomposição QR

Por não se ter qualquer informação quanto ao posto da matriz $J_F(x_k)$, é empregada a decomposição QR com *pivotação*, dada por:

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix} P^T, \tag{2.21}$$

sendo $Q \in \mathbb{R}^{m \times n}$ uma matriz ortogonal, $R \in \mathbb{R}^{n \times n}$ uma matriz triangular superior e $P \in \mathbb{R}^{n \times n}$ uma matriz de permutação.

É determinado ainda um índice k tal que a submatriz R_{11} (situada nas k primeiras linhas e colunas de R) seja bem-condicionada e a submatriz R_{22} , desprezível:

$$\begin{aligned} R &= \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \\ &\approx \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix} \end{aligned}$$

Tal k é denominado *posto efetivo* de A e seu cômputo é discutido em [15].

A solução do sistema $Ax = b$ considerado pode ser então deduzida:

$$x = P \begin{pmatrix} R_{11}^{-1} \hat{c}_1 \\ 0 \end{pmatrix} \quad (2.22)$$

em que \hat{c}_1 consiste apenas nos k primeiros elementos de $c = Q^T b$.

Resta assinalar que essa estratégia foi escolhida como primeira opção por ser numericamente mais estável que a decomposição de Cholesky, já que, em geral, a matriz $J_F(x_k)$ não é bem-condicionada.

A fim de garantir que a subrotina responsável por essa decomposição fosse o mais robusta e confiável possível, recorreu-se ao projeto LAPACK [1] (versão 3.0), que provê uma ampla biblioteca de rotinas de Álgebra Linear codificadas em FORTRAN 77. Dentre elas, a empregada na implementação deste método é DGELSY.

Mesmo assim, nos casos em que a $J_F(x_k)$ não apresenta posto completo (o que pode ser conferido pela comparação $k < \min(m, n)$), a decomposição tem falhado em prover uma solução satisfatória ao sistema. Só daí a estratégia de resolução é trocada para a decomposição de Cholesky aplicada às equações normais, assunto de que trata a próxima subseção.

2.6.3 Decomposição de Cholesky

A resolução do sistema sobredeterminado $Ax = b$ pode ser encarada também como a tarefa de encontrar o mínimo do quadrado da norma dois de $Ax - b$ (problema de quadrados mínimos). Pelo simples fato de que $\|v\|_2^2 = v^T v$, é possível escrever:

$$\begin{aligned} \|Ax - b\|_2^2 &= (Ax - b)^T (Ax - b) \\ &= (Ax)^T (Ax) - b^T Ax - (Ax)^T b + b^T b \\ &= (Ax)^T (Ax) - 2(Ax)^T b + b^T b \end{aligned}$$

Sabendo-se que o mínimo dessa função ocorre no zero de sua derivada em respeito a x , vem:

$$\begin{aligned} \frac{d}{dx} [(Ax)^T (Ax) - 2(Ax)^T b + b^T b] &= 0 \\ 2A^T Ax - 2A^T b &= 0 \\ A^T Ax &= A^T b \end{aligned} \quad (2.23)$$

Aplicando-se tal resultado ao sistema de interesse, segue:

$$\begin{aligned} Ax &= b \\ J_F(x_k)(x_{k+1} - x_k) &= -F(x_k) \\ J_F^T(x_k) J_F(x_k)(x_{k+1} - x_k) &= -J_F^T(x_k) F(x_k) \end{aligned} \quad (2.24)$$

É preciso notar que a matriz $M = J_F^T(x_k)J_F(x_k) \in \mathbb{R}^{n \times n}$ é simétrica e semidefinida-positiva. Mais importante ainda é perceber que ela é *singular*, em razão de ser verdade que $\text{posto}(M) = \text{posto}(J_F(x_k))$ e por ser esse caminho de resolução tomado apenas quando $\text{posto}(J_F(x_k)) < n$.

Por esses motivos, é necessário empregar a decomposição de Cholesky *Modificada* [12], capaz de fatorar uma matriz $J_F^T(x_k)J_F(x_k)$ ligeiramente perturbada:

$$\begin{aligned} J_F^T(x_k)J_F(x_k)(x_{k+1} - x_k) &= -J_F^T(x_k)F(x_k) \\ (L^T L)(x_{k+1} - x_k) &= -J_F^T(x_k)F(x_k) \end{aligned}$$

sendo $L \in \mathbb{R}^{n \times n}$ uma matriz triangular inferior.

Com isso, o cálculo de $(x_{k+1} - x_k)$ segue facilmente por substituição para frente e substituição para trás. É interessante observar apenas que o número de condição $\kappa(J_F^T(x_k)J_F(x_k))$ é o quadrado de $\kappa(J_F(x_k))$, o que reforça a escolha de Cholesky Modificada para essa tarefa.

A subrotina encarregada de executá-la na implementação feita é `MODCHL` [13], creditada a E. Eskow e R. B. Schnabel (1988) e também codificada em `FORTAN 77`.

2.7 Pré e pós-processamento

Alguns desafios que tiveram de ser transpostos para o correto funcionamento do método merecem destaque e serão abordados nas seções que se seguem.

2.7.1 Detecção de itens soltos

Um obstáculo que se impôs à utilização do método foi a ocorrência de indeterminação dos sistemas de equações construídos, com $m < n$ em vez de $m \gg n$. Constatou-se que isso acontecia devido a itens que participavam de menos de dois contatos, contribuindo, assim, com a adição de mais variáveis (2) que equações (0 ou 1) ao sistema. Com certo abuso de linguagem, chamemos esses itens de “soltos”.

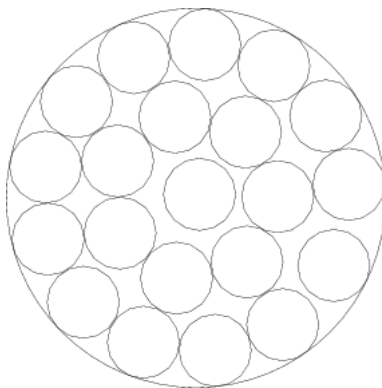


Figura 2.3: Exemplo de configuração com 2 itens “soltos”

A idéia para se contornar tal empecilho é a seguinte: durante o pré-processamento, tais itens devem ser encontrados e então temporariamente removidos do conjunto a ser empacotado. Apenas durante o pós-processamento, quando uma configuração obedecendo às equações determinadas

pelos contatos detectados já for conhecida (Figura 2.4), os soltos deverão ser reintroduzidos ao empacotamento.

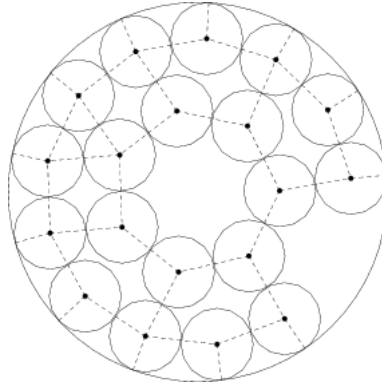


Figura 2.4: Configuração ótima com itens soltos suprimidos

Para isso, é formulado um novo problema de otimização baseado nos mesmos modelos apresentados em [10], porém agora com um número de variáveis bastante reduzido (proporcional ao número de itens soltos). A busca por uma solução do modelo é então conduzida pelo método ALGEN-CAN. Caso falhe a reincorporação de tais itens à configuração, tem-se uma evidência de que os contatos foram mal avaliados e de que o problema precisa ser resolvido novamente, dessa vez com ε_1 e ε_2 devidamente escolhidos. Caso contrário, tem-se uma solução ótima para o problema original (Figura 2.5).

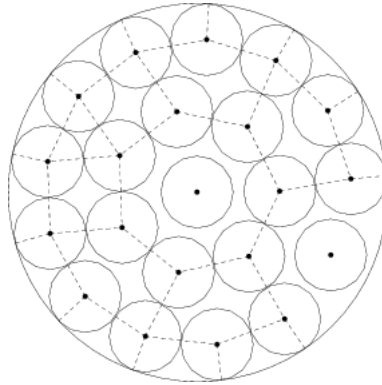


Figura 2.5: Configuração ótima com itens soltos devolvidos

Sejam S o conjunto dos itens soltos e s a sua cardinalidade. As variáveis que compõem o novo problema não-linear são o centro $c_i \in \mathbb{R}^2$ e o raio $r_i \in \mathbb{R}$ do i -ésimo item solto, para todo $i = 1, \dots, s$, além de uma nova variável $r_{min} \in \mathbb{R}$ introduzida para fins de modelagem. São dados do problema o centro $c_j \in \mathbb{R}^2$ de todo item $j \notin S$, cuja posição é obtida da solução x^* recém encontrada pelo método, as dimensões do objeto também conforme calculado pelo algoritmo, e o raio r dos itens no

empacotamento. O novo modelo formulado então fica:

$$\begin{aligned} \text{Minimizar} \quad & -r_{min} \\ \text{sujeito a} \quad & r_i \geq r_{min}, \quad \forall i \in S \end{aligned} \tag{2.25}$$

$$d(c_i, c_j) \geq r_i + r_j, \quad \forall i, j \in S, \quad i \neq j \tag{2.26}$$

$$d(c_i, c_j) \geq r_i + r, \quad \forall i \in S, \quad j \notin S \tag{2.27}$$

restrições de violação (Seção 2.1)

Se para o valor ótimo r_{min}^* obtido por ALGENCAN valer que $r_{min}^* \geq r$, então todos os itens soltos puderam ser realocados no empacotamento.

2.7.2 Iteração de item semi-fixado

Conforme já apontado durante a discussão sobre o método de Newton-Raphson, a convergência quadrática à solução só é garantida se a multiplicidade da raiz buscada for um. É verificável, porém, que, no empacotamento em círculos, existe não apenas uma, mas sim uma “vizinhança” de configurações ótimas arbitrariamente próximas, obtidas com uma simples rotação de todo o conjunto de itens no interior do objeto (Figura 2.6).

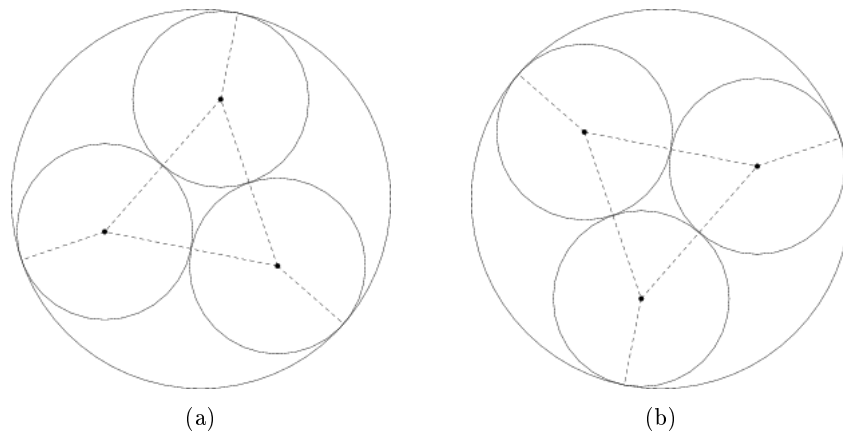


Figura 2.6: Configurações ótimas distintas para 3 itens

A fim de preservar a convergência quadrática do método, tal comportamento teve de ser prevenido escolhendo-se um dos itens do empacotamento para ter uma das coordenadas de seu centro fixada. Isso pode ser feito agregando-se ao sistema uma nova equação da forma $c_i^x - k = 0$, em que c_i^x denota a x -coordenada do centro do i -ésimo item e k o valor inicial dessa variável, como extraído da solução aproximada dada por ALGENCAN.

Infelizmente, não foi possível estabelecer um critério de escolha do item a ser submetido a esse processo. Em vez disso, a implementação feita itera esse papel entre todos os itens não-soltos, ficando a melhor solução encontrada. Com essa modificação, o algoritmo para objeto circular fica:

Algoritmo 2 Algoritmo em pseudo-código

Entrada: N := número de itens no empacotamento

Entrada: R := raio dos itens no empacotamento

```
1: while o tempo já consumido for menor que  $T$  do
2:   Execute ALGENCAN e obtenha uma solução  $x_0$  {aproximada}
3:   Realize o pré-processamento da solução
4:   Detecte os contatos estabelecidos na solução
5:   for all itens não-soltos ainda não fixados do
6:     Formule o sistema de equações não-lineares correspondente
7:     repeat
8:       Aplique o método de Newton usando a solução  $x_{k-1}$ 
9:       Resolva o novo sistema linear e obtenha um novo  $x_k$ 
10:    until o número de iterações tenha ultrapassado  $K$ 
11:    Salve a melhor das soluções obtidas para cada item fixado
12:    Restaure as condições iniciais do algoritmo como em (5)
13:  end for
14:  Realize o pós-processamento da nova solução
15:  Compare com a melhor já obtida e atualize-a
16: end while
17: return  $x^*$  {melhor solução encontrada}
```

2.7.3 Reescalamento da solução

Para que as respostas dadas pelo método possam ser comparadas às publicadas na literatura, é preciso garantir que elas não desobedeçam a qualquer uma das restrições de sobreposição e de violação impostas no modelo. Em outras palavras, deve-se ter certeza de que a configuração descrita por elas esteja correta sob o critério do rigor analítico. O cumprimento dessa exigência é feito submetendo a solução final encontrada a um reescalamento.

Tomando-se todos os itens do empacotamento dois a dois, é calculada a distância entre seus centros. Seja δ o menor valor encontrado. Claramente, vale que $d(c_i, c_j) \geq \delta$ para todo par de itens distintos (i, j) do empacotamento. Se $\delta \geq 2r$, então nenhuma restrição de sobreposição está sendo quebrada e, assim, nenhum ajuste quanto a isso se faz necessário. Por outro lado, se $\delta < 2r$, é preciso afastar os itens de modo a conseguir $d(c_i, c_j) \geq 2r, \forall i \neq j$. Vejamos:

$$\begin{aligned}d(c_i, c_j) &\geq \delta \\ \|c_i - c_j\|_2 &\geq \delta \\ \frac{2r}{\delta} \|c_i - c_j\|_2 &\geq 2r \\ \left\| \frac{2r}{\delta} (c_i - c_j) \right\|_2 &\geq 2r \\ \left\| \frac{2r}{\delta} c_i - \frac{2r}{\delta} c_j \right\|_2 &\geq 2r\end{aligned}$$

Disso, segue que pode ser obtida uma nova disposição tal que $d(c'_i, c'_j) \geq 2r, \forall i \neq j$, bastando para isso fazer $c'_i = c_i \cdot 2r/\delta$ para todo item i .

