Algoritmos de aproximação para o problema de empacotamento em faixa

Gabriel Perri Gimenes
Marcos Okamura Rodrigues
Milene Alves Garcia

ICMC-USP

26 de novembro de 2015

Sumário

Introdução

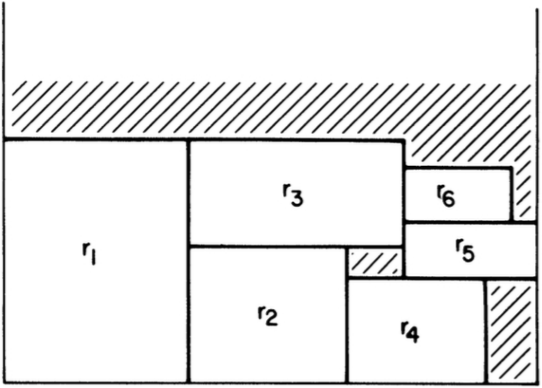
- Problemas de empacotamento são aqueles que requerem que certos itens sejam empacotados em outros de tamanhos maiores, chamados de recipientes;
- Estes problemas devem ser feitos não considerando sobreposições de itens;
- Um dos problemas conhecidos pela indústria é o corte de um rolo de um determinado material para obtenção de itens menores.

O problema de empacotamento em faixa

Definition

Seja S um recipiente retangular de largura W e altura infinita e uma lista de itens retangulares $L = (r_1, ..., r_n)$, onde cada item $r_i = (w_i, h_i)$ é tal que $w_i \in (0, W]$, para $i = 1, ..., n$, w_i é a largura e h_i é a altura do item r_i. O objetivo é empacotar os itens de L em S (sem sobreposições) com a menor altura possível.

Abaixo temos o exemplo de uma instância do empacotamento em faixa:



Introdução

NP-Completude

O problema de empacotamento em faixa é NP-difícil.

- Bin packing (1D) $\rightarrow$ Strip packing
  Empacotamento em faixa de retângulos de mesma altura;
- Makespan minimization $\rightarrow$ Strip packing
  Empacotamento em faixa de retângulos de mesma largura.

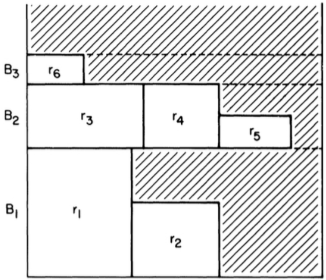
Introdução

Revisão da literatura de algoritmos de aproximação

Publicação	Algoritmo	Fator de aproximação
Harren et al. (2014)		$(5/3 + \epsilon)$
Harren and van Stee (2009)		1.9396
Steinberg (1997)		2
Schiermeyer (1994)	Reverse Fit	2
Sleator (1980)		2.5
Coffman et al. (1980)	First Fit	2.7
Golan (1981)	Split	3
Coffman et al. (1980)	Split Fit	3
Coffman et al. (1980)	Next Fit	3
Baker et al. (1980)	Bottom Left	3

Algoritmo Next Fit (NFDH)

1. Ordene as peças em uma lista *L* de altura não-crescente;
2. Empacote as peças ordenadas na parte inferior esquerda do recipiente até que não haja espaço horizontal suficiente para uma nova peça;
3. Defina uma linha imaginária horizontal sobreposta à parte superior da maior peça;
4. Empacote as peças ordenadas restantes na parte inferior esquerda deste novo nível até que não haja espaço horizontal suficiente para uma nova peça;
5. Repita os passos 3 e 4 até que não haja mais peças.



Algoritmo Next Fit (NFDH)

Teorema

Para qualquer lista L ordenada com altura não-crescente,

$$NFDH(L) \leq 3\ OPT(L)$$

onde a constante 3 é a menor possível.

