

## 2

### O Problema de Bin Packing

#### 2.1

##### Definição do Problema

Dados um número inteiro positivo  $C$  e um conjunto  $N = \{1, \dots, n\}$  de objetos, cada qual com peso (ou tamanho) inteiro e positivo  $w_i$  ( $i = 1, \dots, n$ ) satisfazendo  $w_i \leq C$ , o problema de empacotamento unidimensional (BP) consiste em encontrar o menor inteiro  $m$  tal que exista uma partição  $B_1, B_2, \dots, B_m$  do conjunto  $N$  em que a soma  $w(B_j) = \sum_{w_i \in B_j}^{j=1, \dots, m} w_i$  dos pesos dos objetos em cada subconjunto  $B_j$  ( $j = 1, \dots, m$ ) não exceda  $C$ . Este problema pode ser visto como o de empacotar  $n$  objetos no menor número de caixas idênticas com capacidade de armazenamento igual a  $C$ . Neste caso, cada subconjunto da partição representa o conteúdo de uma caixa.

BP possui uma relação estreita com o problema de escalonamento em processadores paralelos idênticos, conhecido como  $P||C_{\max}$  segundo a terminologia introduzida por Graham et al. [41]. Neste problema, dados o conjunto  $N$  e um número  $m \geq 2$  de processadores, um escalonamento corresponde a uma partição  $B_1, B_2, \dots, B_m$  do conjunto  $N$  em  $m$  subconjuntos disjuntos, um para cada processador. Cada objeto corresponde a uma tarefa  $i$  com tempo de processamento  $w_i$ , que deverá ser alocada a um dos  $m$  processadores. Da mesma forma, a soma  $w(B_j)$  corresponde ao tempo de processamento do processador  $j$ , considerando-se tarefas independentes e tempo de comunicação inexistentes (cada máquina pode processar uma tarefa por vez e cada tarefa tem que ser processada em exatamente uma única máquina). O tempo de escalonamento é dado pelo maior tempo de processamento entre todas as máquinas (*makespan*). O objetivo é minimizar o tempo de escalonamento ( $\min \{ \max_{1 \leq j \leq m} w(B_j) \}$ ).

Esta relação, chamada de dualidade por diversos autores [11, pág. 283], [12, pág. 190], [68, pág. 21], [13, 47, 48, 49, 62, 77], será utilizada neste trabalho. O acrônimo DBP será utilizado para se referir ao problema *dual*

*bin packing*<sup>1</sup>.

## 2.2

### Revisão Bibliográfica

Nesta seção apresenta-se uma breve descrição dos principais métodos para a resolução do BP: algoritmos exatos, algoritmos de aproximação (aqueles para os quais existe garantia de desempenho) e heurísticas. Em especial, descrevem-se os algoritmos que serão utilizados nos capítulos subseqüentes. Em seguida apresentam-se limites inferiores e o método de redução que serão utilizados pela heurística proposta no próximo capítulo.

#### 2.2.1

##### Algoritmos Exatos

Os primeiros métodos exatos descritos na literatura são os de Eilon e Christofides (1971) [19] e Hung e Brown (1978) [50].

Martello e Toth (1990) [64] apresentam limites inferiores e um critério de dominância. Os limites são avaliados segundo seu desempenho de pior caso. Mostram como um critério de dominância pode ser utilizado para reduzir o tamanho do problema e produzir novos limites. Os autores voltam ao problema BP em [65]. Apresentam um resumo de resultados clássicos de algoritmos de aproximação. Fazem uma revisão do limite  $L_2$ , do critério de dominância MTRP e do limite, mais forte,  $L_3$  apresentados na referência anterior e que serão explicados na Seção 2.2.4. Apresentam o procedimento de *branch and bound* MTP baseado em um esquema de *branching* que utiliza a mesma estratégia, orientada por itens, do algoritmo FFD (descrito na próxima seção). De forma genérica, MTP pode ser considerado um método exato. Entretanto, para evitar tempos de processamento inaceitáveis, o número de *backtrackings* ou o tempo computacional pode ser limitado. Desta forma, MTP torna-se uma heurística. Um experimento computacional avalia o desempenho dos limites inferiores e do método exato propostos. É importante salientar, ainda hoje, a importância do trabalho de Martello e Toth como referência básica, por diversas razões: a qualidade dos limites inferiores  $L_2$  e  $L_3$ , do critério de dominância e do método de redução MTRP que podem ser usados em procedimentos exatos e heurísticas, a qualidade

---

<sup>1</sup>A expressão “dual” também é usada por outros autores em referência ao problema de empacotar objetos no maior número possível de caixas de forma que o peso de cada caixa seja pelo menos igual à sua capacidade [58].

do método exato MTP, analisada em diversos estudos comparativos [77, 81], além da disponibilidade do seu código.

Scholl, Klein e Jürgens (1997) [77] propõem um procedimento híbrido (BISON) para a solução exata de BP que combina limites inferiores e procedimentos de redução tradicionais e novos, diversas heurísticas e um procedimento de *branch and bound* com novo esquema de *branching*. Além dos limites tradicionais  $L_2$  e  $L_3$  de Martello e Toth, usam a relação de BP com outros problemas de otimização combinatória (roteamento de veículos com restrição de capacidade, particionamento, escalonamento em processadores paralelos, balanceamento em linhas de montagem) para introduzir novos limites, por eles denominados de  $L_4$ ,  $L_5$  e  $L_6$ . Entre as heurísticas propostas, a que utiliza a metaheurística busca tabu é a mais eficiente. Esta usa uma estratégia que os autores chamam de *dual*. Tentam encontrar uma solução para o problema de escalonamento em processadores paralelos ( $P||C_{\max}$ ) em que o número de processadores seja igual ao melhor limite inferior conhecido para BP, obedecendo a restrição de capacidade da caixa. Enquanto tal solução não existir e o limite inferior no número de caixas (processadores) for menor do que o limite superior menos um, o limite inferior é incrementado e o processo é repetido. O procedimento BISON pode ser resumido em seis passos: i) aplicar o procedimento de redução MTRP; ii) computar os limites inferiores  $L_2$ ,  $L_4$  e  $L_5$ ; iii) aplicar as heurísticas FFD, B2F, BFD, WFD (explicados na próxima seção) e uma nova heurística chamada de FFD-B2F; iv) computar os limites  $L_3$  e  $L_6$ ; v) aplicar a estratégia *dual* de busca tabu para melhorar o limite superior no número de caixas e, por fim, vi) aplicar um algoritmo de *branch and bound* que, diferentemente de MTP, é orientado por caixas. Em cada passo, se a solução corrente for igual ao melhor limite inferior conhecido, o procedimento pára. Outra contribuição deste trabalho é a geração de três conjuntos de problemas testes, totalizando 1210 instâncias, disponíveis em <http://www.bwl.tu-darmstadt.de/bwl3/>. Os autores fazem um levantamento dos métodos de aproximação e dos métodos exatos existentes e apresentam um experimento computacional que incluiu a análise da eficácia do procedimento de redução MTRP de Martello e Toth, a análise dos limites inferiores, a análise das diferentes heurísticas usadas como componentes de BISON e a comparação do método exato MTP em relação ao método proposto BISON.

Schwerin e Wäscher (1997) [80] introduzem um gerador de problemas e novas classes de problemas testes para BP. As instâncias são caracterizadas por quatro parâmetros: capacidade da caixa, tamanho do problema e limite inferior e superior no peso dos objetos. O número total de combinações des-

tes parâmetros resulta em 440 diferentes classes. Estas foram qualificadas de acordo com a percentagem  $p$  de instâncias resolvidas de forma ótima pela heurística FFD. Assim, uma classe é classificada como **ffd-fácil** quando  $p$  maior ou igual a 80%, uma classe é **ffd-difícil** quando  $p$  maior ou igual a 20% e menor do que 80%, e uma classe é **ffd-extremamente-difícil** quando  $p$  menor do que 20%. O limite inferior  $L_{CS}$  derivado para o valor ótimo da função objetivo de 1D-CSP (*Integer One-Dimensional Cutting Stock Problem*), equivalente a BP [46], é utilizado para provar a otimalidade das soluções de BP. Além disso, os autores fazem diversos experimentos numéricos com MTP e FFD e concluem que MTP obtém bons resultados para determinadas classes de problemas, nas quais FFD consegue resolver poucas instâncias de forma exata. Posteriormente [81], voltam ao limite  $L_{CS}$ , avaliando sua aplicabilidade para instâncias BP em extensivos experimentos numéricos. Primeiro mostram a superioridade deste limite em relação ao limite  $L_3$ . Em 44.000 instâncias, em apenas um caso o limite  $L_3$  é superior. Para as demais instâncias,  $L_{CS}$  é pelo menos tão bom quanto  $L_3$ . Em seguida estudam o efeito da integração do limite  $L_{CS}$  no procedimento MTP, resultando no procedimento modificado MTPCS, e comparam, para um conjunto de 8.100 instâncias, os resultados obtidos por este procedimento com os resultados obtidos pelo MTP tradicional. Com poucas exceções, a integração de  $L_{CS}$  melhora a qualidade da solução do MTP. Outro experimento avalia o procedimento modificado MTPCS em comparação a BISON. Os resultados indicam que a superioridade de um método em relação ao outro é fortemente dependente da classe de problemas. Além disso, para uma determinada classe de problemas, com dez instâncias, o limite  $L_{CS}$  é mais forte do que o limite  $L_3$ . Se  $L_{CS}$  fosse integrado em BISON, todas as instâncias desta classe, e não apenas três, teriam sido identificadas por BISON como ótimas. Uma desvantagem do limite inferior  $L_{CS}$  é seu custo computacional elevado.