Resta apenas assegurar o cumprimento das restrições de violação, que variam com a forma do objeto. Para a circular, por exemplo, afere-se a distância entre o centro de cada item e o do

objeto. Sendo γ o maior desses valores, é correto afirmar que $d(c_i, C) \leq \gamma$ para todo item i do empacotamento. Se $\gamma \leq R - r$, então não há nenhuma restrição de violação desrespeitada e mais nenhuma correção precisa ser efetuada, já que $d(c_i, C) \leq R - r$ e, conseqüentemente, $d(c_i, C) + r \leq R$ para todo item i . Contudo, se $\gamma > R - r$, tem-se que $d(c_i, C) > R - r$ para algum i , sendo preciso definir para o objeto um novo raio R' tal que $\gamma = R' - r$. De fato:

$$\begin{aligned} d(c_i, C) &\leq \gamma \\ d(c_i, C) &\leq R' - r \\ d(c_i, C) + r &\leq R', \quad \forall i \end{aligned}$$

Já no caso dos demais formatos de objeto, é preciso computar quatro valores $\gamma_{esq}, \gamma_{dir}, \gamma_{inf}, \gamma_{sup}$, definidos como a maior violação das bordas esquerda, direita, inferior e superior do objeto, respectivamente. Suas expressões matemáticas são dadas por:

$$\gamma_{esq} = \max_i \{r - c_i^x\} \quad (2.28)$$

$$\gamma_{dir} = \max_i \{c_i^x + r - L\} \quad (2.29)$$

$$\gamma_{inf} = \max_i \{r - c_i^y\} \quad (2.30)$$

$$\gamma_{sup} = \begin{cases} \max_i \{c_i^y + r - L\} & \text{em quadrados} \\ \max_i \{c_i^y + r - W\} & \text{nos demais} \end{cases} \quad (2.31)$$

Finalmente, basta transladar os centros dos itens e redimensionar as dimensões do objeto apropriadamente a fim de eliminar quaisquer violações de suas bordas.

$$\text{Em quadrados} \quad \begin{cases} c_i^x = c_i^x + \gamma_{esq}, & \forall i \\ c_i^y = c_i^y + \gamma_{inf}, & \forall i \\ L = L + \max\{\gamma_{esq} + \gamma_{dir}, \gamma_{inf} + \gamma_{sup}\}, & \forall i \end{cases} \quad (2.32)$$

$$\text{Em strips} \quad \begin{cases} c_i^x = c_i^x + (\gamma_{esq} - \gamma_{dir})/2, & \forall i \\ c_i^y = c_i^y + \gamma_{inf}, & \forall i \\ W = W + \gamma_{inf} + \gamma_{sup}, & \forall i \end{cases} \quad (2.33)$$

$$\text{Em retângulos} \quad \begin{cases} c_i^x = c_i^x + \gamma_{esq}, & \forall i \\ c_i^y = c_i^y + \gamma_{inf}, & \forall i \\ L = L + \gamma_{esq} + \gamma_{dir}, & \forall i \\ W = W + \gamma_{inf} + \gamma_{sup}, & \forall i \end{cases} \quad (2.34)$$

Uma última ressalva quanto a esse procedimento se faz necessária — em empacotamentos em *strips*, por terem tais objetos a sua dimensão ao longo do eixo x fixada, a translação definida acima só é capaz de minimizar eventuais violações $\gamma_{esq}, \gamma_{dir}$, porém nenhuma operação desse tipo será bem sucedida em eliminá-las completamente.

Realizados esses pequenos ajustes, torna-se possível comparar as respostas conseguidas pelo método com as demais já conhecidas, o que será feito no próximo capítulo.

Capítulo 3

Resultados

O método descrito nesta monografia foi implementado na linguagem FORTRAN 77 e o programa resultante submetido a testes, cujos resultados constam a seguir.

3.1 Experimentos numéricos

Todos os testes foram executados em uma máquina INTEL CORE 2 QUAD 2,4GHz com 4Gb de memória RAM e sistema operacional GNU/LINUX. Além disso, o código foi compilado pelo GNU FORTRAN COMPILER (g77) 3.4.6 com a diretiva `-O4` de otimização habilitada.

Testaram-se instâncias do problema com número de itens variando de 1 a 50. Os valores obtidos para o raio de objetos circulares e o lado de objetos quadrados em cada um dos casos foram comparados à base de dados PACKOMANIA [23] e estão listados nas Tabelas 3.1 e 3.2, respectivamente. Nelas a primeira coluna contém o número de itens de cada empacotamento, a segunda a melhor resposta fornecida pelo método implementado, a terceira os valores de referência, a quarta a diferença entre eles e a quinta o tempo de CPU (em segundos) transcorrido até a obtenção de tal resposta pelo algoritmo.

Já os resultados conseguidos para *strips* e retângulos não puderam ser comparados devido à falta de dados de referência disponíveis para os problemas considerados. Assim, as Tabelas 3.3, 3.4 e 3.5 não possuem a terceira e quarta colunas descritas no parágrafo anterior.

3.2 Análise e comentários

Primeiramente, é preciso atentar para o fato de que, apesar de os valores de referência divulgados pelo projeto PACKOMANIA para empacotamentos em círculos possuírem até 30 casas decimais, a precisão das respostas dadas pela implementação feita é não mais que 10^{-16} . Portanto, diferenças dessa ordem de magnitude entre os dois valores podem ser creditadas apenas a imprecisões de representação de números pela máquina. Dito isso, é possível constatar pela Tabela 3.1 que os valores ótimos conhecidos já foram constatados para 48 das 50 instâncias testadas do problema.

Ainda no que tange à qualidade das soluções obtidas, ressalta-se que os valores de referência para objetos quadrados foram disponibilizados com apenas 12 casas decimais, o que justifica o fato de as menores diferenças para esses problemas serem da ordem de 10^{-12} . Da Tabela 3.2, conclui-se então que em 40 dos 50 empacotamentos testados o método foi bem sucedido em igualar as melhores

respostas já conhecidas, o que se mostra expressivo tendo em vista se tratar de uma nova extensão, menos aprimorada que a sua anterior (para objetos circulares).

Tais resultados satisfatórios, somados ao fato de que o algoritmo foi capaz de determinar uma solução viável para cada uma das 50 instâncias testadas para *strips* e retângulos (relacionadas nas Tabelas 3.3, 3.4 e 3.5), sugerem que essas configurações estejam também ao menos razoavelmente próximas das ótimas, quiçá idênticas. Como não há hoje uma base de dados com a qual se possa estabelecer uma comparação confiável, cabe ao autor deste trabalho divulgar seus resultados com a esperança de que, no futuro próximo, possam ser confrontados com novos que surgirem.

# de itens	Resultado	Referência	Diferença	Tempo (s)
1	1.0000000000000000	1.0000000000000000	0.0E+00	0.00
2	2.0000000000000000	2.0000000000000000	0.0E+00	0.11
3	2.1547005383792515	2.1547005383792515	0.0E+00	0.00
4	2.4142135623730949	2.4142135623730949	0.0E+00	0.00
5	2.7013016167040798	2.7013016167040798	0.0E+00	0.08
6	3.0000000000000000	3.0000000000000000	0.0E+00	0.02
7	3.0000000000000000	3.0000000000000000	0.0E+00	0.00
8	3.3047648709624866	3.3047648709624866	0.0E+00	0.04
9	3.6131259297527532	3.6131259297527532	0.0E+00	0.35
10	3.8130256313981246	3.8130256313981246	0.0E+00	0.14
11	3.9238044001630872	3.9238044001630872	0.0E+00	30.37
12	4.0296019301161836	4.0296019301161836	0.0E+00	16.42
13	4.2360679774997898	4.2360679774997898	0.0E+00	0.91
14	4.3284285548608370	4.3284285548608370	0.0E+00	9.75
15	4.5213569647061647	4.5213569647061647	0.0E+00	29.16
16	4.6154255948731944	4.6154255948731944	0.0E+00	71.36
17	4.7920337483105797	4.7920337483105788	8.9E-16	6.52
18	4.8637033051562728	4.8637033051562728	0.0E+00	9.70
19	4.8637033051562728	4.8637033051562728	0.0E+00	12.24
20	5.1223207369915293	5.1223207369915285	8.9E-16	9.07
21	5.2523174750102433	5.2523174750102424	8.9E-16	36.09
22	5.4397189590722146	5.4397189590722137	8.9E-16	87.33
23	5.5452042225748590	5.5452042225748581	8.9E-16	140.86
24	5.6516610917654377	5.6516610917654369	8.9E-16	29.83
25	5.7528243308571163	5.7528243308571154	8.9E-16	388.56
26	5.8281765369429506	5.8281765369429497	8.9E-16	42.27
27	5.9063978473941567	5.9063978473941550	1.8E-15	2490.16
28	6.0149380973715152	6.0149380973715125	2.7E-15	14164.40
29	6.1385979039781473	6.1385979039781455	1.8E-15	98.04
30	6.1977410708792275	6.1977410708792258	1.8E-15	270.06
31	6.2915026221291814	6.2915026221291814	0.0E+00	71.60
32	6.4294629709501150	6.4294629709501132	1.8E-15	790.48
33	6.4867381281037089	6.4867031235604333	3.5E-05	14877.17
34	6.6109570900010057	6.6109570900010031	2.7E-15	8803.57
35	6.6971710917902456	6.6971710917902438	1.8E-15	3188.97
36	6.7467537934241761	6.7467537934241752	8.9E-16	258.74
37	6.7587704831436346	6.7587704831436337	8.9E-16	1097.84
38	6.9618869652281514	6.9618869652281488	2.7E-15	639.76
39	7.0578841626240081	7.0578841626240045	3.6E-15	246.10
40	7.1238464359431282	7.1238464359431246	3.6E-15	273.35
41	7.2600123286770479	7.2600123286770462	1.8E-15	638.11
42	7.3469494647914715	7.3467964069427687	1.5E-04	5295.15
43	7.4199448563412131	7.4199448563412114	1.8E-15	3615.91
44	7.4980366829952523	7.4980366829952487	3.6E-15	1326.19
45	7.5729123263675255	7.5729123263675211	4.4E-15	3176.21
46	7.6501799146936715	7.6501799146936680	3.6E-15	845.46
47	7.7241700525980201	7.7241700525980184	1.8E-15	2433.36
48	7.7912714305586714	7.7912714305586679	3.6E-15	12876.93
49	7.8868709588028691	7.8868709588028647	4.4E-15	14088.66
50	7.9475152747835143	7.9475152747835107	3.6E-15	1575.35

Tabela 3.1: Resultados de empacotamentos em círculos

# de itens	Resultado	Referência	Diferença	Tempo (s)
1	2.0000000000000000	2.000000000000	0.0E+00	0.00
2	3.4142135623730949	3.41421356237	-5.3E-12	0.00
3	3.9318516525781364	3.93185165258	-3.9E-12	0.02
4	4.0000000000000000	4.000000000000	0.0E+00	0.00
5	4.8284271247461898	4.82842712473	1.1E-11	0.01
6	5.3282011773513762	5.32820117736	-1.4E-11	0.48
7	5.7320508075688776	5.73205080756	-3.1E-13	9.42
8	5.8637033051562746	5.86370330515	-1.9E-12	4.78
9	6.0000000000000000	6.000000000000	1.2E-11	0.64
10	6.7474415232381135	6.74744152324	-1.0E-11	40.27
11	7.0225095034303822	7.02250950342	9.8E-12	477.26
12	7.1449575542752672	7.14495755429	-2.2E-11	124.70
13	7.4631768820241113	7.46304782885	1.3E-04	2.97
14	7.7320508075688776	7.73205080757	-2.0E-12	71.94
15	7.8637033051562764	7.86370330516	-7.7E-12	133.07
16	8.0000000000000000	8.000000000000	0.0E+00	0.00
17	8.5326603474980978	8.53266034749	3.7E-12	846.87
18	8.6564023547027524	8.65640235470	3.9E-14	13.03
19	8.9074609393260822	8.90746093932	3.0E-13	6.22
20	8.9780833528217379	8.97808335286	-3.9E-11	8.60
21	9.3580199588727595	9.35801995887	-5.6E-12	1135.21
22	9.4639295431339381	9.46384509097	8.4E-05	12.86
23	9.7274066103125492	9.72740661029	2.0E-11	4330.55
24	9.8637033051562764	9.86370330511	3.8E-11	220.21
25	10.0000000000000000	10.000000000000	0.0E+00	0.81
26	10.3774982039134311	10.3774982039	1.2E-11	275.04
27	10.4799830400508842	10.4799830400	7.0E-12	915.63
28	10.6754536943453164	10.6754536943	2.4E-11	80.03
29	10.8151200175936939	10.8151200176	-3.6E-11	21.71
30	10.9085683308339956	10.9085683308	1.4E-12	28.68
31	11.1934033520469818	11.1934033520	-2.9E-11	205.91
32	11.3819824265232441	11.3819824264	2.7E-11	782.35
33	11.4641016151377571	11.4639440323	1.6E-04	113.58
34	11.7274066103125492	11.7274066102	6.5E-11	6109.47
35	11.8637033051562764	11.8637033052	-5.0E-11	1705.13
36	12.0000000000000000	12.000000000000	-4.8E-11	28.08
37	12.1818588307319349	12.1817863967	7.2E-05	6274.72
38	12.2389635913615287	12.2384376438	5.3E-04	6541.01
39	12.2899151085505363	12.2899151085	3.9E-12	117.32
40	12.6283749264972833	12.6283749265	-4.5E-11	2133.86
41	12.7469384531873349	12.7469384531	1.3E-11	104.11
42	12.8532221454500828	12.8532221454	-2.7E-11	69.30
43	13.0994720835212828	13.0993251411	1.5E-04	1970.51
44	13.1958689906919524	13.1957481262	1.2E-04	12415.98
45	13.3819824265232494	13.3819824265	2.6E-12	83.15
46	13.4641016151377588	13.4639878881	1.1E-04	146.45
47	13.6775877082279198	13.6774298825	1.6E-04	4509.99
48	13.8059970535441447	13.8059970536	-9.5E-11	1713.92
49	13.9484250865937067	13.9484250865	7.3E-11	3746.27
50	14.0124815721935363	14.0100949163	2.4E-03	14263.90

