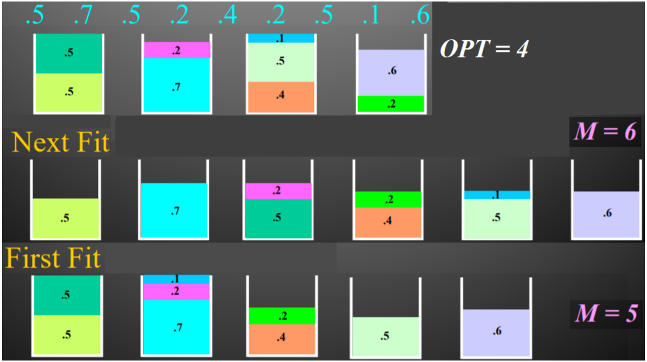
Bin Packing

- **Instância:** n itens com pesos $p_1, p_2, \ldots, p_n$ racionais em $(0, 1]$.
- **Objetivo:** Obter o menor número M de caixas (*bins*) de capacidade 1 para empacotar (*packing*) todos os itens.
- Limite inferior básico: $opt \geq \sum_{k=1}^{n} p_k = p_1 + p_2 + \ldots + p_n$
- Algoritmo Next-Fit: Empacotar itens na ordem que chegam. Se não cabe na caixa atual, fecha a caixa e abre uma nova caixa (atual).
- Algoritmo First-Fit: Colocar o item na primeira caixa na qual o item cabe. Se não cabe em nenhuma, abre uma nova caixa.

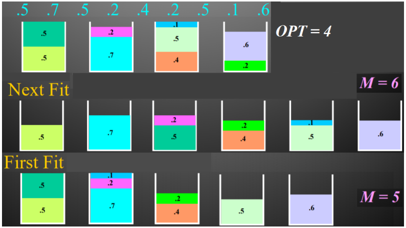


Next-Fit e First-Fit são 2-aproximativos

Algoritmo Next-Fit (NF): Empacotar itens na ordem que chegam. Se não cabe na caixa atual, fecha a caixa e abre uma nova caixa (atual).
Algoritmo First-Fit (FF): Colocar o item na primeira caixa na qual o item cabe. Se não cabe em nenhuma, abre uma nova caixa.

PROVA:

- Sejam $B_1, \ldots, B_M$ as caixas (bins) do Next Fit
- Para duas caixas consecutivas B_i e B_{i+1}: $p(B_i) + p(B_{i+1}) > 1$
- Senão B_{i+1} caberia em B_i.
- Somando de 2 em 2: $\sum_{i=1}^{M} p(B_i) > \lfloor M/2 \rfloor \geq (M-1)/2$
- Logo: $opt \geq \sum_{k=1}^{n} p_k = \sum_{i=1}^{M} p(B_i) > (M-1)/2$
- Logo: $M - 1 < 2 \cdot opt$. Portanto: $M \leq 2 \cdot opt$.



Next-Fit é 2-aproximativo (e não menos !!)

Algoritmo Next-Fit (NF): Empacotar itens na ordem que chegam. Se não cabe na caixa atual, fecha a caixa e abre uma nova caixa (atual).

Tight Example (Next-Fit 2-aproximativo)

- $4n$ itens: 2n grupos em sequência cada um com 2 itens: $\frac{1}{2}$ e $\frac{1}{2n}$
- $opt = n + 1$: n caixas $\{\frac{1}{2}, \frac{1}{2}\}$ e 1 caixa com $2n$ itens $\frac{1}{2n}$
- **Next-Fit:** $M = 2n$ grupos $\{\frac{1}{2}, \frac{1}{2n}\}$
- $M = 2n = 2 \cdot opt - 2.$
- Fator de aprox.: $M/opt = 2 - 2/opt.$

First-Fit e Best-Fit são 1.7-aproximativos

Algoritmo First-Fit (FF): Colocar o item na primeira caixa na qual o item cabe. Se não cabe em nenhuma, abre uma nova caixa.
Algoritmo Best-Fit (BF): Se o item couber em alguma caixa aberta, colocar na que sobra menos espaço. Caso contrário, abre uma nova caixa.