Vance et al. (1994) [86], Vance (1998) [85], Vanderbeck (1999) [87] e Carvalho (1999) [84] apresentam algoritmos exatos para o problema de *cutting stock*, combinando geração de colunas e *branch and bound*. Outro método exato de *branch and bound* para o BP é apresentado por Scheithauer e Terno (1995) [74]. Scheithauer et al. (2001) [75] apresentam outra abordagem exata para o *cutting stock* baseada em métodos de *cutting plane*.

Elita et al. [69] propõem dois algoritmos para o problema de *cutting stock* e de *bin packing*: um método exato de *branch-and-bound* modificado (MBB) e uma heurística baseada em programação linear (*Sequential Value*

*Correction* - SVC). Os experimentos computacionais são realizados com todas as classes de problemas descritas em [80] e os resultados obtidos comparados com os apresentados em [81].

## 2.2.2 Algoritmos de Aproximação

Nesta seção faz-se um resumo de alguns algoritmos de aproximação para BP e de seu desempenho assintótico de pior caso, segundo Coffman et al. [12, 14].

Sejam  $P$  um problema de minimização e  $A$  uma heurística para  $P$ . Para qualquer instância  $I$  do problema  $P$ ,  $z(I)$  e  $A(I)$  denotam, respectivamente, o valor de sua solução ótima e o valor da solução obtida pela heurística  $A$ . A razão de desempenho de pior caso absoluto da heurística  $A$  é definida como

$$r(A) = \sup \left\{ \frac{A(I)}{z(I)} : \text{para todas as instâncias } I \right\}.$$

Para alguns problemas, e para o problema de *bin packing* em especial [12, 32, 65], o pior caso absoluto é uma informação muito pessimista sobre o comportamento de pior caso do algoritmo. Nestes casos, a medida padrão é sua razão de desempenho de pior caso assintótico, isto é, o limite do desempenho de pior caso quando o tamanho da instância cresce. A razão de desempenho de pior caso assintótico, daqui em diante chamada de razão de pior caso, é definida como

$$r^\infty(A) = \lim_{z(I) \rightarrow \infty} \sup \left\{ \frac{A(I)}{z(I)} : \text{para todas as instâncias } I \right\}.$$

Esta medida de qualidade diz que o quão mais próximo  $r^\infty(A)$  for de 1, tão melhor será o algoritmo  $A$ .

Um algoritmo *on-line* para BP é aquele em que os objetos chegam em alguma ordem e devem ser empacotados tão logo possível, sem o conhecimento dos demais objetos. A seguir descreve-se quatro algoritmos *on-line* para BP. Assume-se que as caixas estão indexadas  $1, 2, 3, \dots$  na ordem em que foram criadas, isto é, em que receberam o primeiro objeto.

- *Next-Fit* (NF): o primeiro objeto é colocado na caixa de índice 1. Cada um dos demais objetos é colocado na caixa onde foi colocado o objeto anterior, se houver espaço suficiente; caso contrário o objeto é

colocado em uma nova caixa. A complexidade deste algoritmo é  $O(n)$  e sua razão de pior caso, cf. [12], é  $r^\infty(\text{NF}) = 2$ .

- *First-Fit* (FF): diferentemente do método anterior, ao se empacotar um objeto, considera-se todas as caixas parcialmente cheias. Coloca-se o objeto sendo considerado na caixa de menor índice onde ele cabe. Somente quando um objeto não puder ser incluído em uma caixa já usada, uma nova caixa será utilizada. Este pode ser implementado com complexidade  $O(n \log n)$  [51]. Sua razão de pior caso [52] é  $r^\infty(\text{FF}) = 17/10$ .
- *Best-Fit* (BF): coloca o objeto corrente na caixa mais cheia onde ele cabe, abrindo uma nova caixa se tal caixa não existir. O desempate é feito escolhendo-se a caixa de menor índice. Esta estratégia consegue razão de pior caso tão boa quanto a de FF e tem a mesma complexidade  $O(n \log n)$  [52].
- *Worst-Fit* (WF): em oposição à anterior, coloca o objeto corrente na caixa menos cheia onde ele cabe. Em caso de empate escolhe a caixa de menor índice. Se não existir tal caixa, abre uma nova. Sua complexidade é  $O(n \log n)$  e sua razão de pior caso [51] é  $r^\infty(\text{WF}) = r^\infty(\text{NF})$ .

Johnson (1974) [51] questiona se existe algoritmo *on-line* melhor do que FF. Em resposta afirmativa, Yao (1980) [89] apresenta o algoritmo *Refined First-Fit* (RFF) e prova que sua razão de pior caso é  $r^\infty(\text{RFF}) = 5/3$ . Lee e Lee (1985) [61] apresentam o algoritmo *Harmonic-Fit* e provam que sua razão de pior caso é  $r^\infty(\text{HF}) = 1.69103\dots$

Relaxando-se a restrição de processamento *on-line*, é possível melhorar o comportamento de pior caso obtido por *First-Fit*. Tipicamente, o pior caso ocorre quando muitos objetos com peso pequeno são colocados nas primeiras caixas. Ao invés de colocar os objetos na ordem em que eles são apresentados, eles são ordenados em ordem decrescente de peso e posteriormente aplica-se uma regra de empacotamento. Quando a regra de empacotamento é a de NF o algoritmo resultante é conhecido como *Next-Fit Decreasing* (NFD) cuja razão de desempenho de pior caso é  $r^\infty(\text{NFD}) = 1.69103\dots$ . Os algoritmos *First-Fit Decreasing* (FFD) e *Best-Fit Decreasing* (BFD) são definidos de forma análoga. O comportamento de pior caso [52] melhora significativamente para  $r^\infty(\text{FFD}) = r^\infty(\text{BFD}) = 11/9$ . Considerando-se o passo de ordenação, as três heurísticas possuem a mesma complexidade  $O(n \log n)$ .

Outras estratégias foram desenvolvidas, a maioria delas extensões das apresentadas acima. A primeira melhoria em relação a FFD é apresentada por Yao (1980) [89], que mostra um algoritmo chamado *Refined FFD* (RFFD) de complexidade  $O(n^{10} \log n)$  e razão de pior caso  $r^\infty(\text{RFFD}) \leq \frac{11}{9} - 10^{-7}$ . Garey e Johnson (1985) propõem o algoritmo *Modified First-Fit Decreasing* (MFFD) [53]. Seja  $S$  o conjunto de objetos de peso  $w_i \in (\frac{1}{6}, \frac{1}{3}]$ . A estratégia de seleção da caixa é a mesma que a de FFD até o momento em que os objetos de  $S$  são considerados. A partir deste ponto, MFFD considera, em ordem decrescente de folga da caixa, as caixas contendo um único objeto de peso maior do que a metade da capacidade da caixa. Para a caixa corrente, primeiro verifica se os dois objetos mais leves de  $S$  cabem nessa caixa. Se for o caso, seleciona o objeto mais leve juntamente com o mais pesado de  $S$  que cabem na caixa corrente, empacotando-os na mesma. Do contrário, esta fase especial acaba e os objetos restantes são colocados segundo a regra de FFD. Esta versão modificada de FFD possui essencialmente a mesma complexidade  $O(n \log n)$ , seu comportamento médio é semelhante e seu comportamento assintótico de pior caso melhora para  $r^\infty(\text{MFFD}) = 71/60$ .

Friensen e Langston (1991) propõem outra modificação de FFD conhecida como *Best Two Fit* (B2F) [31]. O algoritmo FFD pode construir empacotamentos considerando uma caixa de cada vez usando a seguinte regra: enquanto existir espaço suficiente, o objeto livre, considerado em ordem não crescente de peso, é colocado na caixa corrente. B2F também considera uma caixa de cada vez e aplica a mesma regra de FFD até não ser possível colocar mais objetos na caixa. Neste ponto, B2F verifica se a caixa possui dois objetos ou mais, e se o menor objeto pode ser trocado pelos dois menores objetos livres de peso maior ou igual a  $1/6$ . Se este for o caso, o menor objeto é substituído pelo par de objetos que some o maior valor e caiba na caixa. Este processo continua até o ponto em que não existem mais objetos de peso maior ou igual a  $1/6$ , quando então aplica-se a regra de FFD. Friensen e Langston mostram que B2F tem comportamento de pior caso  $r^\infty(\text{B2F}) = 5/4$ , que não é melhor do que o que pode obter FFD. Entretanto, eles mostram mais tarde que as instâncias de pior caso dos dois algoritmos são complementares. Chamando-se de CFB o algoritmo composto que executa FFD e B2F e retorna o melhor dos dois resultados, este obtém melhores resultados do que os resultados obtidos executando-se cada um dos dois algoritmos separadamente, apenas dobrando o tempo de execução. Mais precisamente, seu comportamento de pior caso é  $227/195 \leq r^\infty(\text{CFB}) \leq 6/5$ .

