Algoritmos de Aproximação para o Problema de Empacotamento

Felipe Aureliano, Lucas Moreira e Raquel Kitazume

ICMC - Universidade de São Paulo - São Carlos

Tópicos em Otimização

02 de Novembro de 2015

Sumário

O problema de Empacotamento Unidimensional

Dada uma sequência de itens $L = (a_1, a_2, ..., a_n)$, onde cada item possui um tamanho $s(a_i) \in (0, 1]$, devemos alocá-los em um número mínimo possível de recipientes, onde cada recipiente possui capacidade 1. Matematicamente, temos

$$\sum_{a_i \in B_j} s(a_i) \leq 1, 1 \leq j \leq m$$

onde m é o número mínimo de subconjuntos $B_1, B_2, ..., B_m$.

O problema de Empacotamento Unidimensional

O *problema de empacotamento unidimensional* é clássico na literatura e possui um vasto número de aplicações no mundo real. Como por exemplo, ele pode ser usado em:

- Alocação de memória em sistemas paginados;
- Atribuição de artigos em páginas de jornais;
- Alocação de comerciais de televisão em intervalos comerciais;

O problema de Empacotamento Unidimensional

Também, é um problema importante do ponto de vista teórico, já que serviu de base para vários métodos clássicos analisarem o desempenho de algoritmos de aproximação, ou seja, é um dos primeiros problemas com publicações fazendo comparações de razões de pior caso, análise de comportamento de caso médio e identificação de limitantes na melhor performance online entre algoritmos.

O problema de Empacotamento Unidimensional

Por fim, sabe-se que este problema pertence à classe NP-difícil e, a menos que P seja igual a NP, não há algoritmo que determine uma solução ótima em tempo polinomial. Em vista disso, a busca por heurística e algoritmos de aproximação se destacou a partir da década de 70.

Algoritmo First Fit Decreasing (FFD)

O algoritmo FFD consiste do seguinte:

1. Os itens da lista $L = (a_1, a_2, ..., a_n)$ são organizados em ordem não crescente de tamanho;
2. Aloca-se cada item a_i no primeiro recipiente (aquele de menor índice) capaz de acomodá-lo;
3. Caso não exista tal recipiente, um novo é criado e o item é alocado no novo recipiente.

Razão de Aproximação

Teorema

Para qualquer lista L, a solução dada pelo algoritmo FFD é sempre menor ou igual a $\frac{3}{2}$ da solução ótima para L.

Demonstração: Indexe os recipientes na ordem em que eles são abertos, $\{B_1, B_2, ..., B_n\}$. Seja $N_{FFD}(L)$ o número de recipientes utilizados pelo algoritmo FFD para empacotar a lista L e seja da seguinte forma $N_{FFD}(L) = 3x + p$, para algum inteiro $x, x \geq 0$ e $p = 0, 1, 2$. Devemos considerar três casos que dependem do valor de p.

Razão de Aproximação

Caso 1: $p = 0$ ou $p = 1$.

Tome o recipiente de índice $2x + 1$, isto é, B_{2x+1}. Como os itens são ordenados por tamanho, se este recipiente contêm um item grande, então $OPT(L) \geq 2x + 1 > \frac{2}{3} N_{FFD}(L)$.

Caso contrário, nos recipientes B_{2x+1} a B_{3x+p} devem haver, pelo menos, $2x + 2p - 1$ itens pequenos e nenhum destes cabendo nos primeiros $2x$ recipientes. Assim, a soma dos tamanhos de todos os itens, $\sum_{i=1}^{n} s(a_i)$, é maior que $\min\{2x, 2x + 2p - 1\}$ e assim, maior também que $2x + p - 1$, já que p é ou 0 ou 1. Assim, temos que $OPT(L) \geq 2x + p \geq \frac{2}{3} N_{FFD}(L)$.

Caso 2: $p = 2$.

De forma análoga, suponha $p = 2$, ou seja, $N_{FFD}(L) = 3x + 2$. Se o recipiente $2x + 2$ conter um item grande, temos $OPT(L) \geq 2x + 2 > \frac{2}{3}N_{FFD}(L)$. Caso contrário, os recipientes $2x + 2$ a $3x + 2$ contém pelo menos $2x + 1$ itens pequenos, nenhum deles cabendo nos $2x + 1$ recipientes anteriores, implicando que a soma de seus tamanhos é maior que $2x + 1$ e assim, $OPT(L) \geq 2x + 2 > \frac{2}{3}N_{FFD}(L)$.