Tabela 3.2: Resultados de empacotamentos em quadrados

# de itens	Resultado	Tempo (s)
1	2.0000000000000000	0.00
2	1.9999999999999998	0.03
3	1.9999999999999998	0.03
4	1.9999999999999998	0.04
5	2.6959705453537524	0.02
6	3.3228756555322954	0.00
7	3.5612494995995996	0.00
8	3.6887986310766987	0.07
9	3.9996673748986948	9.20
10	4.6959705453537524	0.04
11	5.1224989991991992	0.05
12	5.3775972621533974	2.66
13	5.8538814987957917	824.89
14	5.9993347497973915	1102.88
15	6.6959705453537559	0.02
16	7.0663958932300979	5.33
17	7.4525364626094142	7.86
18	7.8539155052528402	696.39
19	7.9996778073991033	2518.06
20	8.6959705453537559	41.24
21	9.1732901304367189	3111.61
22	9.4524415753023714	31.15
23	9.8537288511462791	2460.95
24	9.9993623469257145	155.29
25	10.6959705453537577	0.67
26	11.1733168632638993	2774.49
27	11.4522265853969447	4575.57
28	11.8539308197359858	95.07
29	11.9993600450532973	827.04
30	12.6959705453537595	1.34
31	13.1733140058769500	6671.62
32	13.4519215824209422	6682.79
33	13.8515722749965562	629.29
34	13.9993894685688876	2800.31
35	14.6959705453537612	8.38
36	15.1731171516082881	5425.17
37	15.4520220872811613	1510.42
38	15.8330122294352655	564.42
39	15.999999998085887	1005.56
40	16.6959705453537595	14.85
41	17.1729034249762442	6251.27
42	17.4516358985406050	2438.89
43	17.8140606308452512	2584.16
44	17.9999999989393835	4689.29
45	18.6959705453537630	3.68
46	19.1726994816344884	1478.88
47	19.4518461649909042	4186.35
48	19.7495596428938782	6529.85
49	19.9993724839986804	3455.25
50	20.6959705453537630	4.99

Tabela 3.3: Resultados de empacotamentos em *strips* ($L = 9.5$)

# de itens	Dimensão L	Dimensão W	$L \times W$	Tempo (s)
1	2.0000000000000000	2.0000000000000000	4.0000000000000000	0.00
2	4.0000000000000000	2.0000000000000000	8.0000000000000000	0.00
3	6.0000000000000000	2.0000000000000000	12.0000000000000000	0.00
4	8.0000000000000000	2.0000000000000000	16.0000000000000000	0.00
5	10.0000000000000000	2.0000000000000000	20.0000000000000000	0.00
6	4.0000000000000000	6.0000000000000000	24.0000000000000000	0.32
7	2.0000000000000000	14.0000000000000000	28.0000000000000000	0.65
8	8.0000000000000000	4.0000000000000000	32.0000000000000000	1.46
9	6.0000000000000000	6.0000000000000000	36.0000000000000000	0.01
10	10.0000000000000000	4.0000000000000000	40.0000000000000000	0.00
11	8.0000000000000018	5.4641016151377553	43.7128129211020493	11.27
12	8.0000000000000000	6.0000000000000000	48.0000000000000000	4.29
13	26.0000000000000000	2.0000000000000000	52.0000000000000000	7.31
14	5.4641016151377553	10.0000000000000018	54.6410161513775634	12.65
15	3.7320508075688776	16.0000000000000036	59.7128129211020564	8.66
16	3.7320508075688776	17.0000000000000036	63.4448637286709314	16.36
17	5.4641016151377553	12.0000000000000018	65.5692193816530704	30.05
18	3.7320508075688776	19.0000000000000036	70.9089653438086884	60.81
19	7.4641016151377562	10.0000000000000018	74.6410161513775705	8.63
20	14.0000000000000036	5.4641016151377553	76.4974226119285987	0.01
21	15.0000000000000036	5.4641016151377553	81.9615242270663487	0.02
22	3.7320508075688776	23.0000000000000036	85.8371685740842025	26.53
23	5.4641016151377553	16.0000000000000036	87.4256258422040986	8.04
24	5.4641016151377553	17.0000000000000036	92.8897274573418628	35.05
25	3.7320508075688776	26.0000000000000071	97.0333209967908488	23.42
26	5.4641016151377553	18.0000000000000036	98.3538290724796127	114.74
27	5.4641016151377553	19.0000000000000036	103.8179306876173627	10.31
28	12.0000000000000036	8.9282032302755123	107.1384387633061834	80.62
29	20.0000000000000036	5.4641016151377553	109.2820323027551268	78.87
30	5.4641016151377553	21.0000000000000036	114.7461339178928768	9.93
31	3.7320508075688776	32.0000000000000071	119.4256258422041128	83.87
32	5.4641016151377553	22.0000000000000036	120.2102355330306409	176.68
33	14.0000000000000053	8.9282032302755123	124.9948452238572258	1672.70
34	7.1961524227066338	18.0000000000000036	129.5307436087194333	142.33
35	24.0000000000000036	5.4641016151377553	131.1384387633061408	53.69
36	25.0000000000000071	5.4641016151377553	136.6025403784439334	186.64
37	5.4641016151377553	25.8612097182042078	141.3082777906558363	2355.32
38	26.0000000000000071	5.4641016151377553	142.0666419935816691	38.70
39	27.0000000000000071	5.4641016151377553	147.5307436087194333	301.38
40	7.1961524227066338	21.0000000000000036	151.1192008768393293	329.70
41	5.4641016151377553	28.0000000000000071	152.9948452238571974	233.12
42	22.0000000000000036	7.1961524227066338	158.3153532995459614	289.37
43	18.0000000000000071	8.9282032302755123	160.7076581449592823	406.87
44	5.4641016151377553	30.0000000000000071	163.9230484541326973	1444.90
45	5.4641016151377553	31.0000000000000071	169.3871500692704615	507.19
46	7.1961524227066338	24.0000000000000036	172.7076581449592254	490.41
47	5.4641016151377553	32.0000000000000071	174.8512516844081972	515.30
48	20.0000000000000071	8.9282032302755123	178.5640646055103105	2965.15
49	5.4641016151377553	33.8612097182042078	185.0210907117578643	5050.76
50	5.4641016151377553	34.0000000000000071	185.7794549146837255	82.93

Tabela 3.4: Resultados de empacotamentos em retângulos (min. $L \times W$)

# de itens	Dimensão L	Dimensão W	$L + W$	Tempo (s)
1	2.0000000000000000	2.0000000000000000	4.0000000000000000	0.00
2	4.0000000000000000	2.0000000000000000	6.0000000000000000	0.00
3	3.7320508075688776	4.0000000000000000	7.7320508075688776	0.00
4	4.0000000000000000	4.0000000000000000	8.0000000000000000	0.17
5	4.0000000000000000	5.4641016151377544	9.4641016151377535	0.23
6	4.0000000000000000	6.0000000000000000	10.0000000000000000	0.32
7	5.8612097182041998	5.4641016151377553	11.3253113333419542	1.06
8	5.4641016151377553	6.0000000000000009	11.4641016151377571	1.50
9	6.0000000000000000	6.0000000000000000	12.0000000000000000	0.00
10	7.1961524227066338	6.0000000000000009	13.1961524227066356	44.19
11	6.0000000000000009	7.4641016151377562	13.4641016151377571	3.71
12	6.0000000000000000	8.0000000000000000	14.0000000000000000	5.46
13	7.4626564857803901	7.4632672693142670	14.9259237550946580	1926.49
14	7.1961524227066338	8.0000000000000018	15.1961524227066356	41.85
15	8.0000000000000018	7.4641016151377562	15.4641016151377571	15.53
16	8.0000000000000000	8.0000000000000000	16.0000000000000000	26.07
17	8.9282032302755123	7.9427193491449888	16.8709225794205011	731.28
18	8.0000000000000036	8.9282032302755123	16.9282032302755141	288.89
19	7.4641016151377562	10.0000000000000018	17.4641016151377571	8.11
20	8.9282032302755123	9.0000000000000036	17.9282032302755141	539.07
21	9.4337452285295686	9.2209018981307658	18.6546471266603362	112.60
22	9.9427193491449888	8.9282032302755123	18.8709225794205011	982.69
23	8.9282032302755123	10.0000000000000036	18.9282032302755141	401.12
24	9.4641016151377553	10.0000000000000000	19.4641016151377571	87.86
25	11.0000000000000036	8.9282032302755123	19.9282032302755141	213.09
26	10.6602540378443891	9.9427193491449888	20.6029733869893761	3543.59
27	10.6602540378443891	10.0000000000000036	20.6602540378443926	89.14
28	12.0000000000000036	8.9282032302755123	20.9282032302755141	74.64
29	9.4641016151377571	12.0000000000000018	21.4641016151377571	23.65
30	10.6602540378443891	11.0000000000000036	21.6602540378443926	416.69
31	10.9282032302755141	11.4265717909344140	22.3547750212099281	2152.24
32	11.9427193491449870	10.6602540378443891	22.6029733869893761	1442.84
33	12.0000000000000036	10.6602540378443891	22.6602540378443926	216.60
34	10.9282032302755123	12.0000000000000036	22.9282032302755141	739.16
35	12.3923048454132676	11.0000000000000036	23.3923048454132712	136.00
36	10.6602540378443891	13.0000000000000071	23.6602540378443962	84.79
37	12.3923048454132676	11.8612097182042024	24.2535145636174718	7134.12
38	11.9841557353269081	12.3924132707237522	24.3765690060506586	2887.87
39	12.3923048454132676	12.0000000000000036	24.3923048454132712	3357.04
40	12.9282032302755123	12.0000000000000036	24.9282032302755141	156.03
41	12.3923255275586577	12.9454336393696003	25.3377591669282580	1188.44
42	12.3923048454132676	13.0000000000000071	25.3923048454132747	1661.79
43	13.6029140930960750	12.3923268915991969	25.9952409846952719	1037.94
44	14.3131610181206010	12.0000246095545560	26.3131856276751570	7067.18
45	13.9841229965638760	12.3923048454132676	26.3764278419771436	1034.79
46	14.0000000000000071	12.6602540378443891	26.6602540378443962	1859.56
47	14.0000000000000071	12.9282032302755123	26.9282032302755212	1295.91
48	13.0000000000000071	14.1243556529821479	27.1243556529821532	1705.95
49	12.3923048454132676	15.0000000000000071	27.3923048454132747	7192.26
50	12.3923048454132712	15.6028097181778964	27.9951145635911658	6074.76

Tabela 3.5: Resultados de empacotamentos em retângulos (min. $L + W$)

Capítulo 4

Conclusão

O presente trabalho foi desenvolvido conforme proposto em seu plano inicial, ainda que nem todas as extensões possíveis tenham sido implementadas. Os resultados esperados foram de fato alcançados e, sob certos aspectos, até mesmo superados. O método elaborado e a sua correspondente implementação encerram em si diversos conceitos importantes, tanto teóricos quanto práticos, podendo servir de insumo a estudos posteriores.

Dentre suas contribuições está o significativo aperfeiçoamento do método oriundo de [10], não apenas quanto à qualidade das soluções providas, mas também quanto ao custo computacional de obtê-las. Ademais, proporcionou inestimável aprendizado de teoria da Otimização, além de experiência em um dos mais respeitáveis ambientes de pesquisa acadêmica nesse campo.

Tanto a versão mais atualizada da implementação realizada do método quanto os melhores resultados conseguidos por meio dela encontram-se em <http://www.linux.ime.usp.br/~jgmarcel/mac499/sources>. As representações gráficas correspondentes às soluções dadas nas tabelas do capítulo anterior, por sua vez, estão reunidas no Apêndice A. Nelas, as linhas tracejadas denotam os contatos estabelecidos. Por fim, deve-se ressaltar que, por meras questões tipográficas, a escala entre tais figuras *não* pôde ser preservada, ao contrário do que acontece com suas versões originais disponibilizadas no endereço supracitado.