Algoritmo Next Fit

Demonstração (1)

Considere o empacotamento NFDH em L com blocos $B_1, ..., B_t$, e para cada i seja x_i a largura do primeiro retângulo em B_i e y_i a largura total dos retângulos em B_i. Para cada $i < t$, o primeiro retângulo em B_{i+1} não pode ser inserido em B_i. Porém, como $y_i + x_{i+1} > 1, 1 \leq i < t$, e como cada retângulo em B_i tem altura no mínimo H_{i+1}, e o primeiro retângulo em B_{i+1} tem altura H_{i+1}, $A_i + A_{i+1} \geq H_{i+1}(y_i + x_{i+1}) > H_{i+1}$.

Demonstração (2)

Mais que isso, se A é a área total de todos os retângulos,

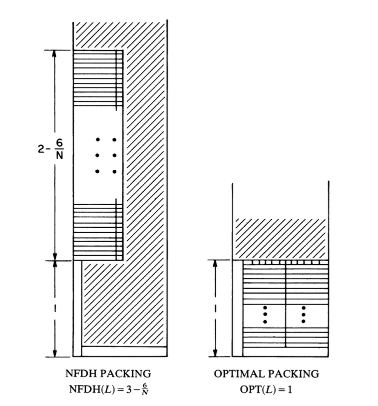
$$\begin{aligned} NFDH(L) = & \sum_{i=1}^{\text{t}} H_i \leq H_1 + \sum_{i=1}^{\text{t-1}} A_i + \sum_{i=2}^{\text{t}} A_i \leq H_1 + 2A && (1) \\ \leq & \quad OPT(L) + 2\ OPT(L) = 3\ OPT(L), && (2) \end{aligned}$$

pois a altura da maior peça é menor ou igual ao valor ótimo.

Algoritmo Next Fit

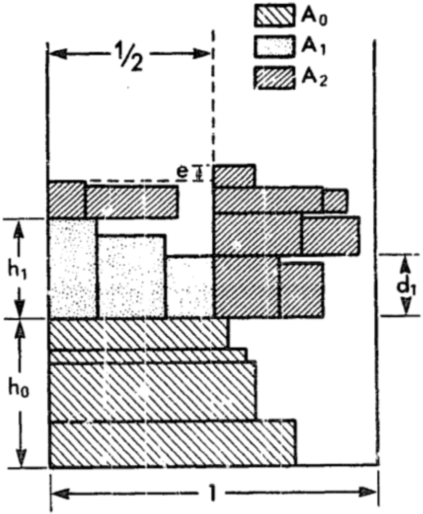
Demonstração (3)

A figura ao lado ilustra como o fator de aproximação algoritmo NFDH é arbitrariamente próximo a 3.
De fato, estamos considerando uma lista L com $NFDH(L) = (3 - 6\epsilon)\ OPT(L)$ para qualquer $\epsilon = 1/N$. Esta lista é dada por um retângulo $2\epsilon \times 1$, um retângulo $(1 - 2\epsilon) \times 2\epsilon$, e $2N - 6$ pares de retângulos com dimensões $(1/2 - \epsilon) \times \epsilon$ e $3\epsilon \times \epsilon$.



Algoritmo de Sleator

1. Empilhe as peças com largura $w_i > \frac{1}{2}$;
2. Ordena as peças restantes em ordem não-crescente de altura;
3. Aplique o algoritmo Next Fit para criar a primeira linha de peças;
4. Desenhe uma reta vertical que divida a placa ao meio e aplique o algoritmo Next Fit, criando apenas uma linha de peças no lado com menor altura;
5. Repita o passo 4 até que não hajam mais peças disponíveis.



Algoritmo de Sleator

Lema

No algoritmo de Sleator, temos que a seguinte desigualdade é satisfeita:

$$S \leq 4\ A_2 + d_1$$

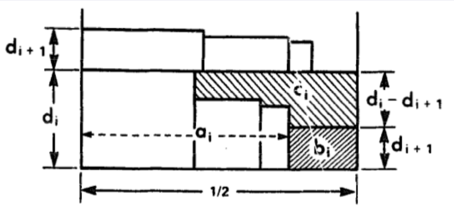
onde S é a soma das alturas de todas as linhas em A_2, A_2 é a área das peças na região definida no passo 3 e d_1 é a altura acima de h_0 do maior ponto de qualquer peça na metade à direita (ou parcialmente na metade à direita).