Kenyon (1996) [56] propôs a variante *Best Fit Randomized* (BFR) de

*Best Fit Decreasing*. Assim como BFD, este algoritmo aplica a regra de empacotamento *Best Fit* após alterar a ordem dos objetos. Ao invés de ordená-los, porém, faz uma permutação randômica dos objetos. Desta forma, para um dado conjunto de objetos, BFR produz uma distribuição randômica de empacotamentos. Para se avaliar a medida de eficiência deste algoritmo utiliza-se o número “esperado” de caixas  $E[A(I)]$ . O comportamento de pior caso assintótico é definido como

$$r^\infty(A) = \limsup_{z(I) \rightarrow \infty} \frac{E[A(I)]}{z(I)}.$$

A razão esperada, analiticamente determinada, satisfaz  $1.08 \leq r^\infty(\text{BFR}) \leq 1.5$ , que é pior do que a razão de MFFD. Entretanto, até o presente, não existe exemplo tal que a razão seja maior do que 1.144. Conjectura-se que  $r^\infty(\text{BFR}) < 1.15$ .

Diz-se que um algoritmo é uma  $\epsilon$ -aproximação para um problema de minimização quando ele produz soluções cujo valor não é maior do que  $(1 + \epsilon)$  vezes o valor ótimo. Um esquema de aproximação é uma família de algoritmos  $A_\epsilon$ , tal que para cada  $\epsilon > 0$  o algoritmo  $A_\epsilon$  é uma  $\epsilon$ -aproximação. Yao (1980) [89] questiona se, para um algoritmo de complexidade  $O(n)$ , existe um limite para o valor de  $\epsilon$  tal que  $A_\epsilon$  é um esquema de aproximação. Fernandez de la Vega e Lueker (1981) [23] respondem esta questão de forma negativa, apresentando um esquema de aproximação assintótico que garante uma solução que é uma  $\epsilon$ -aproximação assintótica para qualquer valor fixo de  $\epsilon$ , com complexidade  $c_\epsilon + cn \log(1/\epsilon)$ , onde  $c_\epsilon$  depende somente de  $\epsilon$  e  $c$  é uma constante. Apesar do tempo de execução ser linear em função de  $n$ , é exponencial em função de  $1/\epsilon$ . Karmarkar e Karp (1982) [54] apresentaram um algoritmo que é uma  $\epsilon$ -aproximação assintótica com tempo de execução polinomial em  $1/\epsilon$ . No entanto, sua complexidade é pior do que  $O(n^8)$ . Estes resultados possuem apenas valor teórico, pois o tempo necessário para garantir resultados com até  $(1 + \epsilon)$  vezes o valor ótimo aumenta muito rápido a medida que  $\epsilon$  se aproxima de zero.

Na prática, FFD e BFD são as heurísticas mais conhecidas para a solução aproximada do BP.

### 2.2.3 Outras Heurísticas

Hübscher e Glover (1994) [49] apresentam uma abordagem de busca tabu para o problema de escalonamento de tarefas em processadores parale-



los idênticos ( $P||C_{\max}$ ) aplicada a instâncias de BP. Os autores utilizam uma estratégia de lista restrita de candidatos que gera um subconjunto pequeno de possíveis movimentos. A transição de uma solução para outra é feita com um movimento simples de troca de dois objetos de duas caixas. Para controlar os movimentos tabu empregam uma lista tabu dinâmica com estratégias de diversificação e intensificação. Os resultados foram comparados com aqueles obtidos por um método de *simulated annealing*.

Problemas de agrupamento consistem, basicamente, em criar partições de conjunto de forma a obter uma coleção de subconjuntos disjuntos. BP faz parte desta família de problemas. O algoritmo genético de agrupamento *Grouping Genetic Algorithm* (GGA) é um paradigma de algoritmo genético para tratar este tipo de problema. O algoritmo híbrido *Hybrid Grouping Genetic Algorithm* (HGGA) para BP de Falkenauer (1996) [20] combina GGA com uma otimização local inspirada no critério de dominância de MT. Falkenauer compara seu HGGA com o procedimento MTP. Outra contribuição de Falkenauer é a geração de dois conjuntos de problemas testes, totalizando 160 instâncias, disponíveis no banco de dados de *benchmarks* OR-Library mantido por John Beasley [6]. A dificuldade destas instâncias é contestada por Gent [33] que, assim como Valério de Carvalho [84], encontra a solução ótima das instâncias consideradas em aberto por Falkenauer. Estas instâncias têm sido utilizadas como *benchmarks*, por exemplo por Fleszar e Hindi [25] e nesta tese.

Soma et al. (1999) [83] apresentam uma heurística pseudopolinomial orientada por caixas para BP, baseada em múltiplas soluções para o problema de soma de subconjuntos (SSP). Inicialmente,  $k$  permutações do conjunto de objetos são geradas. Para cada uma das  $k$  permutações, o conjunto de objetos é dividido em dois subconjuntos: o conjunto  $L$  com objetos de peso maior ou igual à metade da capacidade da caixa e o conjunto  $S$  com os objetos restantes. A cada tentativa de encher uma caixa e até não existirem mais objetos livres, fixa-se o primeiro objeto livre de  $L$  na caixa corrente e resolve-se o problema de soma de conjuntos considerando-se o conjunto de objetos  $S$  e a capacidade residual da caixa corrente. Os autores usam as instâncias apresentadas em [80] para verificar a eficiência do seu algoritmo e comparam os resultados obtidos com MTP e com os dos experimentos relatados em [80].

Outra heurística orientada por caixas é proposta por Gupta e Ho (1999) [43], chamada de *Minimal Bin Slack* (MBS). Esta considera uma caixa por vez, e nela coloca o conjunto de objetos cuja soma de peso mais se aproxima da capacidade da caixa. Os autores comparam a eficácia de MBS

em relação a BFD e FFD. Eles mostram a superioridade do algoritmo, em especial para instâncias cuja solução ótima necessita da maioria das caixas completas.

Mais recentemente, Fleszar e Hindi (2002) [25] propõem novas heurísticas para BP. A maioria delas baseadas em uma variante da heurística MBS de Gupta e Ho [43]. A diferença da heurística proposta modificada MBS', que serve de base para três outras heurísticas (*Relaxed MBS'*, *Perturbation MBS'* e *Sampling MBS'*), está na fixação do objeto livre mais pesado a cada tentativa de completar uma caixa. Embora MBS' não domine MBS em termos de qualidade da solução, os autores apresentam experimentos computacionais que mostram, na prática, a eficácia de MBS'. O mais efetivo dos algoritmos usa a variante *Perturbation MBS'* para gerar uma solução inicial seguido de busca local baseada na metaheurística VNS [44], juntamente com novos limites inferiores propostos por Fekete e Schepers [22]. Fleszar e Hindi comparam seu trabalho com BISON e com o algoritmo de Valério de Carvalho, usando as instâncias propostas por Falkenauer e Scholl.

#### 2.2.4 Limites e Reduções

Uma forma de se avaliar a qualidade de uma solução obtida por uma heurística para um problema de minimização é através de bons limites inferiores (ou superiores, no caso de problemas de maximização) para a instância em questão. Tipicamente, isso pode envolver o uso de algum tipo de relaxação do problema, de tal forma que o novo problema seja mais fácil de ser resolvido [72]. Nesta seção descreve-se os limites inferiores que serviram de base para a heurística proposta no Capítulo 3. Apresenta-se um novo limite inferior para BP. Uma boa referência para outros limites inferiores do BP é [77]. Nem todos os limites apresentados neste trabalho foram incorporados e investigados pela heurística proposta.

Dado um procedimento  $L$  para calcular o limite inferior do problema  $P$ , sejam  $L(I)$  e  $z(I)$ , respectivamente, o valor produzido pelo procedimento  $L$  e o valor da solução ótima para a instância  $I$ . A razão de desempenho de pior caso [65] do limite inferior  $L$  é definida como

$$R(L) = \inf \left\{ \frac{L(I)}{z(I)} : I \text{ é uma instância de } P \right\}.$$

Não havendo confusão, pode-se denotar por  $L$  o procedimento que calcula o limite inferior e o valor que ele produz para uma determinada

instância de um problema, tendo  $z(I)$  como o valor da solução ótima correspondente.