Apêndice A

Imagens produzidas

A.1 Objeto circular

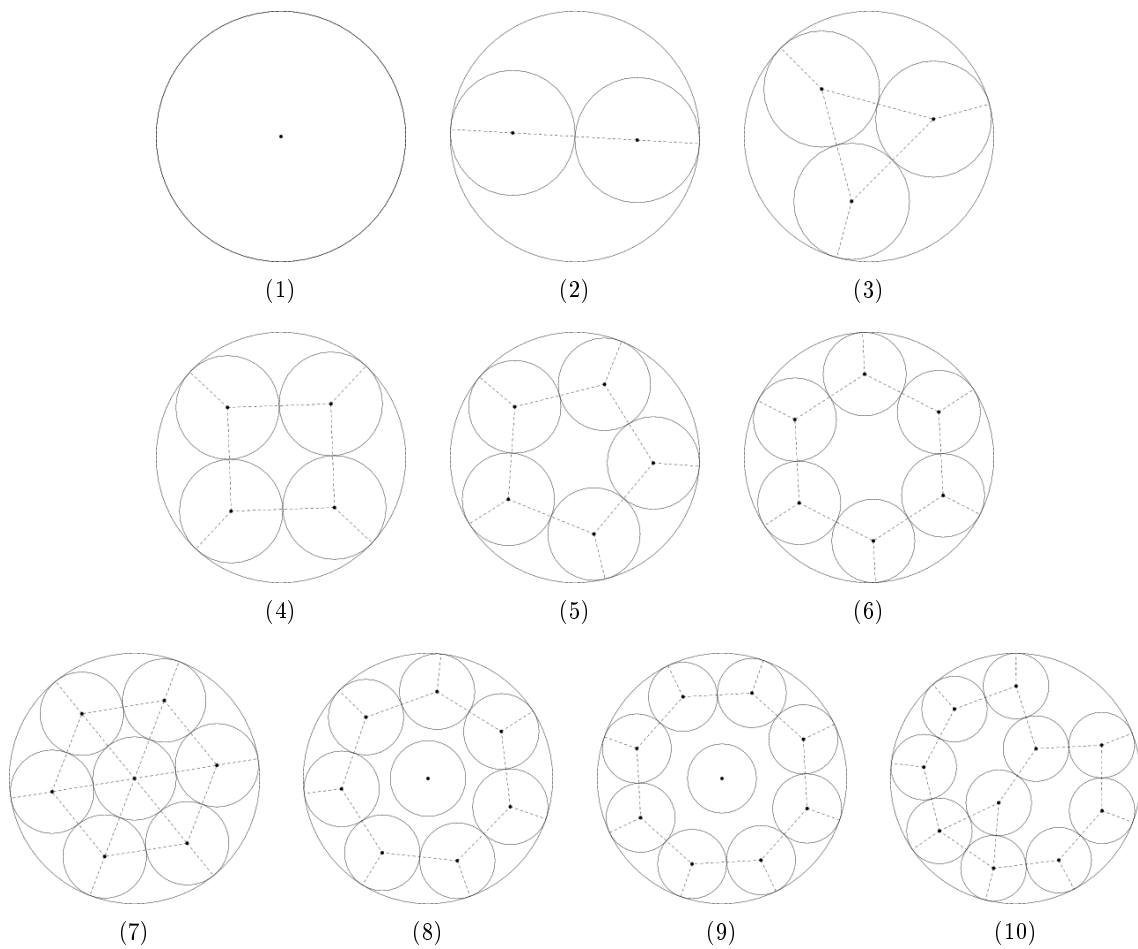


Figura A.1: Representações de empacotamentos de 1 a 10 itens

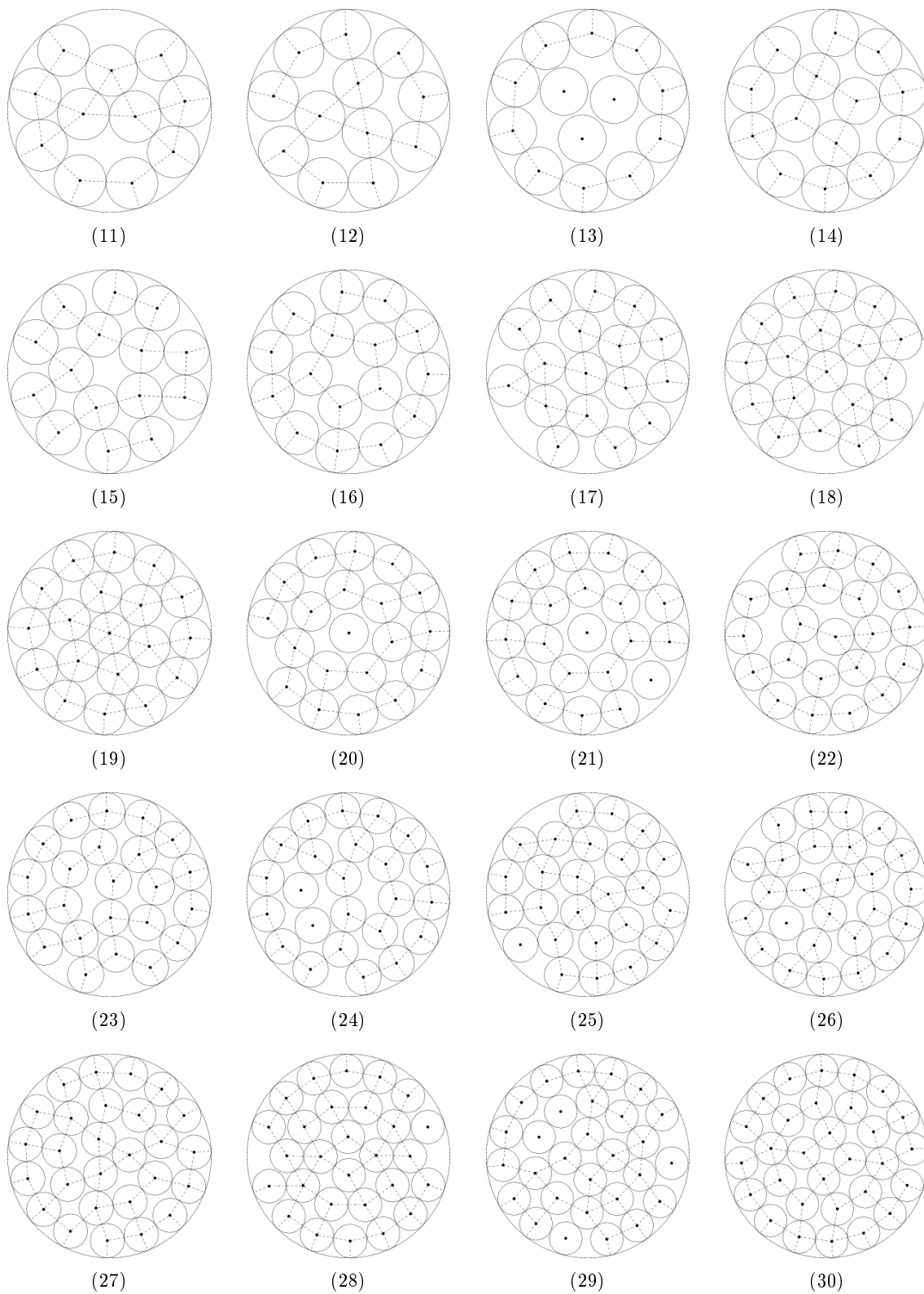


Figura A.2: Representações de empacotamentos de 11 a 30 itens

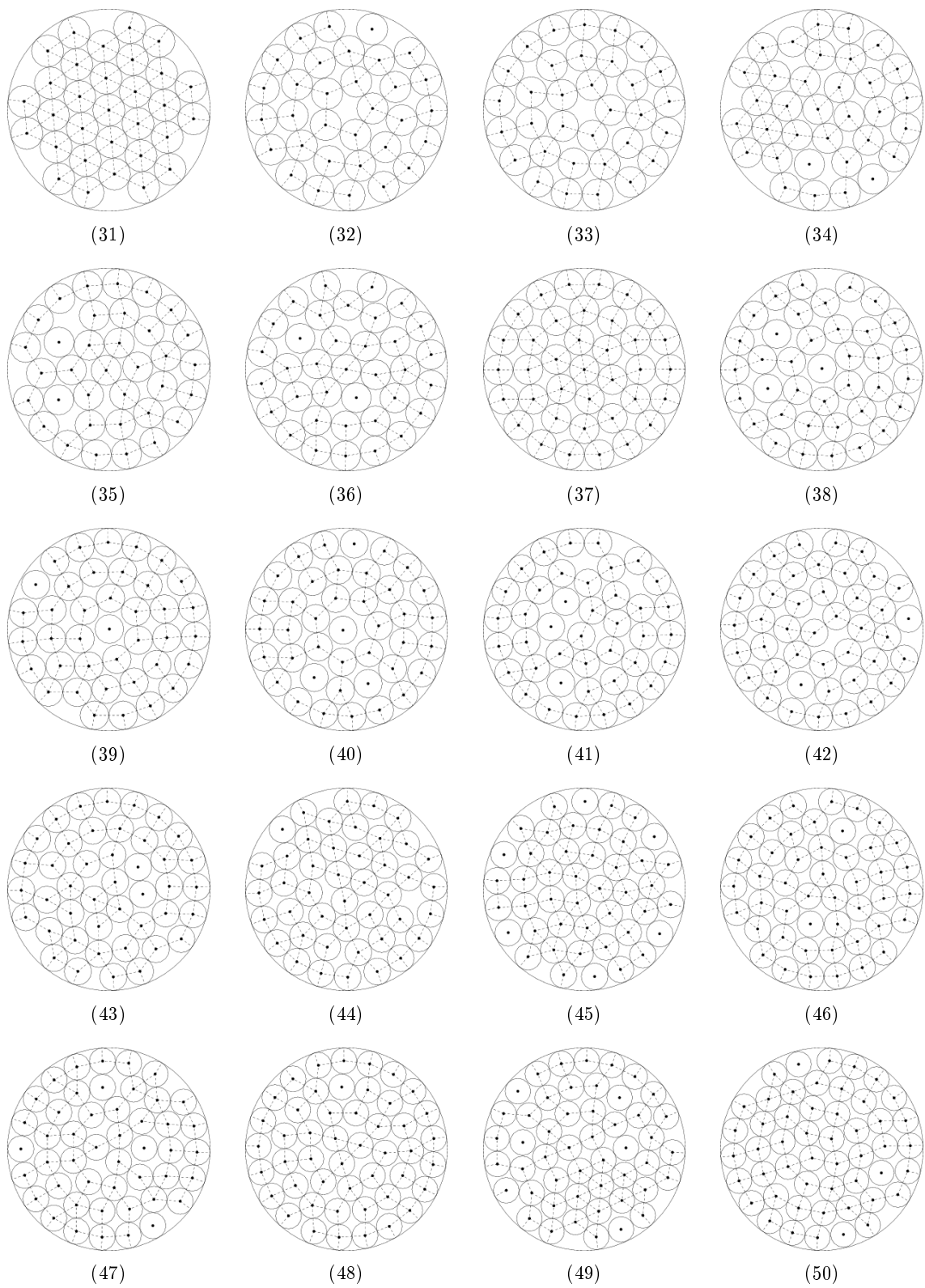


Figura A.3: Representações de empacotamentos de 31 a 50 itens

A.2 Objeto quadrado

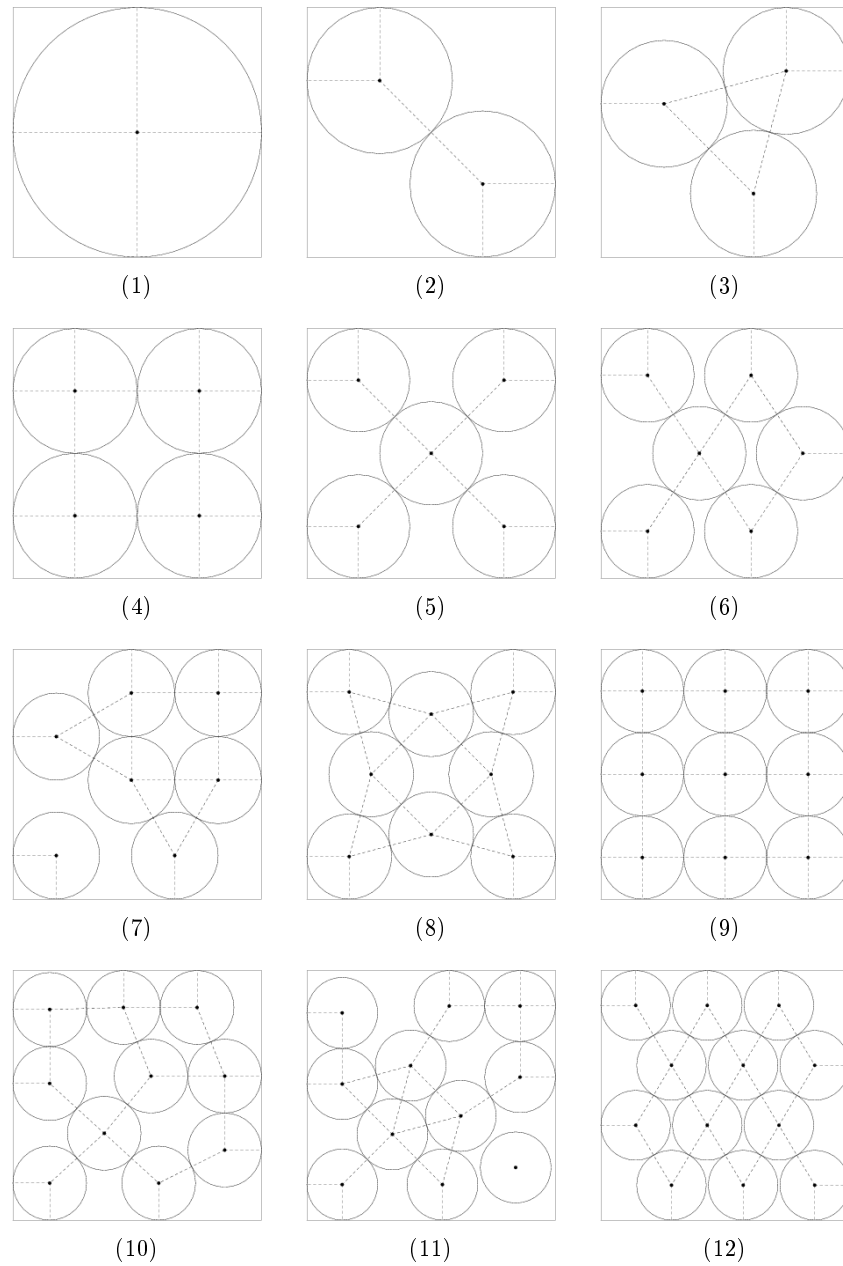


Figura A.4: Representações de empacotamentos de 1 a 12 itens

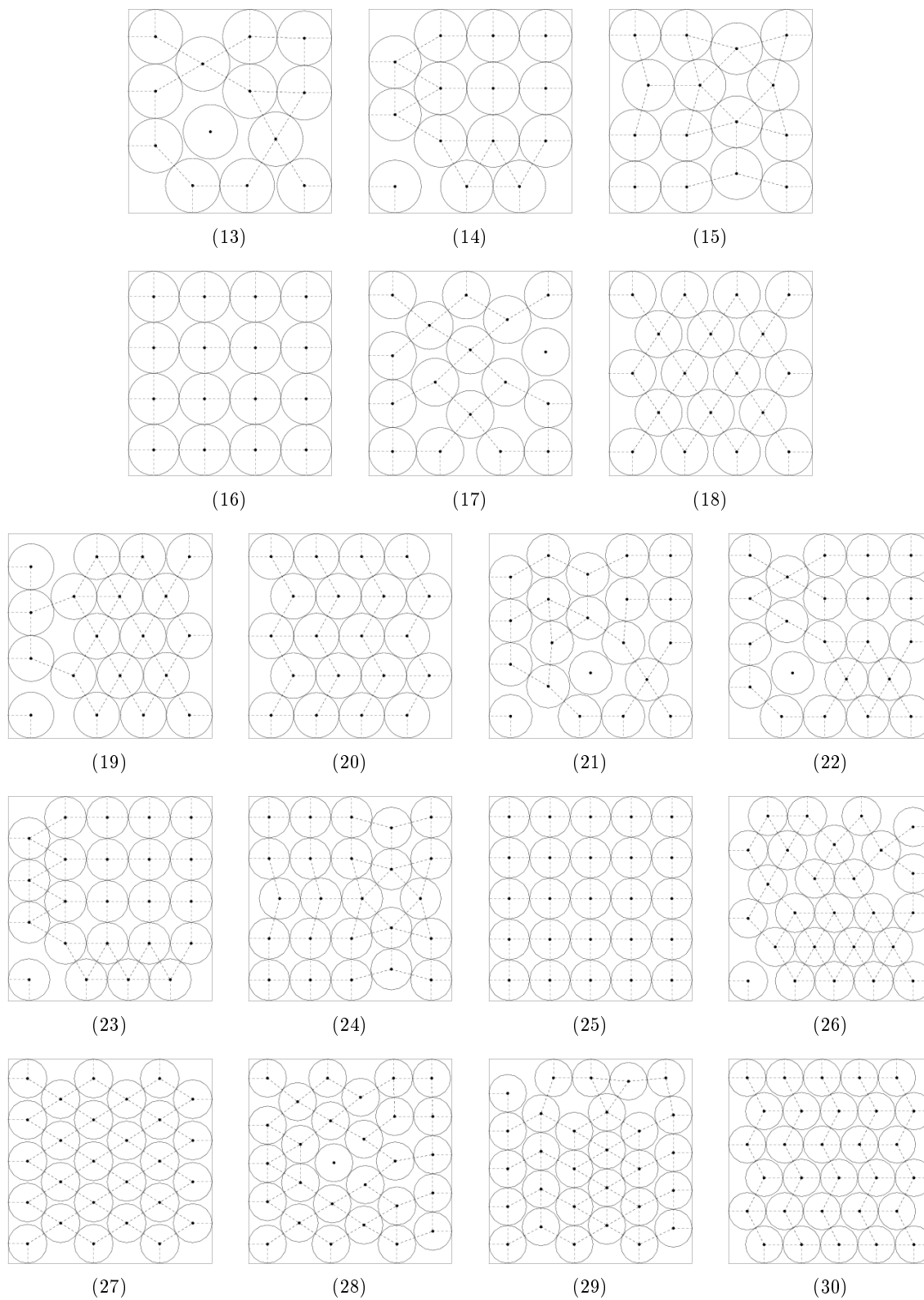


Figura A.5: Representações de empacotamentos de 13 a 30 itens

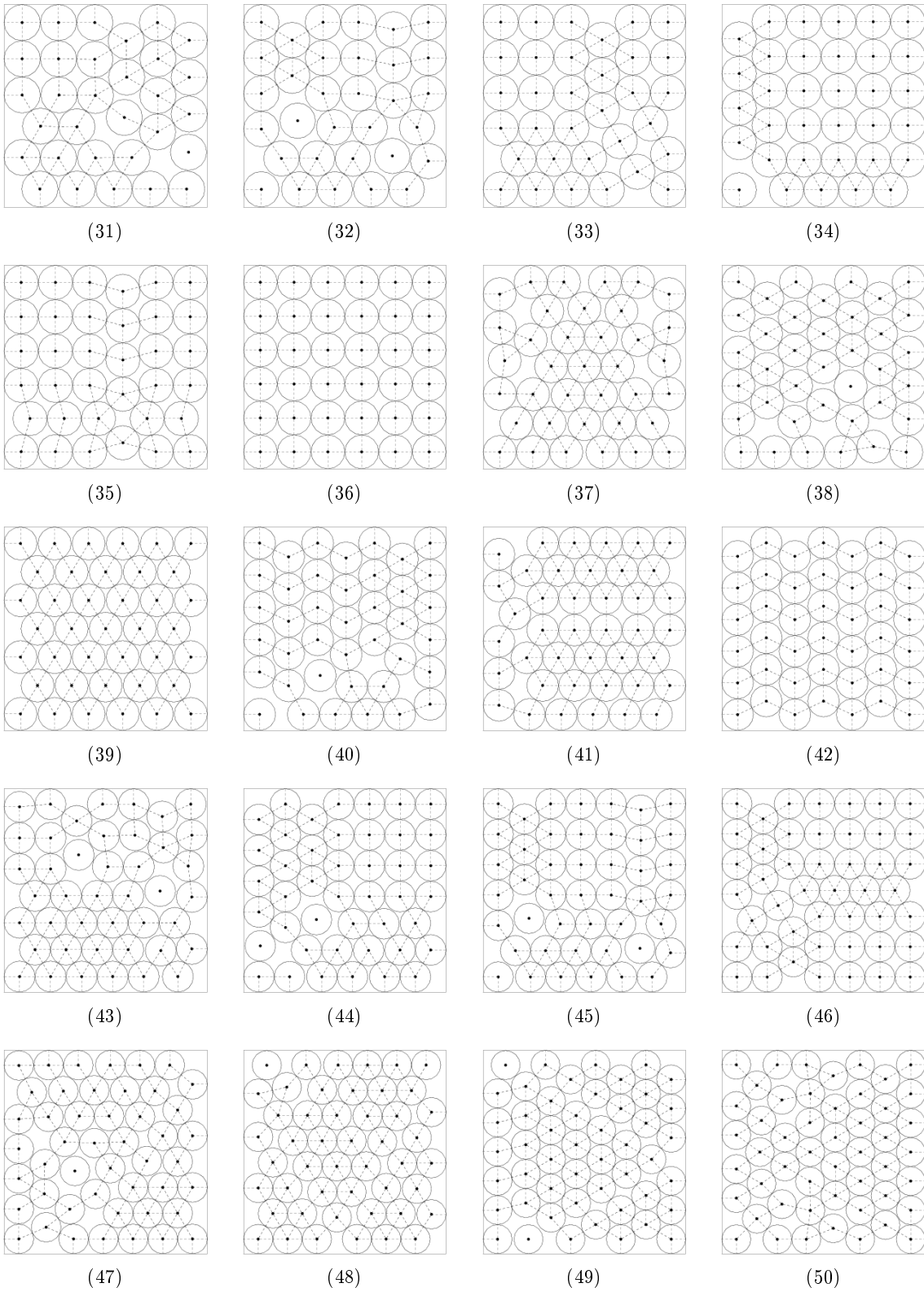


Figura A.6: Representações de empacotamentos de 31 a 50 itens

A.3 Objeto em *strip*

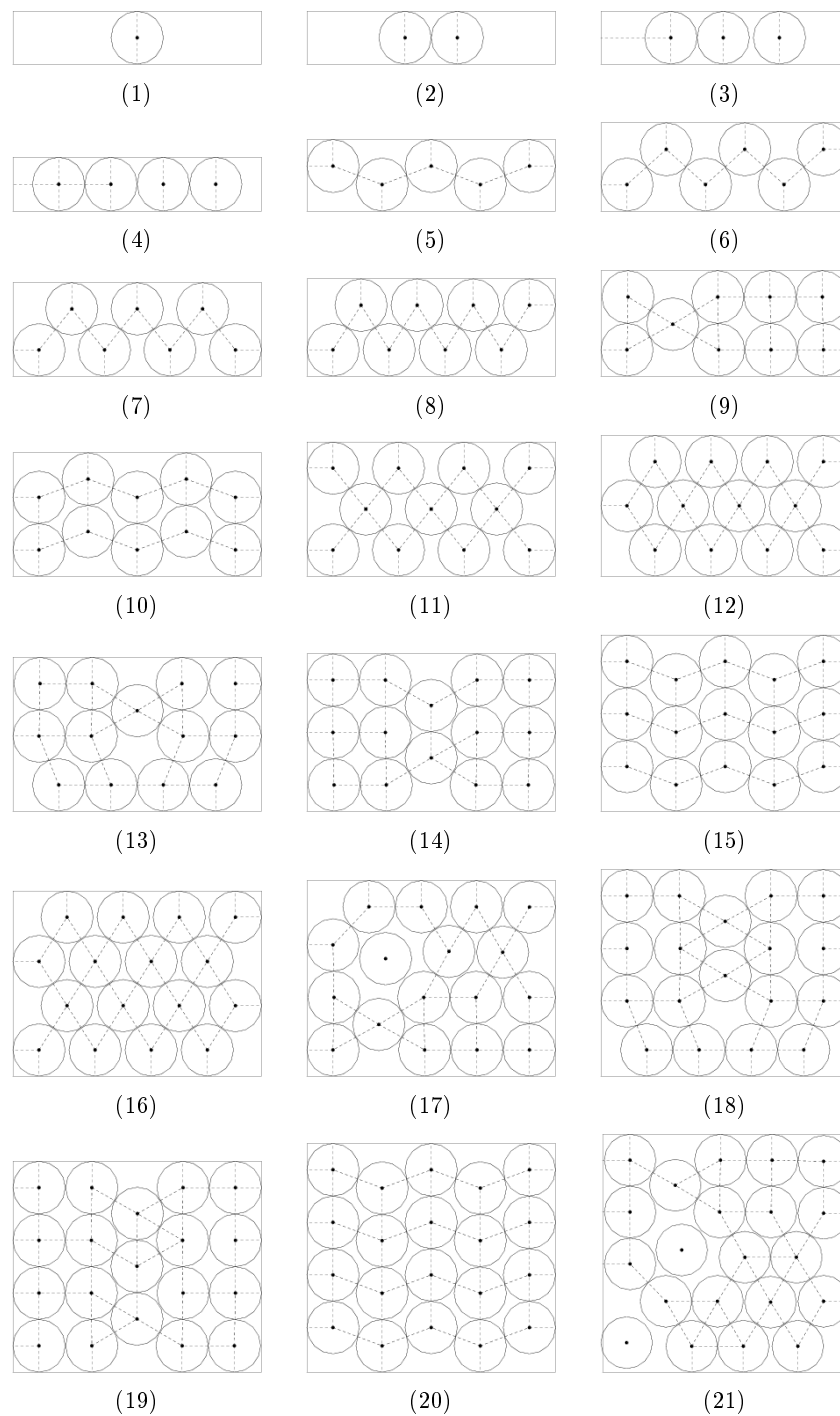


Figura A.7: Representações de empacotamentos de 1 a 21 itens

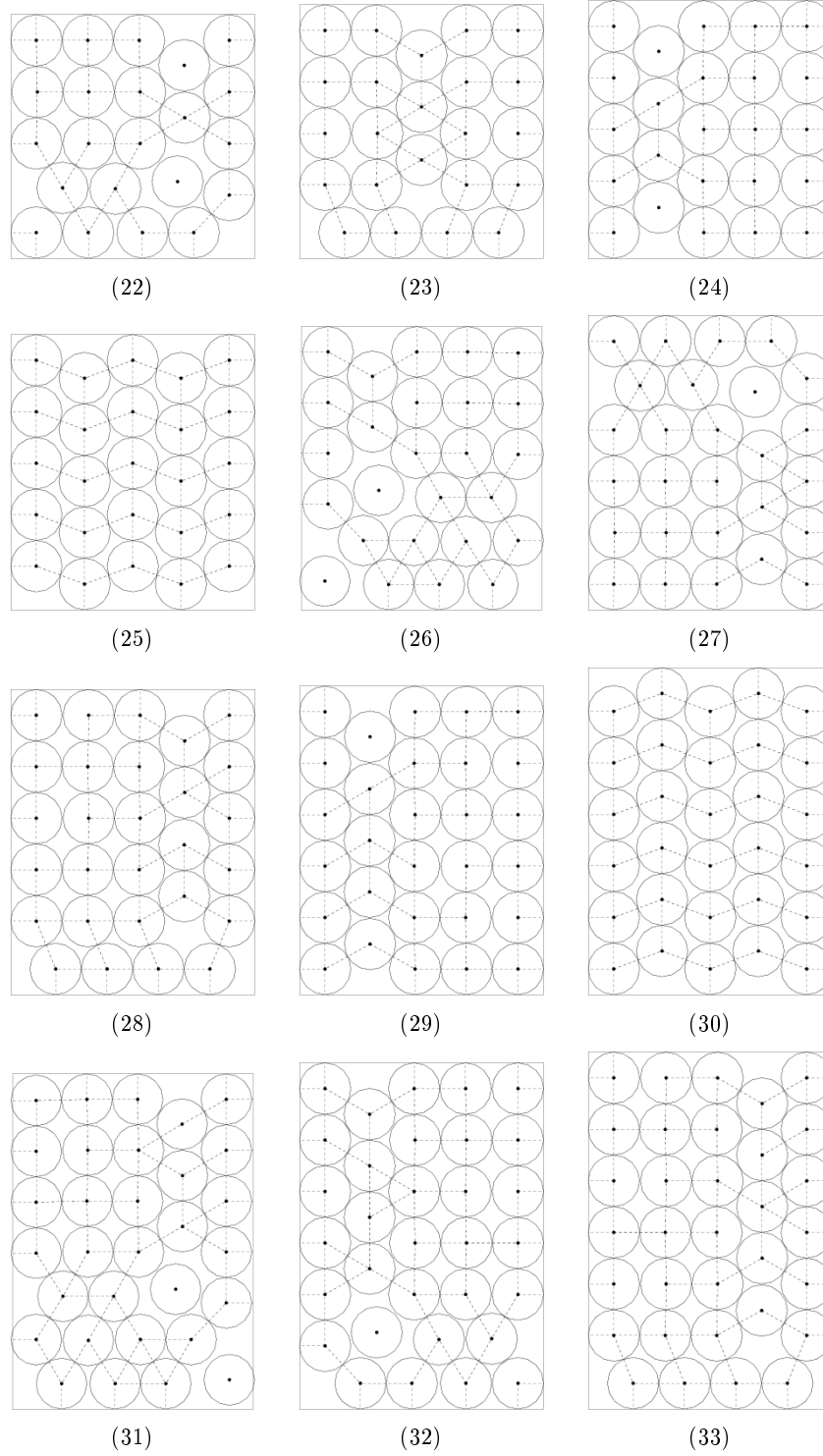


Figura A.8: Representações de empacotamentos de 22 a 33 itens

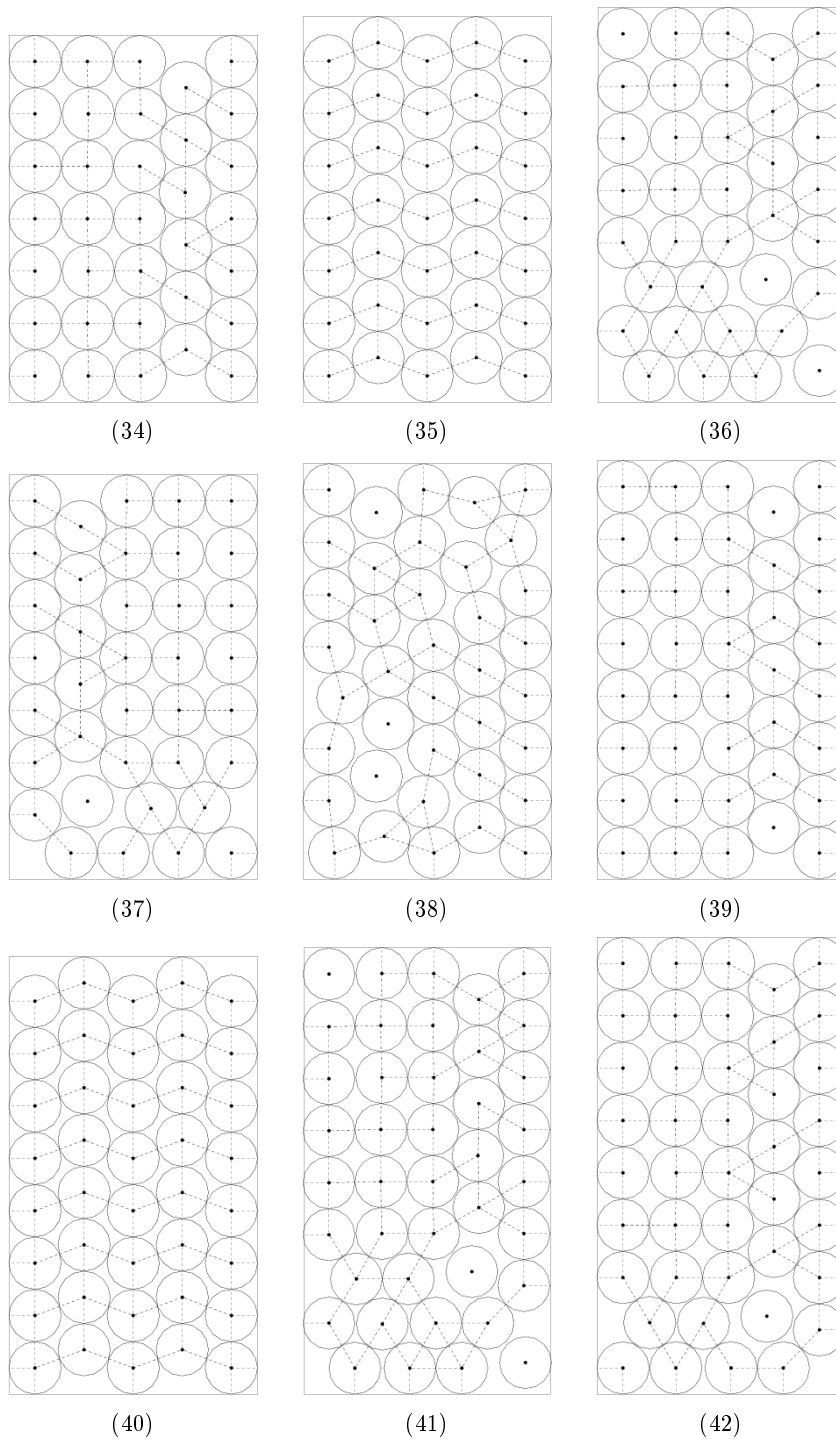


Figura A.9: Representações de empacotamentos de 34 a 42 itens

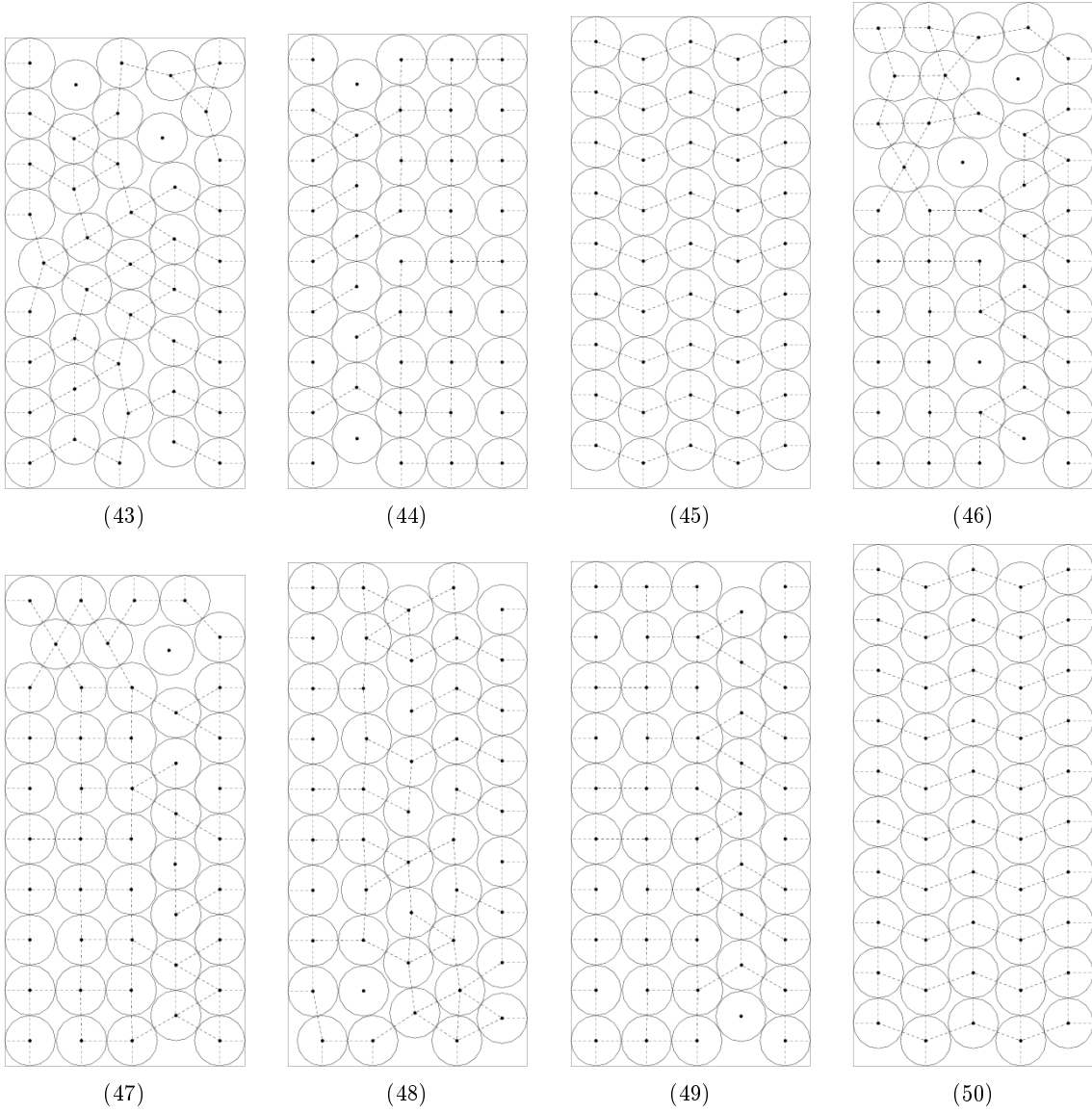


Figura A.10: Representações de empacotamentos de 43 a 50 itens

A.4 Objeto retangular

A.4.1 Minimização de área

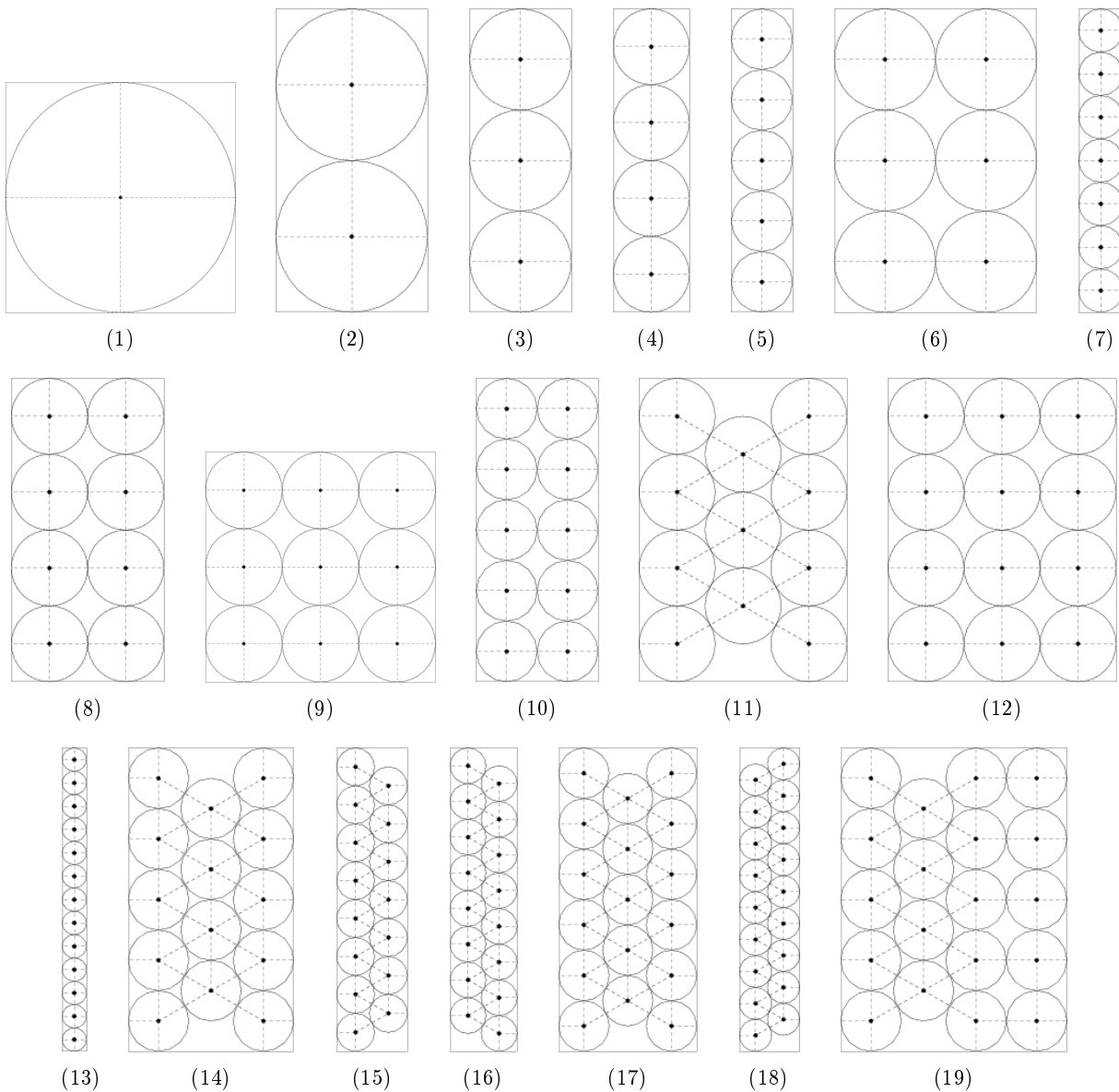


Figura A.11: Representações de empacotamentos de 1 a 19 itens

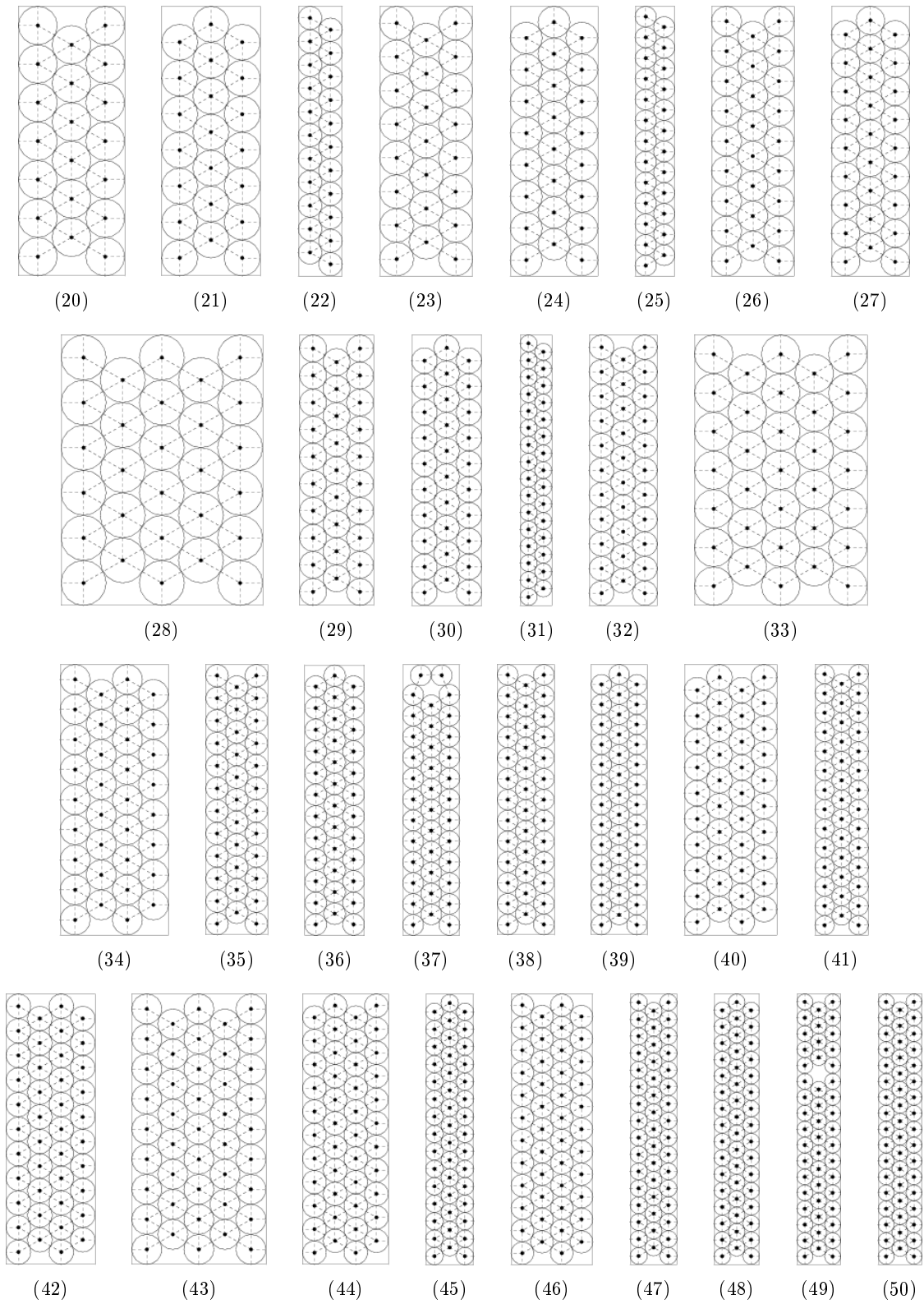


Figura A.12: Representações de empacotamentos de 20 a 50 itens

A.4.2 Minimização de perímetro

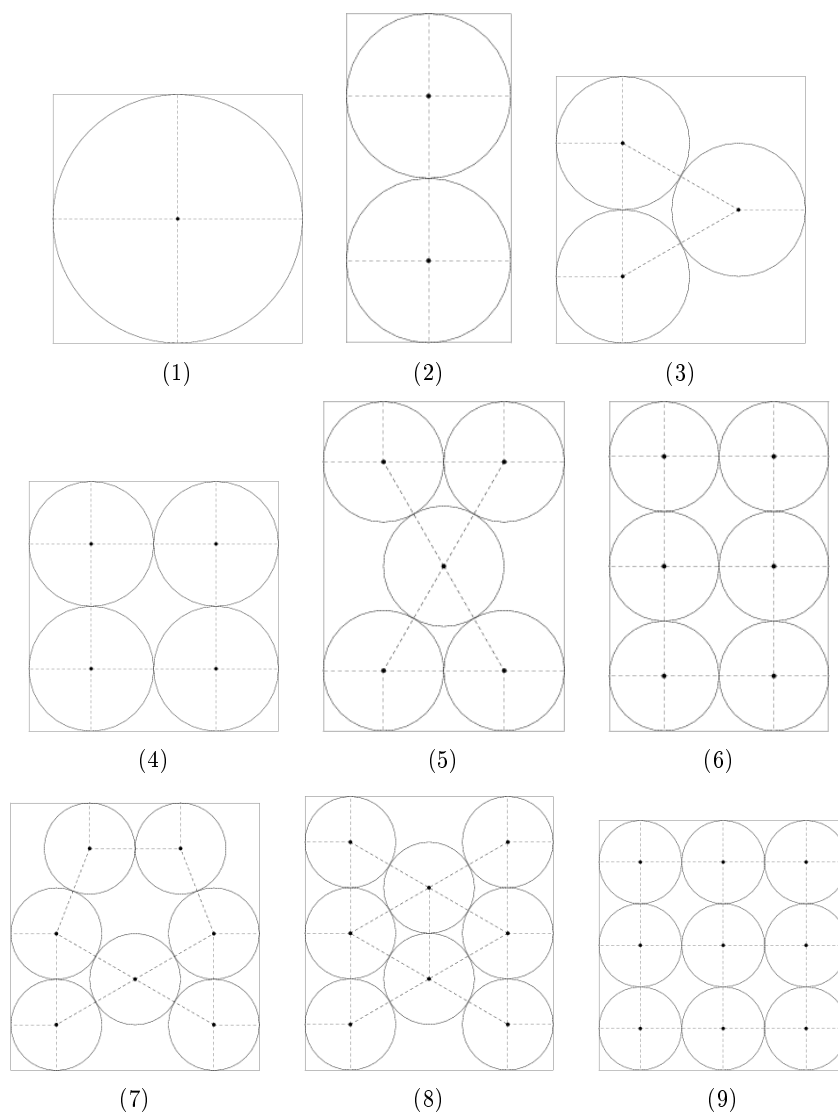


Figura A.13: Representações de empacotamentos de 1 a 9 itens

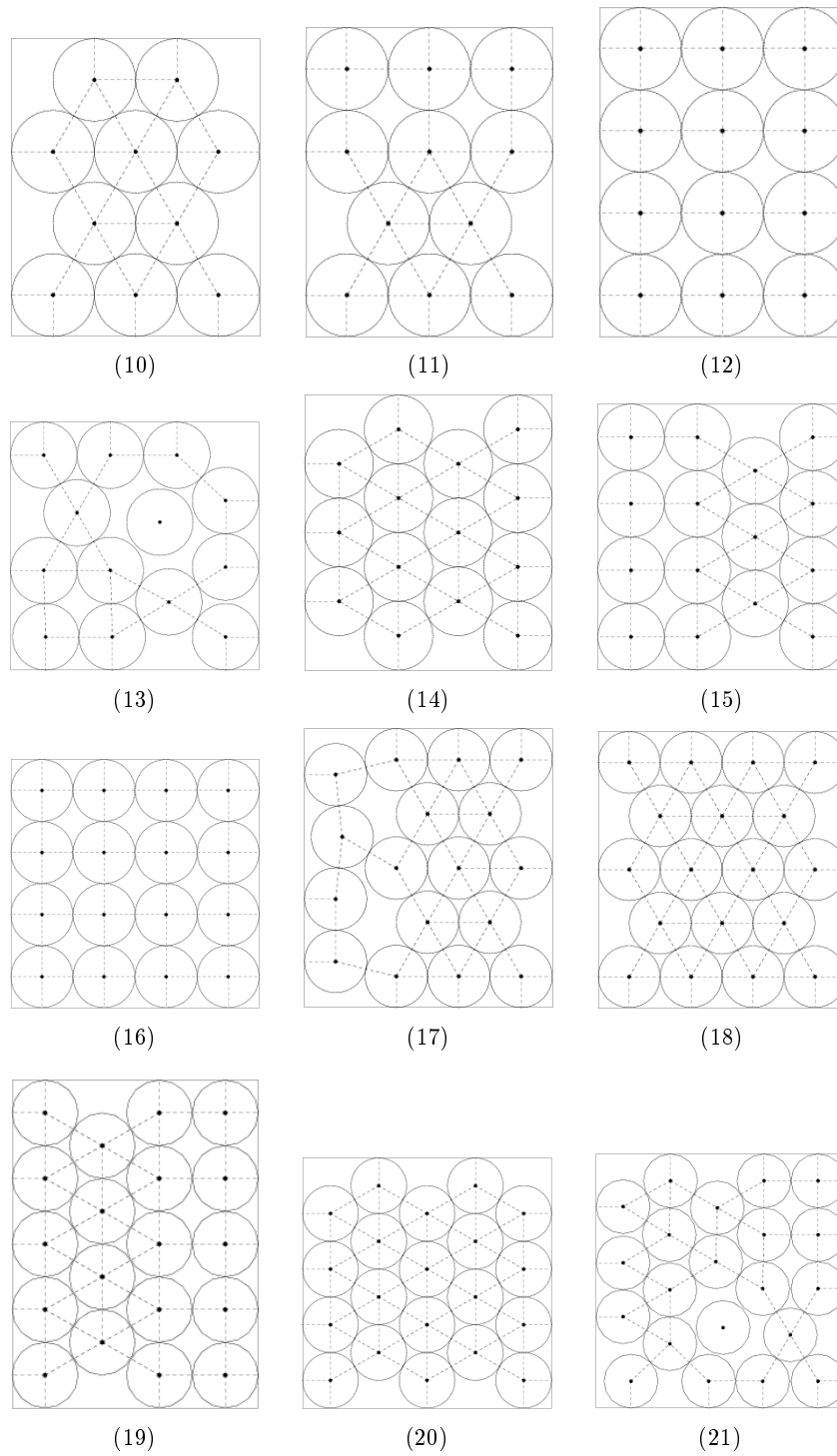


Figura A.14: Representações de empacotamentos de 10 a 21 itens

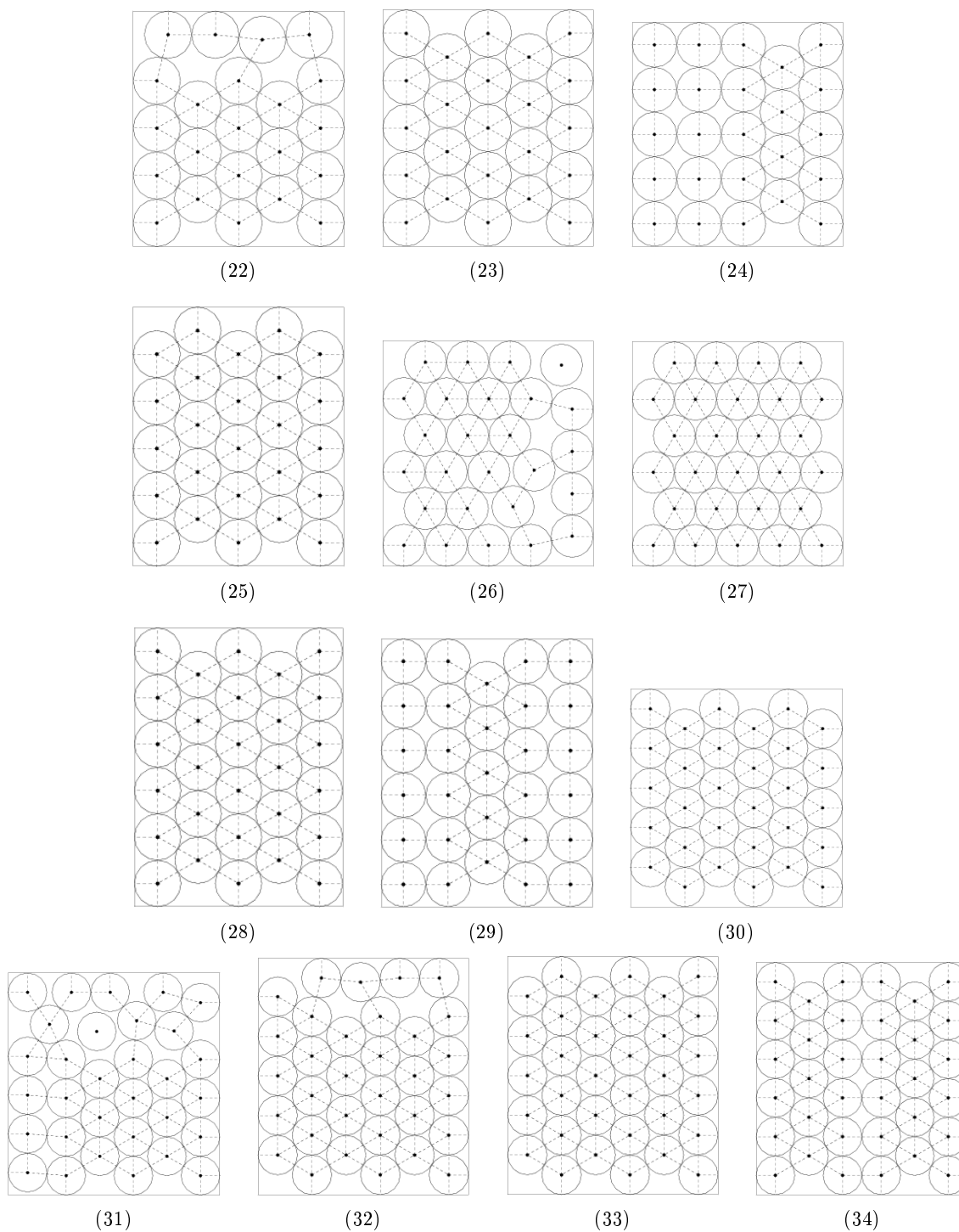


Figura A.15: Representações de empacotamentos de 22 a 34 itens

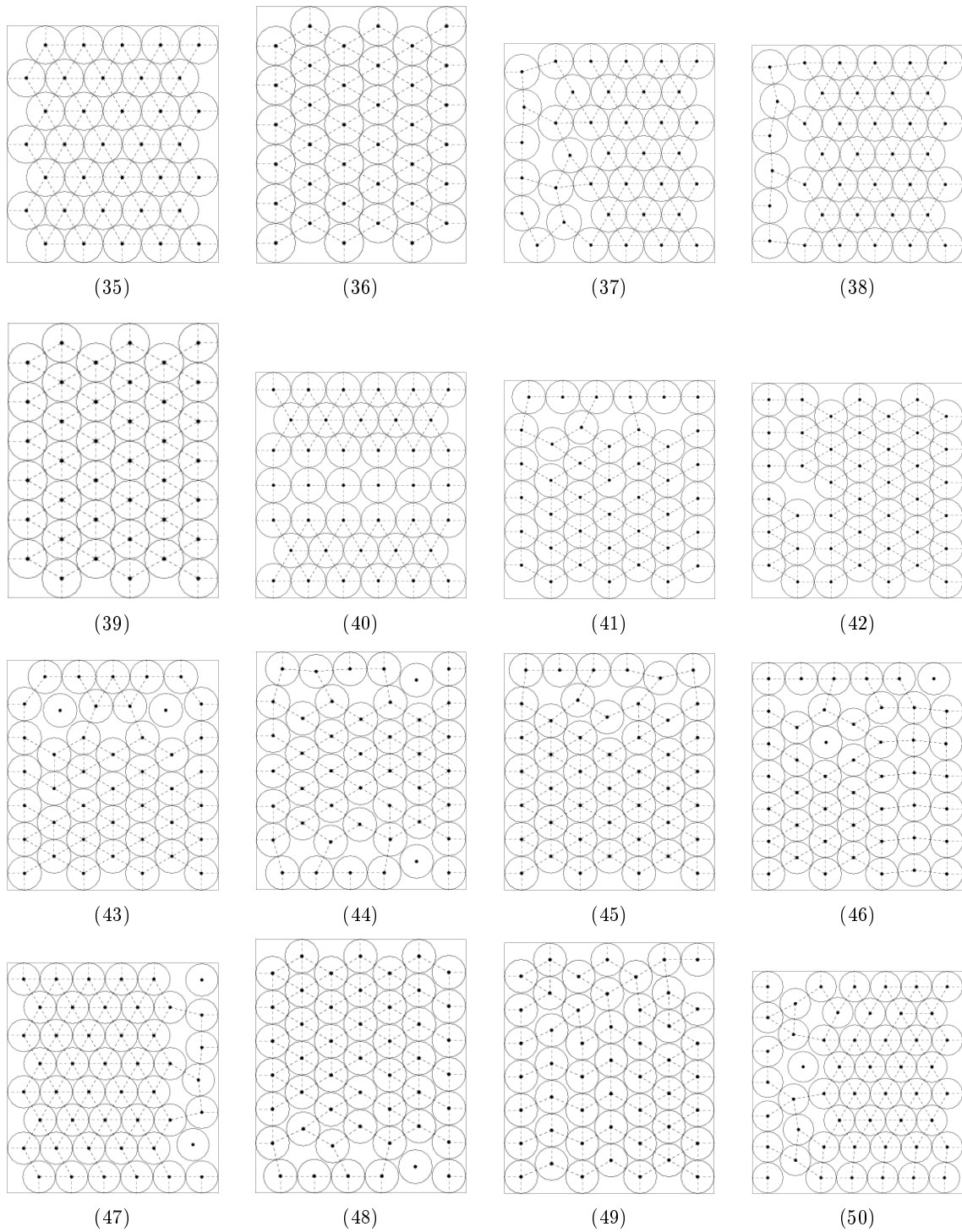


Figura A.16: Representações de empacotamentos de 35 a 50 itens

Parte II

Parte subjetiva

4.5 A Iniciação Científica

Mesmo já nutrindo anteriormente ao meu ingresso no bacharelado em Ciência da Computação um grande interesse pela pesquisa acadêmica, não tinha planos de me lançar em um programa de Iniciação Científica antes do quarto ano de curso, por considerar esse o momento ideal para um projeto de maior complexidade. Felizmente, porém, após ter tido uma excelente experiência na disciplina MÉTODOS NUMÉRICOS DA ÁLGEBRA LINEAR (MAC0300) no quarto semestre, recebi do professor Ernesto G. Birgin a proposta de iniciar um trabalho de IC já no terceiro ano, a qual não hesitei em aceitar.