Algoritmo de Sleator

Demonstração (1)

As peças de A_2 serão empacotadas em um recipiente de altura S e largura $\frac{1}{2}$ em ordem não-crescente de altura, linha por linha.

Seja p_1 a primeira linha empacotada, p_2 a próxima, até a última linha empacotada p_n. Seja d_i a altura de p_i. Seja R_i o retângulo de dimensões $d_i \times \frac{1}{2}$ no qual as peças de p_i são empacotadas. Nós particionamos R_i em três partes disjuntas a_i, b_i e c_i, conforme é ilustrado na figura.



Algoritmo de Sleator

Demonstração (2)

Sejam $A_2 = \bigcup_i a_i$, $B = \bigcup_i b_i$ e $C = \bigcup_i c_i$.

Como A_2, B e C são disjuntos e cobrem o retângulo $\frac{1}{2} \times S$, temos que:

$$\frac{1}{2}S = A_2 + B + C.$$

Além disso, sabemos que $b_i \leq a_{i+1}$, porque a primeira peça em a_{i+1} é mais larga e possui pelo menos a mesma altura de b_i. Logo, $B \leq A_2$.

Algoritmo de Sleator

Demonstração (3)

Por outro lado, considere a partição C. A altura de c_i é $d_i - d_{i+1}$ para todo $i \leq n-1$ e a altura de c_n é d_n. Então, segue que:

$$\sum_i \text{altura de } c_i = d_n + \sum_{1 \leq i \leq n-1} (d_i - d_{i+1}) = d_1.$$

Algoritmo de Sleator

Demonstração (4)

Assim, as peças de C podem ser posicionadas em um retângulo $\frac{1}{2} \times d_1$ sem sobreposição, logo, $C \leq \frac{1}{2}d_1$.
Combinando as três relações acima, obtemos:

$$\frac{1}{2}S \leq A_2 + A_2 + \frac{1}{2}d_1$$

que implica no resultado desejado:

$$S \leq 4A_2 + d_1.$$

Algoritmo de Sleator

Teorema

Sejam H_{ALG} a altura do empacotamento dado pelo algoritmo de Sleator e H_{OPT} a altura do empacotamento ótimo. Então, temos que:

$$H_{ALG} \leq 2.5\ H_{OPT}$$

onde a constante 2.5 é a menor possível.

Algoritmo de Sleator

Demonstração (1)

Seja S a soma das alturas de todas as linhas em A_2. Seja e a diferença de altura entre as metades à direita e à esquerda no final do empacotamento. Observe que $2h_0 + h_1 + S + e$ é exatamente o dobro de H_{ALG}.

Como mais da metade da área do recipiente abaixo de h_0 está preenchida com peças, temos que:

$$\frac{1}{2}h_0 \leq A_0.$$

Usando o Lema 1, temos que:

$$S \leq 4A_2 + d_1.$$

Combinando as desigualdades acima, segue que:

$$2H_{ALG} = 2h_0 + h_1 + S + e \leq 4A_0 + h_1 + 4A_2 + d_1 + e.$$

Demonstração (2)

Como as peças são empacotadas em ordem não-crescente de altura, todas as peças em A_1 tem altura pelo menos d_1. Então, ou d_1 não é nulo, caso as larguras de todas as peças em A_1 somem exatamente $\frac{1}{2}$; ou d_1 é nulo, caso não hajam mais peças no passo 3 antes de qualquer peça cruzar a linha central. Em qualquer um dos casos, temos que:

$$\frac{1}{2}d_1 \leq A_1.$$

As duas desigualdades acima implicam que:

$$2H_{ALG} \leq 4(A_0 + A_1 + A_2) + h_1 + e - d_1.$$

Algoritmo de Sleator

Demonstração (3)