Supõe-se, daqui em diante, que os objetos estarão ordenados de forma que:

$$w_1 \geq w_2 \geq \dots \geq w_n. \quad (2-1)$$

### 2.2.4.1

#### Limite Inferior $L_1$

Este limite é baseado na relaxação da propriedade de indivisibilidade de um objeto, isto é, permite-se que um objeto seja dividido e empacotado em diversas caixas:

$$L_1 = \lceil \sum_{i=1}^n w_i / C \rceil.$$

O limite  $L_1$  equivale ao menor espaço necessário para acomodar todos os objetos e pode ser muito pobre para instâncias com poucos objetos pequenos em relação à capacidade da caixa. Isto acontece, por exemplo, se todos os objetos possuem peso igual a  $w_i = \lfloor C/2 + 1 \rfloor$ . Neste caso, o limite inferior se aproxima de  $1/2$  à medida que  $C$  cresce. A razão de desempenho de pior caso de  $L_1$  é  $R(L_1) = 1/2$ . Para problemas em que os pesos são suficientemente pequenos em relação à capacidade da caixa, espera-se um bom comportamento médio de  $L_1$ , uma vez que nestes casos a estimativa do valor ótimo oferecida pelo limite inferior não é muito afetada pela relaxação da restrição de integralidade.

### 2.2.4.2

#### Limite Inferior $L_2$ de Martello e Toth

Martello e Toth [64] propõem o limite  $L_2$ , mais forte, para problemas cujos pesos dos objetos são maiores e onde poucos objetos cabem em uma mesma caixa. Este limite é baseado em criar uma partição do conjunto de objetos em três subconjuntos, de acordo com o parâmetro inteiro  $0 \leq \alpha \leq C/2$ :

$$\begin{aligned} I_1 &= \{i \in N : w_i > C - \alpha\}, \\ I_2 &= \{i \in N : C - \alpha \geq w_i > C/2\}, \\ I_3 &= \{i \in N : C/2 \geq w_i \geq \alpha\}. \end{aligned}$$

O limite inferior  $L(\alpha)$  no número de caixas necessárias para empacotar os objetos da instância relaxada (desconsiderando-se os objetos de peso menor do que  $\alpha$ ) é calculado da seguinte forma:

$$L(\alpha) = |I_1| + |I_2| + \max \left\{ 0, \left\lceil \frac{\sum_{i \in I_3} w_i - (|I_2| C - \sum_{i \in I_2} w_i)}{C} \right\rceil \right\}, \quad (2-2)$$

onde  $|I_i|$  denota a cardinalidade do conjunto  $I_i$ . Os objetos em  $I_1$  e  $I_2$  não podem ser combinados com nenhum outro objeto nestes conjuntos, pois possuem peso maior do que  $C/2$ . Desta forma, o número de objetos nos dois conjuntos é um limite inferior no número de caixas. Objetos em  $I_3$  só podem ser combinados com objetos em  $I_2$ . A capacidade residual das  $|I_2|$  caixas reservadas para os objetos em  $I_2$  é  $\overline{C} = |I_2| C - \sum_{i \in I_2} w_i$ . Na melhor das hipóteses, os objetos de  $I_3$  caberiam em  $\overline{C}$ . Se este não for o caso, utiliza-se o limite  $L_1$  para calcular o número de caixas necessárias para empacotar os objetos restantes, cujo peso é  $\sum_{i \in I_3} w_i - \overline{C}$ .

O limite inferior  $L_2$  é dado pelo maior valor de  $L(\alpha)$ , para  $0 \leq \alpha \leq C/2$ :

$$L_2 = \max\{L(\alpha) : 0 \leq \alpha \leq C/2\}.$$

Assumindo que os pesos estão ordenados segundo (2-1), Martello e Toth [64] provam que este limite pode ser calculado considerando-se sucessivamente apenas valores distintos de  $w_i \leq C/2$  e 0 como valores do parâmetro  $\alpha$ . Assumindo que existe algum objeto com peso  $w_i \leq C/2$ , deve-se provar que, dados  $\alpha_1 < \alpha_2$ , se  $\alpha_1$  e  $\alpha_2$  produzem o mesmo conjunto  $I_3$ , então  $L(\alpha_1) \leq L(\alpha_2)$ . De fato, o valor  $|I_1| + |I_2|$  é independente de  $\alpha$ . Além disso, o espaço  $|I_2| C - \sum_{i \in I_2} w_i$  que sobra para colocar os objetos de  $I_3$  considerando  $\alpha_1$  não pode ser menor do que o valor correspondente produzido por  $\alpha_2$ . Isto pois, o conjunto  $|I_2|$  produzido por  $\alpha_2$  é um subconjunto do conjunto  $|I_2|$  produzido por  $\alpha_1$ . Logo, uma vez que somente valores distintos de  $w_i \leq C/2$  produzem diferentes conjuntos  $I_3$  quando usados como  $\alpha$ , e como cada valor  $w_i$  domina  $w_i - 1, \dots, w_{i+1} + 1$  a consideração é válida. Além disso, fazendo-se  $\alpha = 0$  e substituindo-se em (2-2) tem-se

$$\begin{aligned} L(0) &= 0 + |I_2| + \max \left\{ 0, \left\lceil \frac{\sum_{i \in N} w_i - |I_2| C}{C} \right\rceil \right\} = \\ &= |I_2| + \max\{0, L_1 - |I_2|\}. \end{aligned}$$

Conseqüentemente,  $L_2 \geq L(0) = \max\{|I_2|, L_1\}$ . Este resultado abrange o

caso em que todos os objetos são maiores do que  $C/2$ , quando então todos os objetos pertencem ao conjunto  $I_2$  e o valor do limite é  $|I_2|$ . Também mostra que o limite  $L_2$  domina o limite  $L_1$ . É possível calcular  $L_2$  em tempo  $O(n)$ . A razão de desempenho de pior caso de  $L_2$  é  $R(L_2) = 2/3$  [64].

### 2.2.4.3

#### **Critério de Dominância e Redução**

Nesta seção descreve-se o critério de dominância apresentado por Martello e Toth [65, 64] e mostra-se como ele pode ser utilizado para reduzir o tamanho de um problema de BP e determinar um novo limite inferior.

Define-se um conjunto viável de objetos como qualquer conjunto  $F \subseteq N$  tal que  $\sum_{i \in F} w_i \leq C$ . Dados dois conjuntos viáveis distintos  $F_1$  e  $F_2$ , diz-se que  $F_1$  domina  $F_2$  se e somente se o número de caixas em uma solução ótima obtida fazendo a caixa  $B_1 = F_1$  não for maior do que o número de caixas em uma solução ótima obtida fazendo a caixa  $B_1 = F_2$ . Uma maneira possível de se verificar tal situação é aplicando o critério de dominância descrito a seguir.

Dados dois conjuntos viáveis distintos  $F_1$  e  $F_2$ , se existe uma partição  $P_1, \dots, P_\ell$  de  $F_2$  e um subconjunto  $\{i_1, \dots, i_\ell\} \subset F_1$  tal que  $w_{i_h} \geq \sum_{k \in P_h} w_k$  para  $h = 1, \dots, \ell$ , então  $F_1$  domina  $F_2$ . Isto porque, se existe uma solução viável tendo uma das caixas  $B_1 = F_2$ , então é possível obter uma solução viável com o mesmo número de caixas com  $B_1 = F_1$  da seguinte forma: troca-se cada partição  $P_h$  de  $F_2$  com o objeto correspondente  $i_h$  de  $F_1$  e move-se os objetos de  $F_1 \setminus \{i_1, \dots, i_\ell\}$  para  $B_1$ . A transformação no sentido oposto nem sempre é possível.

Martello e Toth usam o critério de dominância acima para apresentar o procedimento de redução MTRP, sumarizado a seguir. Um procedimento genérico para reduzir o tamanho de uma instância do problema de BP poderia enumerar todos os conjuntos viáveis, encontrar um (por exemplo,  $F$ ) que domine todos os demais, atribuir os objetos de  $F$  a uma nova caixa e removê-los de  $N$ . Este procedimento seria repetido enquanto existisse tal conjunto  $F$  ou até que não houvesse mais objetos em  $N$ . Uma vez que verificar todas as possibilidades é impraticável, o procedimento MTRP limita a busca a conjuntos viáveis de cardinalidade não maior do que três e evita enumerações inúteis.