Teorema [Dósa, Sgall, 2013]: First-Fit e Best-Fit são 1.7-aproximativos.
PROVA: um aluno apresentará

Tight Example: First-Fit 1.7-aproximativo (e não menos !!)

- $2K$ grupos em sequência com 5 itens $\left(\frac{1}{6}-\varepsilon\right)$ e 1 item (7ε)
- $5K$ grupos em sequência com 2 itens $\left(\frac{1}{3}-\varepsilon\right)$ e 1 item (7ε)
- $10K$ itens $\left(\frac{1}{2}+2\varepsilon\right)$, onde $0<\varepsilon<\frac{1}{49K}$
- $opt = 10K+1$**:** $10K$ caixas $\{\frac{1}{2}+2\varepsilon, \frac{1}{3}-\varepsilon, \frac{1}{6}-\varepsilon\}$ e 1 caixa com $7K$ itens 7ε
- **First-Fit:** $M = 17K$ (a própria sequência dos itens)
- Fator de aprox.: $M/opt \;=\; 17K/(10K+1) \;\geq\; 1.7 - \frac{1}{5K}$.

Bin Packing não é $(1.5 - \varepsilon)$-aprox. poli, se NP$\neq$P

Teorema:
Bin Packing não tem algoritmo $(1.5 - \varepsilon)$-aprox. poli, a menos que P=NP

PROVA:

- Suponha existe algoritmo poli $(1.5 - \varepsilon)$-aprox $\mathcal{A}$ para Bin Packing. Vamos usar $\mathcal{A}$ p/ decidir PARTIÇÃO em tempo polinomial.
- Instância de PARTIÇÃO: $S = \{s_1, \ldots, s_n\}$ com $s_1 + \ldots s_n = 2$.
- Pergunta: É possível particionar S em 2 conjuntos com soma 1 ?
- Redução para Bin Packing: tome elementos de S como itens no BP. Se couberem em 2 caixas (SIM). Se 3 ou mais caixas (NÃO).
- $\mathcal{A}(S)$: num. caixas solução de $\mathcal{A}$ p/ S. Logo $\mathcal{A}(S) \leq (1.5 - \varepsilon) \cdot opt$.
- Se $\mathcal{A}(S) \geq 3$, então $opt \geq 3$ e então PARTIÇÃO é NÃO.
- Se $\mathcal{A}(S) \leq 2$, então PARTIÇÃO é SIM.
- Com isso, $\mathcal{A}$ pode ser usado para decidir PARTIÇÃO. Logo P=NP.

Técnica do GAP para provar Inaproximabilidade.

First-Fit-Decreasing é assintot. 1.222-aproximativo

Algoritmo First-Fit-Decreasing (FFD): Ordenar os itens do maior para o menor. Depois aplicar o algoritmo First-Fit.

Teorema [Dósa, 2007]: $FFD(I) \leq (11/9) \cdot opt(I) \ + \ 6/9$ para toda instância I de Bin Packing.

PROVA: não será mostrada

Tight Example: $FFD(I) \ = \ (11/9) \cdot opt(I) \ + \ 6/9$

- $opt(I) = 9K + 6$: $6K + 4$ caixas $\{\frac{1}{2} + \varepsilon, \frac{1}{4} + \varepsilon, \frac{1}{4} - 2\varepsilon\}$ e $3K + 2$ caixas $\{\frac{1}{4} + 2\varepsilon, \frac{1}{4} + 2\varepsilon, \frac{1}{4} - 2\varepsilon, \frac{1}{4} - 2\varepsilon\}$
- **Ordem decrescente:** $6K + 4$ itens $(\frac{1}{2} + \varepsilon)$, $6K + 4$ itens $(\frac{1}{4} + 2\varepsilon)$, $6K + 4$ itens $(\frac{1}{4} + \varepsilon)$ e $12K + 8$ itens $(\frac{1}{4} - 2\varepsilon)$
- Solução $FFD(I) = 11K + 8$: $6K + 4$ caixas $\{\frac{1}{2} + \varepsilon, \frac{1}{4} + 2\varepsilon\}$, $2K + 1$ caixas $\{\frac{1}{4} + \varepsilon, \frac{1}{4} + \varepsilon, \frac{1}{4} + \varepsilon\}$, 1 caixa $\{\frac{1}{4} + \varepsilon, \frac{1}{4} - 2\varepsilon, \frac{1}{4} - 2\varepsilon, \frac{1}{4} - 2\varepsilon\}$, $3K + 1$ caixas $\{\frac{1}{4} - 2\varepsilon, \frac{1}{4} - 2\varepsilon, \frac{1}{4} - 2\varepsilon, \frac{1}{4} - 2\varepsilon\}$ e 1 caixa $\{\frac{1}{4} - 2\varepsilon\}$
- Fator de aprox.:
  $(11/9) \cdot opt + 6/9 \ = \ (99K + 72)/9 \ = \ 11K + 8 \ = \ FFD(I)$.