Muitas foram as vantagens de tal antecipação, todas decorrentes da maior disponibilidade de tempo. Durante os primeiros semestres, foi possível explorar o problema a ser tratado (tema desta monografia) em detalhes, além de estudar cuidadosamente os trabalhos já desenvolvidos e as ferramentas teóricas por eles empregadas. Foi nesse período também em que aprendi a linguagem de programação FORTRAN 77, estranha à maior parte dos alunos do curso, mas extremamente valiosa à área da programação matemática. Inerente a esse aprendizado está ainda a familiarização com o meio acadêmico, como a rotina de um pesquisador, o processo de redação e publicação de artigos, a transmissão do conhecimento adquirido, a produção de código robusto e bem documentado, entre outras faculdades que não poderiam ser de outro modo adquiridas.

Outra preciosa experiência foi a participação nos Seminários de Otimização Contínua, organizado pelo respectivo grupo semanalmente no Instituto. Além de assistir a quase todos, fui motivado pelo meu orientador a também preparar e apresentar um, o que se mostrou tão desafiador quanto enriquecedor. Ainda por incentivo do professor Ernesto, desenvolvi uma interface em PYTHON para o método ALGENCAN, hoje disponível já em sua segunda versão na página do projeto TANGO [5], o que não só me levou ao aprendizado de mais uma moderna e poderosa linguagem de programação, mas principalmente me proporcionou profundo contato com um dos mais sólidos algoritmos de otimização da atualidade. Tal realização foi especialmente satisfatória pelo impacto que surtiu na divulgação e popularização de ALGENCAN, tendo recebido crescente atenção e apreciação por parte da comunidade científica mundial.

4.6 Interação com o orientador

O contato entre meu orientador e mim se dava semanal e pessoalmente, o que facilitou bastante o andamento do projeto e muito contribuiu para minha percepção de responsabilidade. Quando o procurei, sempre estive disposto a me dedicar muitos minutos de sua atenção, uma prática que, tendo em vista todas as suas atribuições, é verdadeiramente admirável.

Além de sua dedicação à Iniciação Científica, também foi sempre muito atencioso quanto à minha graduação, sugerindo disciplinas optativas e aconselhando-me nos mais diversos aspectos. Demonstrou ainda preocupação com meu futuro acadêmico, trabalhando para que eu conseguisse uma bolsa da FAPESP já tendo em vista a facilitação de concessão de apoio financeiro durante o mestrado.