A Figura 2.1 ilustra o procedimento MTRP conforme apresentado em [65]. O procedimento considera os objetos ordenados segundo (2-1) e para cada objeto  $i$  verifica a existência de um conjunto viável  $F$  tal que

$i \in F$ , com  $|F| \leq 3$ , dominando todos os conjuntos viáveis que contém  $i$ . Sempre que tal conjunto é encontrado, os objetos correspondentes são atribuídos a uma nova caixa e a busca continua com os objetos restantes. Quando não é possível determinar tal conjunto, armazena-se o objeto  $i$  no conjunto de objetos já considerados  $\bar{N}$ , inicializado na primeira linha. A variável que controla o número de caixas utilizadas  $m$  é inicializada na linha 2. Na linha 4,  $i$  é o índice do objeto mais pesado ainda não considerado. A instrução da linha 5 cria o conjunto  $N'$  de todos os objetos livres, desconsiderando-se o objeto  $i$ . Determina-se que o objeto mais pesado de  $N'$  será identificado pelo índice  $i_1$ , o segundo mais pesado pelo índice  $i_2$  e assim sucessivamente. O conjunto viável  $F$  é inicializado na linha 6. Em cada iteração do laço compreendido entre as linhas 3-30, a variável  $k$ , inicializada na linha 7, representa o maior número de objetos que podem pertencer a um conjunto viável que contém  $i$  (sem contar o próprio  $i$ ). O valor  $k = 0$  significa que não existe objeto algum que possa pertencer a qualquer conjunto viável contendo o objeto  $i$ , sendo possível definir imediatamente  $F = \{i\}$ , na linha 8. Na linha 10,  $i^*$  é o objeto mais pesado que pode pertencer a um conjunto viável contendo  $i$ . Na linha 11, se  $k = 1$ , então qualquer conjunto viável que contém  $i$  pode conter apenas um outro objeto, logo o conjunto viável  $F = \{i, i^*\}$  domina todos os demais conjuntos viáveis com  $i$ . Além disso, se  $w_i + w_{i^*} = C$  então  $F = \{i, i^*\}$  domina todos os conjuntos viáveis com  $i$ , qualquer que seja sua cardinalidade. Quando  $k = 2$  (linhas 13-21), qualquer conjunto viável com o objeto  $i$  pode conter até dois outros objetos. Na linha 14, determina-se o par de objetos  $(i_a, i_b)$  de maior peso que pode pertencer a um conjunto viável que contém o objeto  $i$ . Na linha 15, verifica-se se  $w_{i^*} \geq w_{i_a} + w_{i_b}$ , caso em que o conjunto  $F = \{i^*\}$  domina todos os pares de objetos (e pela definição de  $i^*$  todos os objetos isolados) que podem ser empacotados com  $i$ , logo  $\{i, i^*\}$  domina todos os conjuntos viáveis que contém  $i$ . O teste da linha 17 verifica se  $w_i = w_{i_a}$  e se  $b - a \leq 2$  ou  $w_i + w_{i_{b-1}} + w_{i_{b-2}} > C$ , caso em que o conjunto  $\{i_a, i_b\}$  domina todos os pares e todos os objetos isolados que podem ser empacotados junto com  $i$ . Caso não tenha sido possível determinar um conjunto com o objeto  $i$  que domine os demais conjuntos viáveis com este objeto, atualiza-se o conjunto  $\bar{N}$  na linha 24. Do contrário, cria-se nas linhas 26-28 a  $m$ -ésima caixa com o conjunto dominante  $F$  e atualiza-se o conjunto de objetos livres  $N$ . A busca pára quando não mais existir objetos livres ou todos os objetos tiverem sido considerados. A complexidade de MTRP é  $O(n^2)$ .

A seguir descreve-se um exemplo onde o critério de dominância pode ser aplicado. Seja a instância definida por  $n = 5$ ,  $w = (61, 60, 31, 30, 29)$  e

```

Procedimento MTRP
Entrada:  $N = \{1, \dots, n\}, C$ 
            $w_i (i = 1, \dots, n)$  onde  $w_h \geq w_{h+1}, h = 1, \dots, n - 1,$ 
Saida:  $m, B = (B_1, B_2, \dots, B_m).$ 
1  $\overline{N} \leftarrow \emptyset;$ 
2  $m \leftarrow 0;$ 
3 repita
4    $i \leftarrow \min\{h : h \in N \setminus \overline{N}\};$ 
5    $N' \leftarrow N \setminus \{i\}; N' = \{i_1, \dots, i_l\}$  com  $w_{i_1} \geq \dots \geq w_{i_l};$ 
6    $F \leftarrow \emptyset;$ 
7   Encontre o maior  $k$  tal que  $w_i + \sum_{q=l-k+1}^l w_{i_q} \leq C;$ 
8   se  $k = 0$  então  $F \leftarrow \{i\};$ 
9   senão faça
10     $i^* \leftarrow \min\{h \in N' : w_i + w_h \leq C\};$ 
11    se  $k = 1$  ou  $w_i + w_{i^*} = C$  então  $F \leftarrow \{i, i^*\};$ 
12    senão
13      se  $k = 2$  então faça
14        Encontre  $i_a, i_b \in N',$  com  $a < b,$  tal que  $w_{i_a} + w_{i_b} =$ 
            $\max\{w_{i_r} + w_{i_s}, i_r, i_s \in N' : w_{i_r} + w_{i_s} \leq C\}$ 
15        se  $w_{i^*} \geq w_{i_a} + w_{i_b}$  então  $F \leftarrow \{i, i^*\};$ 
16        senão
17          se  $w_{i^*} = w_{i_a}$  e  $(b - a \leq 2$  ou  $w_i + w_{i_{b-1}} + w_{i_{b-2}} > C)$  então
18             $F \leftarrow \{i, i_a, i_b\};$ 
19          fim-se
20        fim-se
21      fim-se
22    fim-se
23  fim-se
24  se  $F = \emptyset$  então  $\overline{N} \leftarrow \overline{N} \cup \{i\};$ 
25  senão faça
26     $m \leftarrow m + 1;$ 
27     $B_m \leftarrow F;$ 
28     $N \leftarrow N \setminus F;$ 
29  fim-se
30 até que  $N \setminus \overline{N} = \emptyset;$ 
31 retorne
fim MTRP

```

Figura 2.1: Pseudo-código do procedimento de redução MTRP

$C = 120$ . A primeira iteração do procedimento MTRP procura encontrar um conjunto viável  $F$  contendo o objeto  $i = 1$  que domine todos os demais conjuntos viáveis com este objeto. No máximo, os dois objetos mais leves cabem em uma caixa com o objeto  $i = 1$ . Logo, tem-se  $k = 2$  na linha 7. O objeto mais pesado que pode pertencer a um conjunto viável com  $i = 1$  é  $i^* = 3$ , determinado na linha 10, que possui peso igual a  $w_{i^*} = 31$ . O par de objetos  $(i_a, i_b)$  de maior peso que pode pertencer a um conjunto viável que contém o objeto  $i = 1$  é formado por  $i_a = 4$  e  $i_b = 5$  cujos pesos são  $w_{i_a} = 30$  e  $w_{i_b} = 29$ . Como  $w_{i^*} < w_{i_a} + w_{i_b}$  e  $w_{i^*} \neq w_{i_a}$ , o procedimento não é capaz de determinar que exista algum conjunto viável que contém o objeto  $i = 1$  que domine os demais. O procedimento considera então o

próximo objeto  $i = 2$ . No máximo, dois objetos podem ser empacotados em uma caixa junto com o objeto  $i = 2$ , tem-se, assim,  $k = 2$ , na linha 7. O objeto mais pesado  $i^* = 3$  que pode pertencer a um conjunto viável que contém  $i = 2$  possui peso igual a  $w_{i^*} = 31$ . O par de objetos  $(i_a, i_b)$  de maior peso que pode pertencer a um conjunto viável que contém o objeto  $i = 2$  é formado por  $i_a = 3$  e  $i_b = 5$  cujos pesos são  $w_{i_a} = 31$  e  $w_{i_b} = 29$ . Tem-se aqui  $w_{i^*} = w_a$  e a expressão  $b - a \leq 2$ , na linha 17) é verdadeira. Logo,  $F = \{i, i_a, i_b\} = \{2, 3, 5\}$  domina todos os outros conjuntos viáveis contendo  $i$ , que são:  $F_1 = \{2\}$ ,  $F_2 = \{2, 3\}$ ,  $F_3 = \{2, 4\}$ ,  $F_4 = \{2, 5\}$ ,  $F_5 = \{2, 4, 5\}$ . Os objetos 2, 3 e 5 podem ser colocados em uma caixa e removidos de  $N$ .

#### 2.2.4.4

##### Limite Inferior $L_3$ de Martello e Toth

O procedimento de redução descrito na seção anterior pode ser utilizado para determinar um novo limite inferior  $L_3$  para BP [64, 65]. Seja  $I$  a instância original de BP,  $b_1$  o número de caixas retornado pelo procedimento MTRP quando aplicado à instância  $I$ , e  $I_1$  a instância residual correspondente. O limite  $L_2$  quando aplicado à instância  $I_1$  resulta no limite  $L'_1 = b_1 + L_2(I_1)$  para a instância  $I$ . A seguir,  $I_1$  é relaxada através da remoção do objeto mais leve e MTRP é aplicado a esta instância relaxada, resultando em  $b_2$  caixas fixas e um novo problema residual  $I_2$ . Logo,  $L'_2 = b_1 + b_2 + L_2(I_2)$  é um limite inferior do problema original. Este procedimento é repetido até a instância residual ficar vazia. O limite inferior  $L'_k = b_1 + b_2 + \dots + b_k + L_2(I_k)$  é gerado a cada etapa  $k$ . Então,  $L_3 = \max\{L'_1, L'_2, \dots, L'_k\}$  é um limite inferior válido para a instância original. Este procedimento também pode ser usado para melhorar a redução obtida pelo procedimento MTRP quando aplicado à instância original. Em cada uma das iterações do procedimento que calcula  $L_3$ , se todos os objetos removidos puderem ser colocados nas caixas fixadas por MTRP, então esta redução também é válida para o problema original. Em particular, se todos os objetos tiverem sido colocados em alguma caixa, no fim do procedimento, então a solução heurística obtida é ótima. Uma vez que a complexidade de MTRP é  $O(n^2)$  a complexidade do cálculo de  $L_3$  é  $O(n^3)$ .