Portanto, fica demonstrado que a razão de aproximação do algoritmo FFD é $\frac{3}{2}$.□

Algoritmo ZCW

1. Coloque cada item grande em um recipiente. Indexe os recipientes de forma arbitrária e configure-os como ativos.
2. Para os itens pequenos, se há algum recipiente ativo aberto, coloque o item atual a_i no recipiente ativo aberto (menor índice), caso ele caiba. Caso contrário, feche este recipiente ativo e considere o recipiente *extra* da seguinte forma:

Algoritmo ZCW

- Se há algum recipiente *extra* aberto com espaço suficiente para a_i, aloque a_i nele.
- Caso contrário, feche este recipiente *extra*. Abra outro recipiente para a_i e configure ele como *extra*.

3. Se não há recipiente ativo aberto, crie um novo recipiente para a_i e configure-o como ativo.

Exemplo ZCW

Exemplo:

Seja $L = (a_1, a_2, ..., a_8)$, tal que $s(a_1) = 0.8$, $s(a_2) = 0.6$, $s(a_3) = 0.1$, $s(a_4) = 0.9$, $s(a_5) = 0.4$, $s(a_6) = 0.4$, $s(a_7) = 0.5$ e $s(a_8) = 0.3$.

Razão de Aproximação

Teorema

O algoritmo ZCW tem razão de aproximação igual a $\frac{3}{2}$*. Além disso,* $\frac{3}{2}$ *é o melhor possível a menos que* $P = NP$*.*

Demonstração: Sejam:

X: conjunto que contém os recipientes ativos;

Y: conjunto que contém os recipientes extras;

N_a: número de recipientes de X;

N_e: número de recipientes de Y;

N_f: número de recipientes ativos fechados;

Razão de Aproximação

N_f = número total de itens empacotados nos recipientes extras

Pois um recipiente ativo é fechado quando um item não pode ser alocado nele, e então, este item é colocado em um recipiente extra. Observe também que todos os recipientes extras, com exceção do último, possuem no mínimo dois itens, já que recipientes extras contém apenas itens pequenos. Logo,

$$N_e \leq \lceil \frac{N_f}{2} \rceil.$$

Agora, uma vez que N_f é no máximo igual a N_a, então há dois possíveis casos, mostrados a seguir.

Razão de Aproximação

Caso 1: $N_f \leq N_a - 1$.

Nesse caso,

$$ZCW(L) = N_a + N_e \leq N_a + \lceil \frac{N_f}{2} \rceil \leq N_a + \lceil \frac{N_a - 1}{2} \rceil \leq N_a + \frac{N_a}{2} = \frac{3N_a}{2}.$$

Se $N_f = N_a - 1$,

$$OPT(L) \geq \sum_{i=1}^{n} s(a_i) = \sum_{a_i \in X} s(a_i) + \sum_{a_i \in Y} s(a_i) > N_f = N_a - 1,$$

isto é,

$$OPT(L) \geq N_a.$$

Razão de Aproximação

Se $N_f \leq N_a - 2$, então existem no mínimo dois recipientes ativos não fechados. Isso acontece somente quando cada recipiente ativo contém um item grande, ou seja, não foi necessário criar recipiente ativo para alocar qualquer um dos itens pequenos. Assim, $\text{OPT}(L) \geq N_a$.

$$\text{Logo, } \frac{ZCW(L)}{OPT(L)} \leq \frac{3}{2}.$$

Razão de Aproximação

Caso 2: $N_f = N_a$. Nesse caso,

$$ZCW(L) = N_a + N_e \leq N_a + \lceil \frac{N_f}{2} \rceil = N_a + \lceil \frac{N_a}{2} \rceil \leq \frac{3N_a}{2} + \frac{1}{2}.$$

Por outro lado,

$$OPT(L) \geq \sum_{i=1}^{n} s(a_i) = \sum_{a_i \in X} s(a_i) + \sum_{a_i \in Y} s(a_i) > N_a,$$

isto é,

$$OPT(L) \geq N_a + 1.$$

$$\text{Logo, } \frac{ZCW(L)}{OPT(L)} \leq \frac{3}{2}.$$

Razão de Aproximação

Portanto, fica demonstrado que o algoritmo ZCW tem razão de aproximação igual a $\frac{3}{2}$.