Contudo, uma de suas maiores contribuições à minha vida acadêmica tem sido dada de forma inconsciente, que é a provisão de um modelo no qual tenho procurado me pautar e inspirar.

4.7 Aspectos do curso mais relevantes

É inegável a maior importância de algumas disciplinas que outras à minha formação. Dentre as que merecem destaque estão (em nenhuma ordem em particular):

- PRINCÍPIOS DE DESENVOLVIMENTO DE ALGORITMOS (MAC0122), pela transmissão de tantos princípios fundamentais ao raciocínio algorítmico;
- ESTRUTURAS DE DADOS (MAC0323), pelo aprendizado da representação e manipulação computacional de objetos abstratos;
- LABORATÓRIO DE PROGRAMAÇÃO I (MAC0211), pelo ensino de ferramentas tão úteis no dia a dia de qualquer cientista da computação, bem como pelo desenvolvimento do primeiro projeto de maior porte do curso;
- MÉTODOS NUMÉRICOS DA ÁLGEBRA LINEAR (MAC0300), pela visão e o tratamento computacional de importantes conceitos da Álgebra Linear;
- ALGORITMOS EM GRAFOS (MAC0328), pelo estudo de um dos entes matemáticos de maior aplicabilidade na resolução de problemas pela Ciência da Computação;
- ANÁLISE DE ALGORITMOS (MAC0338), pelo aprofundamento no estudo de problemas recorrentes em Computação e métodos eficientes de resolvê-los, além do ferramental matemático necessário à formalização de tal eficiência;
- OTIMIZAÇÃO COMBINATÓRIA (MAC0325), pela consolidação da teoria aprendida deficientemente por mim em PROGRAMAÇÃO LINEAR;
- ÁLGEBRA I PARA COMPUTAÇÃO (MAT0138), pela apresentação a mecanismos de raciocínio, demonstrações e estruturas matemáticas presentes nos mais diversos campos de estudo da Ciência da Computação.

Seria injusto, entretanto, creditar a importância de tais disciplinas apenas à ementa abordada por cada uma delas. A maior parte do mérito, a meu ver, pertence aos docentes que as lecionaram com maestria, respectivamente Nami Kobayashi, Carlos Eduardo Ferreira, Roberto Hirata Jr., Ernesto Julián Goldberg Birgin, Paulo Feofiloff, Yoshiharu Kohayakawa, Yoshiko Wakabayashi e Vitor de Oliveira Ferreira. Mais que professores, esses educadores não se limitaram à transmissão do conteúdo, mas também influenciaram de forma decisiva a formação acadêmica e pessoal daqueles que souberam aproveitar o privilégio de ser seus alunos. Felizmente, posso me considerar um desses.

4.8 Passos futuros de aprofundamento na área