### 2.2.4.5

#### Limite Inferior $L_*^{(p)}$ de Fekete e Schepers

Fekete e Schepers [22] apresentam uma abordagem genérica para obter limites inferiores utilizando o conceito de *funções viáveis duais* apresentado a seguir.

Nesta seção assume-se que os pesos  $w_i \in [0, 1], i = 1, \dots, n$  e, como a capacidade da caixa  $C$  é apenas um fator de escala, esta é normalizada para 1.

**Definição 2.1** [22] *Uma função  $u : [0, 1] \rightarrow [0, 1]$  é chamada de viável dual se, para qualquer conjunto finito  $B$  de números reais não negativos, existe a relação*

$$\sum_{w \in B} w \leq 1 \Rightarrow \sum_{w \in B} u(w) \leq 1.$$

O termo dual refere-se ao fato de que para qualquer função viável dual  $u$  e para qualquer instância de *bin packing* com pesos  $(w_1, \dots, w_n)$  o vetor  $(u(w_1), \dots, u(w_n))$  é uma solução viável para o dual do problema contínuo de *bin packing* associado [54]. Este corresponde à relaxação linear da formulação por programação linear do problema de *bin packing*, que leva em conta o fato de que alguns pesos de objetos aparecem mais de uma vez. Considera-se um conjunto de  $n$  objetos de  $m$  diferentes pesos  $w_i = (i = 1, \dots, m)$ . Uma instância pode ser representada pelo conjunto múltiplo  $\{d_1 : w_1, d_2 : w_2, \dots, d_m : w_m\}$  onde  $d_i$  denota a quantidade de objetos de peso igual a  $w_i$  e  $\sum_{i=1}^m d_i = n$ . Uma *configuração* (ou padrão) é um conjunto múltiplo de quantidades e pesos que somem no máximo 1 (isto é, cabem em uma caixa). Sejam  $q$  o número de configurações distintas e  $a_{ij}$  o número de objetos de peso  $w_i$  na configuração  $j$ . Qualquer configuração  $j$  pode ser caracterizada pelo vetor  $(a_{1j}, a_{2j}, \dots, a_{mj})$  satisfazendo  $\sum_{i=1}^m w_i a_{ij} \leq 1$ . Uma solução viável para o problema contínuo de *bin packing* é um vetor  $(x_1, \dots, x_q)$  de números positivos em que  $x_j$  denota o número de caixas que devem usar a configuração  $j = 1, \dots, q$ . O objetivo é

$$\begin{aligned} & \text{Minimizar} && \sum_{j=1}^q x_j \\ & \text{sujeito a} && \sum_{j=1}^q a_{ij} x_j \geq d_i && i = 1, \dots, m \\ & && x_j \geq 0 && j = 1, \dots, q. \end{aligned}$$

O dual do problema contínuo de *bin packing* consiste então em

$$\begin{aligned} & \text{Maximizar} && \sum_{i=1}^m d_i u_i \\ & \text{sujeito a} && \sum_{i=1}^m a_{ij} u_i \leq 1 \quad j = 1, \dots, q \\ & && u_i \geq 0 \quad i = 1, \dots, m. \end{aligned}$$

As variáveis duais  $u_i$  representam o custo de cada objeto de peso  $w_i$ . O programa procura maximizar a soma dos custos de todos os pesos, sujeito à restrição de que os custos sejam positivos e a soma dos custos de todos os pesos em uma única caixa seja menor ou igual a um.

Baseados no lema abaixo, os autores mostram que funções viáveis duais podem ser usadas para melhorar limites inferiores do BP.

**Lema 2.1** [22] *Seja  $I = (w_1, \dots, w_n)$  uma instância de BP e seja  $u$  uma função viável dual. Qualquer limite para a instância de BP transformada  $u(I) = (u(w_1), \dots, u(w_n))$  é também um limite para  $I$ .*

Utilizando-se um conjunto de funções viáveis duais  $U$  e considerando-se o valor máximo sobre as instâncias transformadas  $u(I)$ ,  $u \in U$ , é possível obter limites melhores do que se usando uma única função para obter um limite.

**Teorema 2.2** [22] *Seja  $k \in \mathbb{N}$ . Então a função “escada”  $u^{(k)} : [0, 1] \rightarrow [0, 1]$*

$$w \rightarrow \begin{cases} w, & \text{se } w \cdot (k + 1) \in \mathbb{Z} \\ \lfloor (k + 1)w \rfloor \frac{1}{k}, & \text{caso contrário} \end{cases}$$

*é uma função viável dual.*

A Figura 2.2 ilustra o mecanismo de arredondamento, mostrando a diferença entre a função escada  $u^{(k)}$ ,  $k \in \{1, \dots, 4\}$  e a função identidade no intervalo  $[0, 1]$ . Seja a zona de “ganho” a área do gráfico (mais clara) que corresponde aos objetos que tiveram o custo do seu peso arredondado para cima e a zona de “perda” a área do gráfico (mais escura) correspondente aos objetos que tiveram o custo do seu peso arredondado para baixo. Uma boa função viável dual  $u^{(k)}$  para uma dada instância do BP é aquela em que a maioria dos objetos se encontram na zona de “ganho”. Quanto mais objetos na zona de “ganho”, mais forte será o limite inferior.

Nesta seção será utilizada a notação  $L(I)$  para representar a aplicação do limite  $L$  à instância  $I$ , definida por um conjunto múltiplo de pesos.

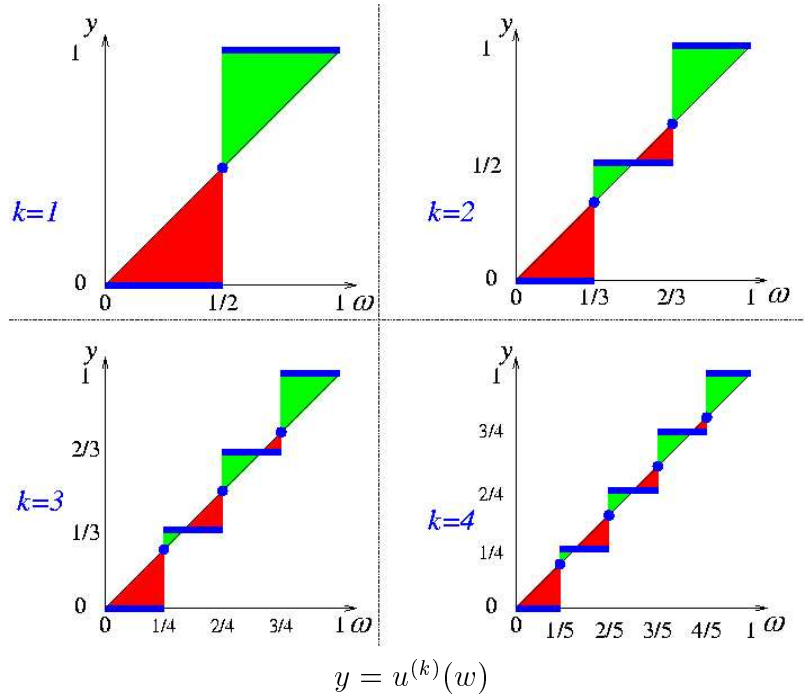


Figura 2.2: Zonas de ganho e de perda para a função  $u^{(k)}$

Fekete e Schepers apresentam uma formulação correspondente e simplificada do limite inferior  $L_2$  de Martello e Toth [65, 64], apresentado na Seção 2.2.4.2:

$$L_2(I) = \max_{\alpha \in [0, \frac{1}{2}]} \{ |\{w_i \in I : w_i > 1 - \alpha\}| + L_1(\{w_i \in I : \alpha \leq w_i \leq 1 - \alpha\}) \}$$

onde  $L_1(I)$  é o limite  $L_1$ , definido na Seção 2.2.4.1.

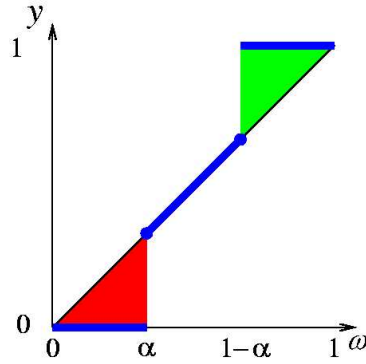
Reduz-se a instância negligenciando-se todos os objetos menores do que um dado valor  $\alpha$ . Reserva-se uma caixa para cada objeto de peso maior do que  $1 - \alpha$ , pois nenhum outro objeto da instância reduzida caberia em uma caixa junto com ele. Para os demais objetos, utiliza-se o limite  $L_1$ . A classe de funções viáveis duais descrita a seguir é a base para o limite inferior  $L_2$ .