Modified First-Fit-Decreasing é assintot. 1.18333-aprox

Algoritmo Modified-First-Fit-Decreasing (MFFD): Ordena do maior para o menor. Classifica os itens em 4 tipos: grande ($>1/2$), médio ($>1/3$), pequeno ($>1/6$) e muito pequeno ($<1/6$). FFD nos grandes e médios. Depois tenta agrupar 2 pequenos em cada caixa criada. Depois tenta colocar o resto nas caixas criadas. Finalmente, FFD nos itens restantes.

Teorema [Yue, Zhang, 1995]: $MFFD(I) \leq (71/60) \cdot opt(I) \ + \ 1$ para toda instância I de Bin Packing.

PROVA: não será mostrada

PTAS assintótico para Bin-Packing (BP)

PTAS (Polynomial Time Approximation Scheme): Dado $\varepsilon > 0$, algoritmo retorna solução $(1+\varepsilon)\cdot opt$ em tempo polinomial em n.

Combinação com Repetição: Número de soluções de $x_1 + \ldots + x_A = B$ para $x_i \geq 0$.

$$\binom{A+B-1}{A-1} = \binom{A+B-1}{B},$$

que é o número de soluções de $x'_1 + \ldots + x'_A = A + B$ para $x'_i \geq 1$.

Combinação com Repetição: Número de soluções de $x_1 + \ldots + x_A \leq B$ para $x_i \geq 0$.

$$\sum_{b=0}^{B} \binom{A-1+b}{A-1} = \binom{(A-1)+B+1}{(A-1)+1} = \binom{A+B}{A} \leq (A+B)^A$$

PTAS assintótico para Bin-Packing (BP)

Combinação com Repetição: Número de soluções de $x_1 + \ldots + x_A \leq B$ para $x_i \geq 0$.

$$\sum_{b=0}^{B} \binom{A-1+b}{A-1} = \binom{(A-1)+B+1}{(A-1)+1} = \binom{A+B}{A} \leq (A+B)^A$$

Lema: Suponha que os n itens do BP são $\geq \varepsilon$ e que só existem K tipos diferentes de itens. Então existe algoritmo poli exato que resolve BP.

PROVA:

- Número de itens numa caixa é $\leq M = \lceil 1/\varepsilon \rceil$. Numa caixa qquer, seja x_i o número de itens do tipo i. Portanto, $x_1 + \ldots + x_K \leq M$. Logo existem $R \leq (K+M)^M$ soluções possíveis (tipos de caixas).
- O número total de caixas é no máximo n. Em uma solução qualquer do BP, seja y_j o número de caixas do tipo j. Logo, $y_1 + \ldots + y_R \leq n$. Então existem no máximo $P \leq (n+R)^R$ soluções possíveis do BP.

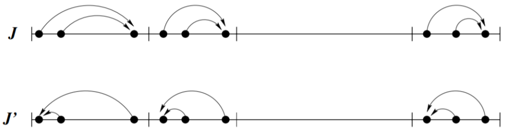
$$P \leq (n+R)^{(K+1/\varepsilon)^{(1/\varepsilon)}} = n^{O((K+1/\varepsilon)^{(1/\varepsilon)})}$$

PTAS assintótico para Bin-Packing (BP)

Lema: Suponha que os n itens do BP são $\geq \varepsilon$. Então existe algoritmo poli aprox. com fator $1 + \varepsilon$.