Além disso, em termos de razão de pior caso absoluta, ZCW é o melhor possível para o problema do empacotamento unidimensional a menos que $P = NP$. Isto é, não existe qualquer algoritmo de tempo polinomial para este problema cuja razão seja menor ou igual a $\frac{3}{2}$, a menos que $P = NP$. Tal afirmação pode ser verificada em [1] e [3] . $\square$

Comparação teórica entre FFD e ZCW

Como visto, tanto FFD como ZCW possuem razão de aproximação igual a $\frac{3}{2}$. Logo, a análise da complexidade de tempo e espaço destes algoritmos torna-se fundamental quando o objetivo é descobrir qual deles possui melhor desempenho.

Segundo Johnson [2], o FFD pode ser implementado de forma que leve tempo $O(n \log n)$ se usado uma estrutura de dados adequada.

O espaço que o FFD ocupa é $O(n)$. Isso ocorre pois durante todo o algoritmo todos os recipientes estão abertos.

Comparação teórica entre FFD e ZCW

Já o algoritmo ZCW leva tempo $O(n)$. De fato, pois para cada item i, o algoritmo tenta alocá-lo em um recipiente ativo, do contrário coloca-o em um recipiente extra.

O espaço que ZCW ocupa é constante, pois em cada iteração o algoritmo mantêm aberto no máximo um recipiente ativo e no máximo um extra.

Comparação teórica entre FFD e ZCW

Assim, podemos concluir que, como ambos os algoritmos possuem a mesma razão de aproximação, isto é, $\frac{3}{2}$, a qualidade das soluções de cada algoritmo deve ser muito próxima, porém, ZCW possui um melhor desempenho levando-se em conta a análise da complexidade de tempo e espaço.

Comparação numérica entre FFD e ZCW

Para realizar o testes, foram usadas as instâncias do *benchmark* criado pelo Prof. A. Scholl e Dr. R. Klein [1].

No total haviam 720 instâncias onde em cada instância era possível ter 50, 100, 200 ou 500 itens; as capacidades dos recipientes podiam ser 100, 120 ou 150; e cada objeto podia ter o peso entre 1 a 100, 20 a 100 ou 30 a 100.

Para cada combinação de instância havia 20 casos diferentes.

Comparação numérica entre FFD e ZCW

Os algoritmos foram implementados em linguagem C. Foi realizada a análise de tempo e da quantidade de recipientes. Para isso a estrutura do código foi:

1. Receber o número de itens;
2. Receber a capacidade de cada recipiente;
3. Rodar o algoritmo;
4. Devolver a quantidade de recipientes ou tempo;

Comparação numérica entre FFD e ZCW

Notas:

1. Na implementação do *FFD*, antes do algoritmo os itens foram ordenados usando *quick sort*.
2. Apesar do FFD poder ser implementado em tempo $O(nlogn)$ [2], durante este projeto ele foi implementado em $O(n^2)$.