**Teorema 2.3** [22] *Seja  $\alpha \in [0, \frac{1}{2}]$ . Então  $U^{(\alpha)} : [0, 1] \rightarrow [0, 1]$*

$$w \rightarrow \begin{cases} 1, & \text{se } w > 1 - \alpha \\ w, & \text{se } 1 - \alpha \geq w \geq \alpha \\ 0, & \text{se } w < \alpha \end{cases}$$

*é uma função viável dual.*

A Figura 2.3 mostra as áreas de perda e ganho correspondentes.



$$y = U^{(\alpha)}(w)$$

Figura 2.3: Zonas de ganho e de perda para a função  $U^{(\alpha)}$

Baseados no Lema 2.1 e no Teorema 2.3, Fekete e Schepers mostram como o limite inferior  $L_2$  pode ser formulado utilizando-se funções viáveis duais:

$$L_2(I) = \max_{\alpha \in [0, \frac{1}{2}]} L_1(U^{(\alpha)}(I)).$$

Em seguida, com o objetivo de melhorar este limite, utilizam funções viáveis duais adicionais para apresentar a nova classe de limites inferiores  $L_*^{(p)}$ . Por definição, composição de funções viáveis duais são viáveis duais. Para qualquer valor de  $k \in N$ , seja

$$L_2^{(k)}(I) = \max_{\alpha \in [0, \frac{1}{2}]} L_1(u^{(k)} \circ U^{(\alpha)}(I)). \tag{2-3}$$

Para qualquer  $p \geq 2$ , define-se o limite

$$L_*^{(p)}(I) = \max\{L_2(I), \max_{k=2, \dots, p} L_2^{(k)}(I)\}.$$

Para ilustrar o cálculo do limite  $L_*^{(p)}(I)$  considera-se a instância  $I$ , representada pelo conjunto múltiplo  $\{2:0.59, 2:0.57, 2:0.56, 4:0.47, 5:0.46, 1:0.45, 3:0.37, 4:0.36, 3:0.24, 2:0.2\}$ .

O primeiro limite calculado é  $L_1(I) = \lceil 11.74/C \rceil = 12$ . O limite  $L_2(I)$  não melhora este resultado. A Tabela 2.1 mostra os passos do cálculo dos limites  $L_2^{(8)}(I)$  e  $L_2^{(7)}(I)$ . As colunas descritas por  $i, d_i$  e  $w_i$  relacionam para cada  $i = 1, \dots, m$  a quantidade de objetos  $d_i$  de peso  $w_i$ . A coluna descrita por  $U^{(0)}(I)$  mostra o resultado da aplicação desta função viável dual ao vetor de pesos  $w_i$ , resultando no primeiro vetor solução  $y_i^{(1)}$ . Observa-se que  $w_i = y_i^{(1)}$  neste exemplo com  $\alpha = 0$ . O limite  $L_2^{(8)}(I)$  é calculado

$i$	$d_i$	$w_i$	$U^{(0)}(I)$	$L_2^{(8)}(I)$	zona	$L_2^{(7)}(I)$	zona
			$y_i^{(1)}$	$u^{(8)} \circ U^{(0)}(I)$		$y_i^{(2)}$	
1	2	0.59	0.59	0.625 = 5/8	ganho	0.571 = 4/7	perda
2	2	0.57	0.57	0.625 = 5/8	ganho	0.571 = 4/7	ganho
3	2	0.56	0.56	0.625 = 5/8	ganho	0.571 = 4/7	ganho
4	4	0.47	0.47	0.500 = 4/8	ganho	0.428 = 3/7	perda
5	5	0.46	0.46	0.500 = 4/8	ganho	0.428 = 3/7	perda
6	1	0.45	0.45	0.500 = 4/8	ganho	0.428 = 3/7	perda
7	3	0.37	0.37	0.375 = 3/8	ganho	0.285 = 2/7	perda
8	4	0.36	0.36	0.375 = 3/8	ganho	0.285 = 2/7	perda
9	3	0.24	0.24	0.250 = 2/8	ganho	0.142 = 1/7	perda
10	2	0.20	0.20	0.125 = 1/8	perda	0.142 = 1/7	perda

Tabela 2.1: Exemplo do limite  $L_*^{(p)}(I)$

utilizando-se o resultado da aplicação da função escada  $u^{(8)}$  a  $y_i^{(1)}$  (solução obtida por  $U^{(0)}(I)$ ), descrito pelo vetor solução  $y_i^{(2)}$  na próxima coluna. A coluna descrita por “zona” identifica se o custo do peso do objeto foi arredondado para cima (ganho) ou para baixo (perda). De acordo com o Lema 2.1, qualquer limite para a instância transformada por funções viáveis duais (como por exemplo  $u^{(8)} \circ U^{(0)}(I)$ ) é um limite válido para a instância original  $I$ . Então, aplicando-se o limite  $L_1$  a esta instância transformada tem-se o limite  $L_2^{(8)}(I) = \lceil \frac{\sum_{i=1}^{10} y_i^{(2)} d_i}{C} \rceil = 13$ , que é mais forte do que o limite  $L_2$ . A próxima coluna mostra os dados para cálculo do limite  $L_2^{(7)}(I)$  para a mesma instância  $I$ . A aplicação da função escada  $u^{(7)}$  a  $y_i^{(1)}$  resulta no terceiro vetor solução  $y_i^{(3)}$ . Neste caso, o limite  $L_2^{(7)}(I) = \lceil \frac{\sum_{i=1}^{10} y_i^{(3)} d_i}{C} \rceil = 11$  é pior do que o limite  $L_2$  e tem-se muitos objetos na zona de perda.

Os autores provam que a razão assintótica de pior caso de  $L_2$  de  $2/3$  é melhorada para  $R(L_*^{(p)}) = 3/4$ . Como ressaltaram, a principal contribuição desse trabalho não é a razão assintótica de pior caso, mas sim que os limites podem ser calculados rapidamente em tempo  $O(n)$  (considerando-se um valor fixo para  $p$ ).

### 2.2.4.6

#### Novo Limite Inferior $L_\vartheta$

Os limites inferiores apresentados até agora são baseados em relaxações. Outra abordagem possível é, dado um determinado limite  $L$ , tentar provar a inexistência de uma solução com este valor, e fazer  $L + 1$  um novo limite inferior válido. Esta técnica é chamada por Scholl et al. [77] de um limite destrutivo. Recentemente, Schoenfeld [76] propôs um novo limite  $L_C$

baseado em uma técnica semelhante, que ele chamou de *bound-derived lower bounding test*. Com base nos resultados de Dell'Amico e Martello [16] e nas idéias de Schoenfeld [76] apresenta-se a seguir o limite  $L_\vartheta$  para BP que é, entretanto, dominado por  $L_C$ .

Utilizando-se um limite inferior  $m$  no número de caixas de uma solução viável para BP, determina-se limites inferior e superior para o número de objetos por caixa, para em seguida tentar provar que não existe solução viável com  $m$  caixas.

Considera-se uma instância de BP definida pelo conjunto de objetos  $N = \{1, \dots, n\}$ , pesos  $w_i (i = 1, \dots, n)$  e capacidade da caixa  $C$ . Supõe-se que existe solução viável com  $m$  caixas. Então,

$$\Theta = \max \left\{ q : \sum_{i=n-q+1}^n w_i \leq C \right\} \quad (2-4)$$

é um limite superior para o número de objetos por caixa.

**Proposição 2.4** *Dada uma instância de BP, sejam  $m$  um limite inferior para o número de caixas e  $\Theta$  calculado segundo a expressão (2-4). Se  $\Theta < \lceil n/m \rceil$ , então  $m + 1$  é um limite inferior para o número de caixas.*

*Prova.* Nenhuma solução viável pode ter mais do que  $\Theta$  objetos por caixa. Dado o número médio de objetos por caixa  $n/m$ , pelo menos uma caixa possui  $\lceil n/m \rceil$  ou mais objetos. Logo, se  $\Theta$  é menor do que  $\lceil n/m \rceil$ , então não existe solução com  $m$  caixas para a instância.  $\square$

Seja  $\lfloor n/m \rfloor$  o maior inteiro menor ou igual ao número médio de objetos por caixa. A seguir apresenta-se duas condições suficientes para determinar valores de  $\sigma$  para os quais  $\vartheta \geq \sigma$ .