Tendo sido brindado com uma proveitosíssima experiência na área de Otimização Contínua durante minha Iniciação Científica, planejo agora dar prosseguimento a esse estudo por meio de um programa de mestrado, no próprio IME e sob orientação do mesmo professor, já no próximo ano. Dentre todas as abordagens possíveis nessa extensa área de pesquisa, pretendo seguir um viés mais algorítmico que teórico, simplesmente por uma questão de preferência.

Após alcançar o grau de mestre, cogito a possibilidade de trabalhar pelo de doutor.

Referências Bibliográficas

- [1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed., SIAM, Philadelphia, 1999.
- [2] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt, *Augmented Lagrangian methods under the Constant Positive Linear Dependence constraint qualification*, *Mathematical Programming* **111** (2008), 5–32.
- [3] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt, *On Augmented Lagrangian methods with general lower-level constraints*, *SIAM Journal on Optimization*, to appear, available at <http://www.ime.usp.br/~egbirgin>.
- [4] E. G. Birgin, R. Castillo, and J. M. Martínez, *Numerical comparison of Augmented Lagrangian algorithms for nonconvex problems*, *Computational Optimization and Applications* **31** (2005), 31–56.
- [5] E. G. Birgin and J. M. Martínez, *TANGO Project*, available at <http://www.ime.usp.br/~egbirgin>.
- [6] E. G. Birgin and J. M. Martínez, *Structured minimal-memory inexact quasi-Newton method and secant preconditioners for Augmented Lagrangian Optimization*, *Computational Optimization and Applications*, to appear, available at <http://dx.doi.org/10.1007/s10589-007-9050-z>.
- [7] E. G. Birgin, J. M. Martínez, W. F. Mascarenhas, and D. P. Ronconi, *Method of sentinels for packing items within arbitrary convex regions*, *Journal of the Operational Research Society* **57** (2006), 735–746.
- [8] E. G. Birgin, J. M. Martínez, F. H. Nishihara, and D. P. Ronconi, *Orthogonal packing of rectangular items within arbitrary convex regions by nonlinear optimization*, *Computers & Operations Research* **33** (2006), 3535–3548.
- [9] E. G. Birgin, J. M. Martínez, and D. P. Ronconi, *Optimizing the packing of cylinders into a rectangular container: a nonlinear approach*, *European Journal of Operational Research* **160** (2005), 19–33.