PROVA: Seja I (instância) o conjunto de itens do BP.

- Ordene os n itens de I em peso crescente e particione-os em $K = \lceil 1/\varepsilon^2 \rceil$ grupos com no máximo $Q = \lfloor n\varepsilon^2 \rfloor$ itens em cada grupo
- Sejam J e J' as instâncias obtidas de I aumentando/diminuindo o peso de cada item para o do maior/menor em seu grupo.
- Solução de J é sol. de I, que é sol. de J': $opt(J') \leq opt(I) \leq opt(J)$
- Solução $opt(J)$ em tempo $n^{O((1/\varepsilon^2)^{(1/\varepsilon)})}$
- Sol. de J' implica uma sol. de J, exceto o grupo mais pesado.
- Logo: $opt(J) \leq opt(J') + Q \leq opt(I) + Q$
- Todo item $\geq \varepsilon \;\Rightarrow\; opt(I) \geq n\varepsilon \;\Rightarrow\; Q = \lfloor n\varepsilon^2 \rfloor \leq \varepsilon \cdot opt(I)$
- $opt(J) \;\leq\; (1 + \varepsilon) \cdot opt(I)$



PTAS assintótico para Bin-Packing (BP)

TEOREMA: Para cada $0 < \varepsilon < 1/2$, existe algoritmo que produz uma solução com $(1+2\varepsilon)opt \; + \; 1$ caixas em tempo $n^{O((1/\varepsilon^2)^{(1/\varepsilon)})}$.

PROVA: Seja I (instância) o conjunto de itens do BP.

- Seja I' a instância I sem itens $< \varepsilon$. Temos algoritmo que obtém $(1+\varepsilon)opt(I')$ caixas. Depois empacotar itens $< \varepsilon$ usando FF.
- Se não foram abertas novas caixas, temos $(1+\varepsilon)opt(I') \leq (1+\varepsilon)opt(I)$ caixas. **OK**
- CC: Seja M o num. caixas do algoritmo. Todas, exceto no máximo uma, estão cheias até $1-\varepsilon$. Logo, a soma dos pesos dos itens é $\geq (M-1)(1-\varepsilon)$. Logo $opt \geq (M-1)(1-\varepsilon)$. Como $\varepsilon < 1/2$:

$$M \;\leq\; \frac{opt}{1-\varepsilon} + 1 \;\leq\; (1+2\varepsilon)opt \; + \; 1. \qquad \textbf{OK}$$

PTAS assintótico para Bin-Packing (BP) - Melhoria

- **Instância:** n itens com pesos $p_1, p_2, \ldots, p_n$ racionais em $(0, 1]$.
- **Objetivo:** Obter o menor número M de caixas (*bins*) de capacidade 1 para empacotar (*packing*) todos os itens.

Lema - Melhoria: Suponha que só existem K tipos diferentes de itens no Bin Packing. Então existe algoritmo poli exato que resolve BP.

PROVA:

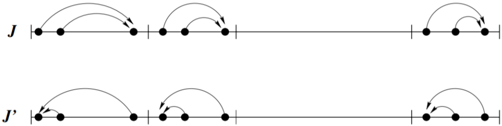
- Seja n_j o número de itens do tipo j: $n_1 + \ldots + n_K = n$
- $BINS(i_1, \ldots, i_K)$: menor núm. caixas p/ i_j itens do tipo j
- Seja $\mathcal{Q}$ conj. das uplas $q = (q_1, \ldots, q_K)$ tq $BINS(q_1, \ldots, q_K) = 1$.
- $|\mathcal{Q}| \leq (n_1 + 1) \cdot \ldots \cdot (n_K + 1) \leq (n + 1)^K \ = \ O(n^K)$
- $BINS(i_1, \ldots, i_K) \ = \ 1 \ + \ \min_{q \in \mathcal{Q}} \{BINS(i_1 - q_1, \ldots, i_K - q_K)\}$
- Recorrência pode ser calculada por Programação Dinâmica
- Cada valor é computado em tempo $O(n^K)$ e existem $O(n^K)$ valores.
- Tempo total: $O(n^{2K})$

PTAS assintótico para Bin-Packing (BP) - Melhoria

Lema: Suponha que os n itens do BP são $\geq \varepsilon$. Então existe algoritmo $(1+\varepsilon)$-aproximativo em tempo polinomial $\boxed{O\left(n^{(2/\varepsilon^2)}\right)}$.