Comparação númerica entre FFD e ZCW

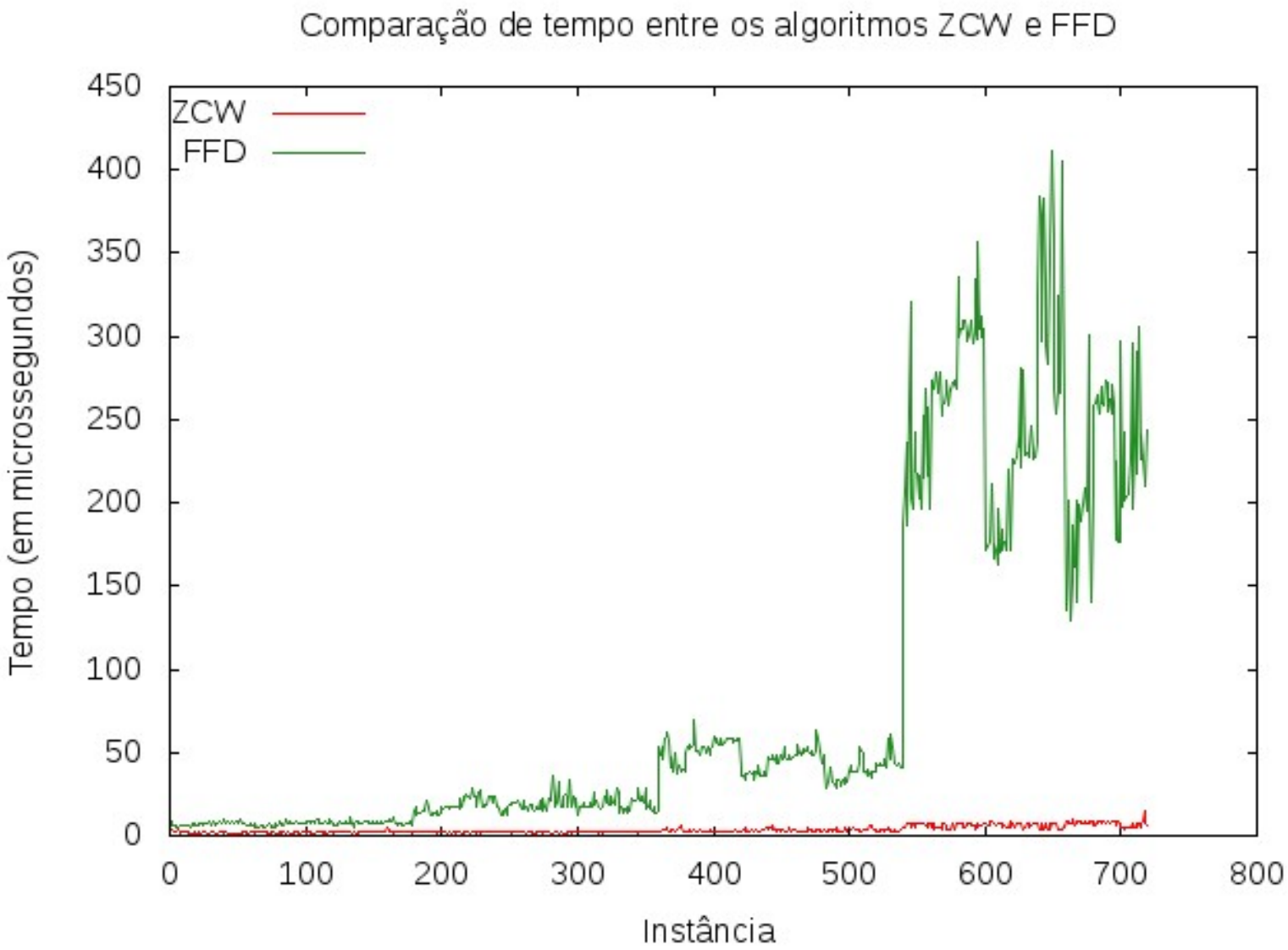
Algoritmo ZCW

1. Recebe a quantidade de itens
2. Recebe a capacidade de cada recipiente
3. Roda o algoritmo
4. Devolve o número de recipientes ou tempo

Algoritmo FFD

1. Recebe a quantidade de itens
2. Recebe a capacidade de cada recipiente
3. Ordenação com *quick sort*
4. Roda o algoritmo
5. Devolve o número de recipientes ou tempo