- [10] E. G. Birgin and F. N. C. Sobral, *Minimizing the object dimensions in circle and sphere packing problems*, *Computers & Operations Research* (2006), to appear, available at <http://dx.doi.org/10.1016/j.cor.2006.11.002>.
- [11] A. Brooke, D. Kendrick, and A. Meeraus, *GAMS Release 2.25: A User's Guide*, The Scientific Press, Redwood City, 1992.
- [12] J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, *Classics in Applied Mathematics*, SIAM Publications, Philadelphia, 1996.
- [13] E. Eskow and R. B. Schnabel, *Algorithm 695: Software for a New Modified Cholesky Factorization*, *ACM Transactions on Mathematical Software* **17** (1991), no. 3, 306–312, available at <http://doi.acm.org/10.1145/114697.116806>.
- [14] M. Goldberg, *The packing of equal circles in a square*, *Mathematics Magazine* **43** (1970), 24–30.
- [15] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1996.
- [16] C. Groot, R. Peikert, and D. Würtz, *The optimal packing of ten equal circles in a square*, Technical Report 90-12, ETH, Zürich, 1990.
- [17] M. Locatelli and U. Raber, *Packing equal circles in a square: a deterministic global optimization approach*, *Discrete Applied Mathematics* **122** (2002), 139–166.
- [18] C. D. Maranas, C. A. Floudas, and P. M. Pardalos, *New results in the packing of equal circles in a square*, *Discrete Mathematics* **142** (1995), 287–293.

- [19] W. F. Mascarenhas and E. G. Birgin, *Using sentinels to detect intersections*, submetido, available at <http://www.ime.usp.br/~egbirgin>.
- [20] N. Mladenovic, F. Plastria, and D. Urosevic, *Reformulation descent applied to circle packing problems*, *Computers & Operations Research* **32** (2005), 2419–2434.
- [21] B. A. Murtagh and M. A. Saunders, *MINOS 5.5 User's Guide*, Technical Report 83-20R, Systems Optimization Laboratory, Stanford, 1998.
- [22] K. J. Nurmela and P. R. Östergård, *Packing up to 50 equal circles in a square*, *Discrete & Computational Geometry* **18** (1997), 111–120.
- [23] E. Specht, *Packomania*, available at <http://www.packomania.com>.
- [24] Y. G. Stoyan and G. N. Yas'kov, *Mathematical model and solution method of optimization problem of placement of rectangles and circles taking into account special constraints*, *International Transactions in Operational Research* **5** (1998), 45–57.
- [25] Y. G. Stoyan and G. N. Yas'kov, *A mathematical model and a solution method for the problem of placing various-sized circles into a strip*, *European Journal of Operational Research* **156** (2004), 590–600.