PROVA: Seja I (instância) o conjunto de itens do BP.

- Ordene os n itens de I em peso crescente e particione-os em $K = \lceil 1/\varepsilon^2 \rceil$ grupos com no máximo $Q = \lfloor n\varepsilon^2 \rfloor$ itens em cada grupo
- Sejam J e J' as instâncias obtidas de I aumentando/diminuindo o peso de cada item para o do maior/menor em seu grupo.
- Solução de J é sol. de I, que é sol. de J': $opt(J') \leq opt(I) \leq opt(J)$
- Solução $opt(J)$ em tempo $O\left(n^{(2/\varepsilon^2)}\right)$
- Sol. de J' implica uma sol. de J, exceto o grupo mais pesado.
- Logo: $opt(J) \leq opt(J') + Q \leq opt(I) + Q$
- Todo item $\geq \varepsilon \;\Rightarrow\; opt(I) \geq n\varepsilon \;\Rightarrow\; Q = \lfloor n\varepsilon^2 \rfloor \leq \varepsilon \cdot opt(I)$
- $opt(J) \;\leq\; (1+\varepsilon) \cdot opt(I)$



PTAS assintótico para Bin-Packing (BP) - Melhoria

TEOREMA: Para cada $0 < \varepsilon < 1/2$, existe algoritmo que produz uma solução com $(1 + 2\varepsilon)opt \ + \ 1$ caixas em tempo $\boxed{O\left(n^{(2/\varepsilon^2)}\right)}$.

PROVA: Seja I (instância) o conjunto de itens do BP.

- Seja I' a instância I sem itens $< \varepsilon$. Temos algoritmo que obtém $(1 + \varepsilon)opt(I')$ caixas. Depois empacotar itens $< \varepsilon$ usando FF.
- Se não foram abertas novas caixas, temos $(1 + \varepsilon)opt(I') \leq (1 + \varepsilon)opt(I)$ caixas. **OK**
- CC: Seja M o num. caixas do algoritmo. Todas, exceto no máximo uma, estão cheias até $1 - \varepsilon$. Logo, a soma dos pesos dos itens é $\geq (M - 1)(1 - \varepsilon)$. Logo $opt \geq (M - 1)(1 - \varepsilon)$. Como $\varepsilon < 1/2$:

$$M \ \leq \ \frac{opt}{1 - \varepsilon} + 1 \ \leq \ (1 + 2\varepsilon)opt \ + \ 1. \qquad \textbf{OK}$$

Conclusão (até agora)

* **Escalonamento:** Algoritmo 2-aproximativo.
* **TSPM:** Alg 1.5-aproximativo. Parece ser o melhor possível, na prática.
* **Weighted Set Cover:** Algoritmo ($\ln n+1$)-aprox. Parece o melhor possível (parece não ter aprox para fator constante).
* **Set Cover:** Algoritmos ($\ln n+1$)-aprox. e f-aprox. Parecem os melhores possíveis (parece não ter aprox para fator constante).
* **Vertex Cover:** Algoritmo 2-aprox. Parece o melhor possível.
* **Mochila:** FPTAS. É o melhor possível.
* **Bin Packing:** PTAS assintótico (parece não ter FPTAS assintótico).

- Problemas NP-Difíceis com FPTAS (Mochila)
- Problemas NP-Difíceis com PTAS, prov **sem** FPTAS (Bin Packing*)
- Problemas NP-Difíceis com α-aprox (p/ α const), provav **sem** PTAS (TSP Métrico, Vertex Cover, BIN PACKING)
- Problemas NP-Difíceis com $\alpha(n)$-aprox (p/ $\alpha(n)$ **não** const), que provav **não** tem para α const (Set Cover)
- Problemas NP-Difíceis que provav **não** tem alg poli $\alpha(n)$-aprox para nenhuma função computável $\alpha(n)$ poli (Caixeiro viajante)