Como a solução ótima tem altura H_{OPT}, nós sabemos que a área de todas as peças não pode exceder H_{OPT}, i.e., $A_0 + A_1 + A_2 \leq H_{OPT}$.
Substituindo esta desigualdade na expressão acima, obtemos:

$$2H_{ALG} \leq 4H_{OPT} + h_1 + e - d_1.$$

Algoritmo de Sleator

Demonstração (4)

Se a altura da coluna à direita nunca exceder a da esquerda, então a altura do empacotamento inteiro é $h_0 + h_1$. Além disso, $h_0 \leq H_{OPT}$ visto que não é possível empacotar duas peças em A_0 no mesmo nível em qualquer solução. Nós também sabemos que $h_1 \leq H_{OPT}$, então nesse caso a altura do empacotamento é limitada por $2H_{OPT}$.

Demonstração (5)

Se a altura da coluna à direita exceder a da esquerda, então e é limitado pela maior altura de uma linha empacotada em A_2, i.e., $e \leq d_1$. Combinando isto com a última linha acima e o fato de que $h_1 \leq H_{OPT}$, nós obtemos o resultado desejado:

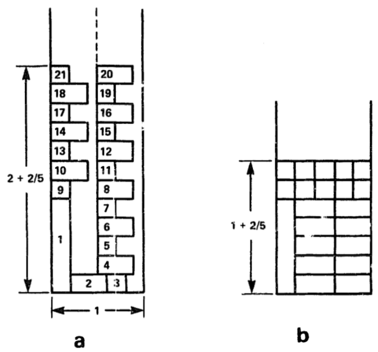
$$H_{ALG} \leq 2H_{OPT} + \frac{1}{2}H_{OPT} = 2.5\ H_{OPT}.$$

Pior Caso

Seja S_k o conjunto das seguintes peças:

- uma peça com altura 1 e largura $\frac{1}{k}$;
- $2k$ peças com altura $\frac{1}{k}$ e largura $\frac{1}{2} - \frac{1}{2k}$;
- $2k$ peças com altura $\frac{1}{k}$ e largura $\frac{1}{k}$.

O empacotamento de Sleator tem altura $1 + (\frac{1}{k})\lceil \frac{1}{2}(3k-1) \rceil$.
O empacotamento ótimo tem altura $1 + \frac{2}{k}$.



Experimentos Computacionais

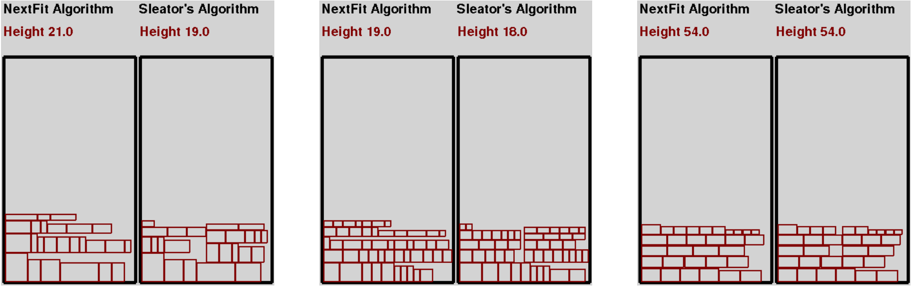
- Comparar empiricamente os dois algoritmos: NextFit e Sleator
- Código desenvolvido em Python
- Recebe o conjunto de retângulos e a largura do recipiente
- Retorna o empacotamento e a altura
- Permite a visualização do empacotamento

Instâncias analisadas

Tabela: Instâncias analisadas.

Instância	# Retângulos	Fonte
J1	25	Jakobs,1996
J2	50	Jakobs,1996
D1	31	Ratanapan,1997
D2	21	Ratanapan,1997
D3	37	Ratanapan,1998
D4	37	Dagli,1997
Kendall	13	Burke,1999
N1a - N1e	17	Hopper,2000