**Proposição 2.5** *Dada uma instância de BP e um inteiro  $\sigma \leq \lfloor n/m \rfloor$ , se*

$$C(\sigma) = \left\lceil \sum_{i=\sigma}^n \frac{w_i}{m-1} \right\rceil > C,$$

*então qualquer solução com  $m$  caixas deve possuir pelo menos  $\sigma$  objetos em cada caixa.*

*Prova.* Para cada valor possível de  $\sigma \leq \lfloor n/m \rfloor$  considera-se uma solução onde a caixa, por exemplo,  $B_1$ , possui  $\sigma - 1$  ou menos objetos. Sobram para as demais caixas  $B_2, \dots, B_m$  pelo menos  $n - (\sigma - 1)$  objetos. Seja  $\left\lceil \sum_{i=\sigma}^n \frac{w_i}{m-1} \right\rceil > C$  um limite inferior para a capacidade das caixas capaz de permitir uma solução viável com  $m - 1$  caixas para a instância reduzida

definida pelos  $n - (\sigma - 1)$  objetos mais leves de pesos  $w_\sigma, \dots, w_n$ . Se  $C(\sigma) > C$ , então uma solução com  $m$  caixas não pode ter menos do que  $\sigma$  objetos por caixa.  $\square$

**Proposição 2.6** *Dados uma instância de BP e um inteiro  $\sigma \leq \lfloor n/m \rfloor$ , se  $\sum_{i=1}^{\sigma} w_i \leq C$  então existe uma solução ótima com pelo menos  $\sigma$  objetos por caixa.*

*Prova.* cf. [16, pág. 194, Proposição 4]  $\square$

As Proposições 2.5 e 2.6 levam ao seguinte teorema:

**Teorema 2.7** *Para qualquer instância de BP, dado um limite inferior  $m$  para o número de caixas, então*

$$\vartheta = \max\left\{ \max_{\sigma=1, \dots, \lfloor n/m \rfloor} \{\sigma : C(\sigma) > C\}, \max_{\sigma=1, \dots, \lfloor n/m \rfloor} \left\{ \sigma : \sum_{i=1}^{\sigma} w_i \leq C \right\} \right\}$$

*é um limite inferior para o número de objetos por caixa de uma solução viável.*

As próximas duas proposições mostram como os limites inferior  $\vartheta$  e superior  $\Theta$  para o número de objetos por caixa, de uma solução com  $m$  caixas, podem ser usados para melhorar o limite inferior para o número de caixas de uma solução viável.

**Proposição 2.8** *Dados uma instância de BP e um limite inferior  $m$  para o número de caixas, seja  $\underline{m}_\vartheta = \max\{m - (n - \vartheta m), 0\}$  um limite inferior para o número de caixas com exatamente  $\vartheta$  objetos, onde  $\vartheta$  é definido segundo o Teorema 2.7. Se  $\left\lceil \sum_{i=\underline{m}_\vartheta \vartheta + 1}^n \frac{w_i}{(m - \underline{m}_\vartheta)} \right\rceil > C$ , então  $m + 1$  é um limite inferior para o número de caixas de uma solução viável.*

*Prova.* Uma solução viável com  $m$  caixas possui pelo menos  $\underline{m}_\vartheta$  caixas com exatamente  $\vartheta$  objetos em cada. Considera-se qualquer solução em que os  $\underline{m}_\vartheta \vartheta$  objetos mais pesados estão, por exemplo, nas caixas  $B_1, \dots, B_{\underline{m}_\vartheta}$ . Se  $\left\lceil \sum_{i=\underline{m}_\vartheta \vartheta + 1}^n \frac{w_i}{(m - \underline{m}_\vartheta)} \right\rceil > C$ , os  $n - (\underline{m}_\vartheta \vartheta)$  objetos restantes não podem ser acomodados nas  $m - \underline{m}_\vartheta$  caixas restantes e o limite inferior para o número de caixas pode ser incrementado em uma unidade.  $\square$

Se  $\Theta = \vartheta + 1$ , cada caixa possui exatamente  $\vartheta$  ou  $\Theta$  objetos. Neste caso, os números  $m_\vartheta$  de caixas com exatamente  $\vartheta$  objetos e  $m_\Theta$  de caixas

com exatamente  $\Theta$  objetos podem ser calculados resolvendo-se o sistema  $(m = m_{\vartheta} + m_{\Theta}, n = \Theta m_{\Theta} + \vartheta m_{\vartheta})$ , o que leva a:

$$m_{\vartheta} = (\vartheta + 1)m - n \quad (2-5)$$

$$m_{\Theta} = n - \vartheta m. \quad (2-6)$$

Logo, um limite inferior para a capacidade das caixas necessária para acomodar os  $n$  objetos em  $m$  caixas é dado por:

$$C_{\vartheta}(m) = \max \left\{ \left\lceil \sum_{i=n-\vartheta m_{\vartheta}+1}^n w_i/m_{\vartheta} \right\rceil, \left\lceil \sum_{i=n-\Theta m_{\Theta}+1}^n w_i/m_{\Theta} \right\rceil \right\}. \quad (2-7)$$

**Proposição 2.9** *Dada uma instância de BP e um limite inferior  $m$  para o número de caixas, se  $\Theta = \vartheta + 1$  e  $C_{\vartheta}(m) > C$  então  $m + 1$  é um limite inferior válido para a instância.*

*Prova.* Considera-se qualquer solução com  $m_{\Theta}$  caixas com exatamente  $\Theta$  objetos em cada e  $m_{\vartheta}$  caixas com exatamente  $\vartheta$  objetos em cada. Se a capacidade da caixa necessária para acomodar os  $m_{\vartheta} \vartheta$  objetos mais leves em  $m_{\vartheta}$  caixas for maior do que  $C$  ou se a capacidade da caixa necessária para acomodar os  $m_{\Theta} \Theta$  objetos mais leves em  $m_{\Theta}$  caixas for maior do que  $C$ , então não existe solução viável com  $m$  caixas e o limite inferior para o número de caixas pode ser incrementado em uma unidade.  $\square$

O próximo teorema é decorrência das Proposições 2.4, 2.8 e 2.9.

**Teorema 2.10** *Seja  $m$  um limite inferior para o número de caixas. Então,*

$$L_{\vartheta}(m) = \begin{cases} m + 1, & \text{se qualquer uma das condições (a), (b), ou (c)} \\ & \text{abaixo for verdadeira:} \\ & (a) \theta < \lceil n/m \rceil \\ & (b) \left\lceil \sum_{i=\underline{m}_{\vartheta} \vartheta + 1}^n \frac{w_i}{(m - \underline{m}_{\vartheta})} \right\rceil > C \\ & (c) \theta = \vartheta + 1 \text{ and } C_{\vartheta}(m) > C; \\ m, & \text{caso contrário} \end{cases}$$

*é um limite inferior válido para o número de caixas de uma solução viável.*

Se  $\Theta > \vartheta + 1$ , pode-se fazer a seguinte redução para tentar melhorar o limite inferior corrente. Se  $\vartheta < \lceil n/m \rceil$  tenta-se provar que não existe solução viável com o número mínimo de objetos por caixa igual a  $\vartheta$ , para então incrementar de um o valor de  $\vartheta$  e em seguida calcular  $L_{\vartheta}$  para este novo valor. Para verificar se existe solução viável tendo pelo menos uma caixa com apenas  $\vartheta$  objetos, coloca-se em uma caixa os  $\vartheta$  objetos mais



pesados. Em seguida, considera-se a instância reduzida com  $n' = n - \vartheta$  objetos de pesos  $w_{\vartheta+1}, \dots, w_n$ . Neste caso,  $m' = m - 1$  é um limite inferior para o número de caixas. O limite  $L_{\vartheta}(m')$  para esta nova instância pode ser calculado conforme o Teorema 2.10. Se  $L_{\vartheta}(m') > m'$ , então provou-se que não existe solução viável para a instância original com o número mínimo de objetos por caixa  $\vartheta$ . Logo, incrementa-se  $\vartheta$  de um e calcula-se o limite  $L_{\vartheta}(m)$  para a instância original.

Para ilustrar o cálculo do limite  $L_{\vartheta}$ , considera-se a instância de BP definida por  $n = 20$ ,  $w = (54, 54, 53, 53, 53, 52, 51, 51, 51, 50, 50, 48, 48, 46, 46, 33, 32, 32, 32, 31)$  e  $C = 120$ .

Calcula-se o limite inferior  $L_1 = \lceil 920/120 \rceil = 8$ . Calcula-se  $\Theta$  e aplica-se o Teorema 2.7 com  $m = 8$ :

$$\begin{aligned} \Theta &= 3 \quad \text{número máximo de objetos por caixa} \\ \vartheta &= 2 \quad \text{número mínimo de objetos por caixa} \end{aligned}$$

Como  $\Theta = \vartheta + 1$ , resolve-se as equações (2-5, 2-6, 2-7), obtendo-se  $m_{\vartheta} = 4$  e  $m_{\Theta} = 4$ . Logo,

$$\begin{aligned} C_{\vartheta}(8) &= \max \left\{ \left\lceil \sum_{i=13}^{20} w_i/4 \right\rceil, \left\lceil \sum_{i=9}^{20} w_i/4 \right\rceil \right\} = \\ &= \max \left\{ \left\lceil 300/4 \right\rceil, \left\lceil 499/4 \right\rceil \right\} = 125 > C. \end{aligned}$$

Então, conforme a Proposição 2.9,  $m + 1 = 9$  é um limite inferior válido.

Outros exemplos da aplicação do limite  $L_{\vartheta}$  a instâncias de BP são apresentados no Apêndice A.

No próximo capítulo propõe-se uma heurística para BP. Entre os diversos componentes desta heurística, estão os limites inferiores e o critério de dominância apresentados nesta seção, assim como algumas heurísticas apresentadas na Seção 2.2.2.