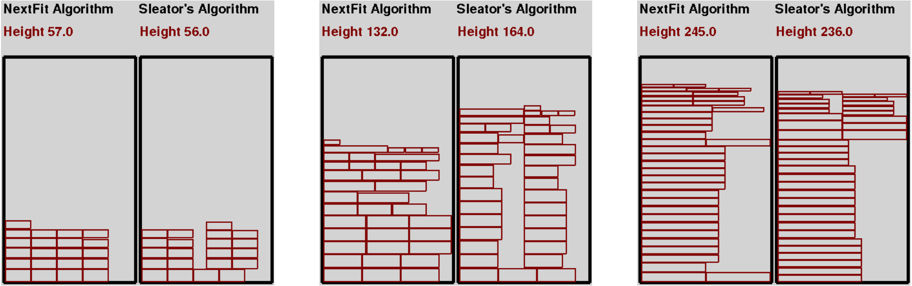
Resultados



(a) J1 (b) J1 (c) J2 (d) J2 (e) D1 (f) D1

Figura: Empacotamento resultante para cada uma das instâncias.

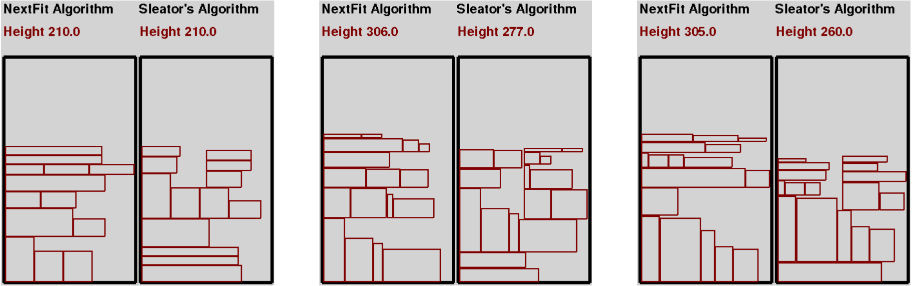
Resultados



(a) D2 (b) D2 (c) D3 (d) D3 (e) D4 (f) D4

Figura: Empacotamento resultante para cada uma das instâncias.

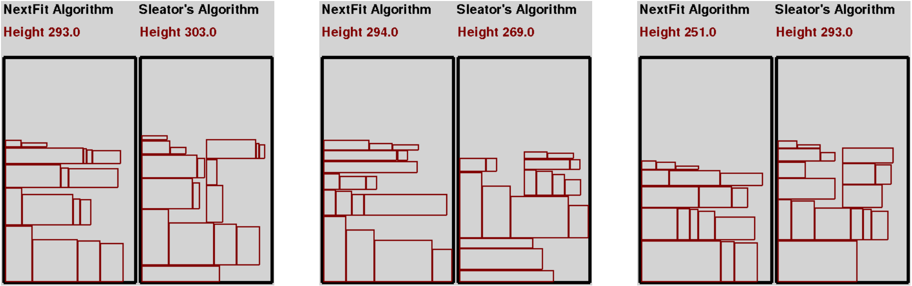
Resultados



(a) Kendall (b) Kendall (c) N1a (d) N1a (e) N1b (f) N1b

Figura: Empacotamento resultante para cada uma das instâncias.

Resultados



(a) N1c (b) N1c (c) N1d (d) N1d (e) N1e (f) N1e

Figura: Empacotamento resultante para cada uma das instâncias.

Comparação

- 12 instâncias
- 3 empates
- 6 vitórias do Sleator
- 3 vitórias do NextFit

Tabela: Comparação dos algoritmos NextFit e Sleator.

Instância	NextFit	Sleator	Ótimo
J1	21	19	15
J2	19	18	15
D1	54	54	?
D2	57	56	?
D3	132	164	?
D4	245	236	?
Kendall	210	210	140
N1a	306	277	200
N1b	305	260	200
N1c	293	303	200
N1d	294	269	200
N1e	251	293	200

Conclusão

- Problema de empacotamento em faixa - NP-Difícil
- Dois algoritmos de aproximação - NextFit e Sleator
- NextFit é uma 3-aproximação
- Sleator é uma 2.5-aproximação
- Resultados mostraram que o Sleator funcionou melhor para maioria das instâncias
- Ainda assim NextFit ganhou algumas: Fator de aproximação vs. Instâncias