Comparação numérica entre FFD e ZCW

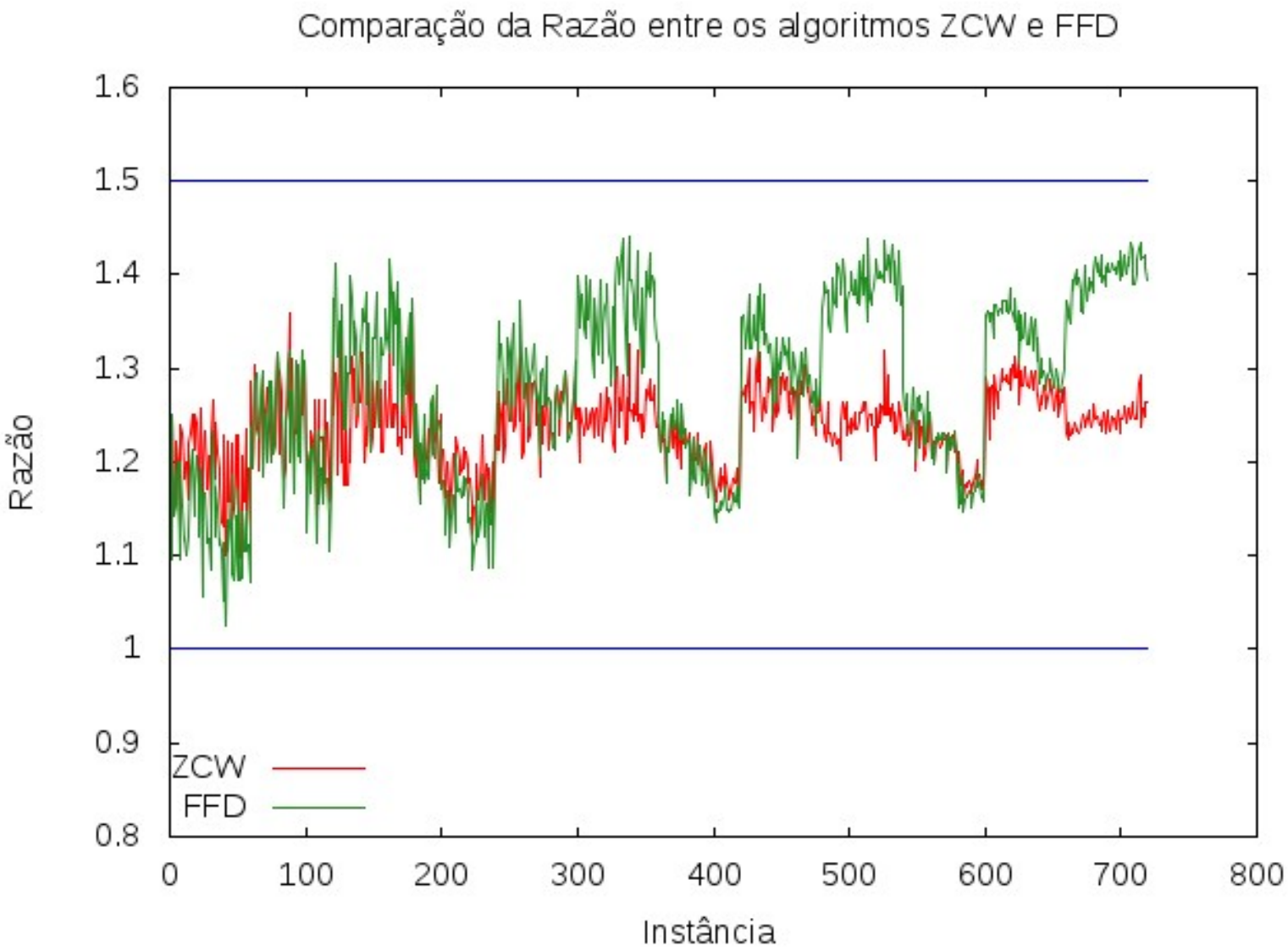


Comparação numérica entre FFD e ZCW

Para analisar estes resultados foi calculada a média de tempo das instâncias com 50, 100, 200 e 500 itens dos dois algoritmos.

	Tempo ZCW	Tempo FFD
50	1,80	7.07
100	2,09	17.60
200	3,36	51.85
500	6,68	240.37

Comparação numérica entre FFD e ZCW



Comparação numérica entre FFD e ZCW

Nas instâncias testadas, o algoritmo ZCW teve razão em média de 1.24 e desvio padrão 0.04, resultando em um coeficiente de variação de 3.35% e o FFD, razão média de 1.28, desvio padrão 0.09 e coeficiente de variação de 7.33%.

Assim, os resultados produzidos pelo FFD têm maior variação que os resultantes do ZCW, ou seja, são mais heterogêneos que os do ZCW. Além disso, a medida que o número de itens aumenta, é possível observar uma pequena melhora do algoritmo ZCW em relação ao FFD.

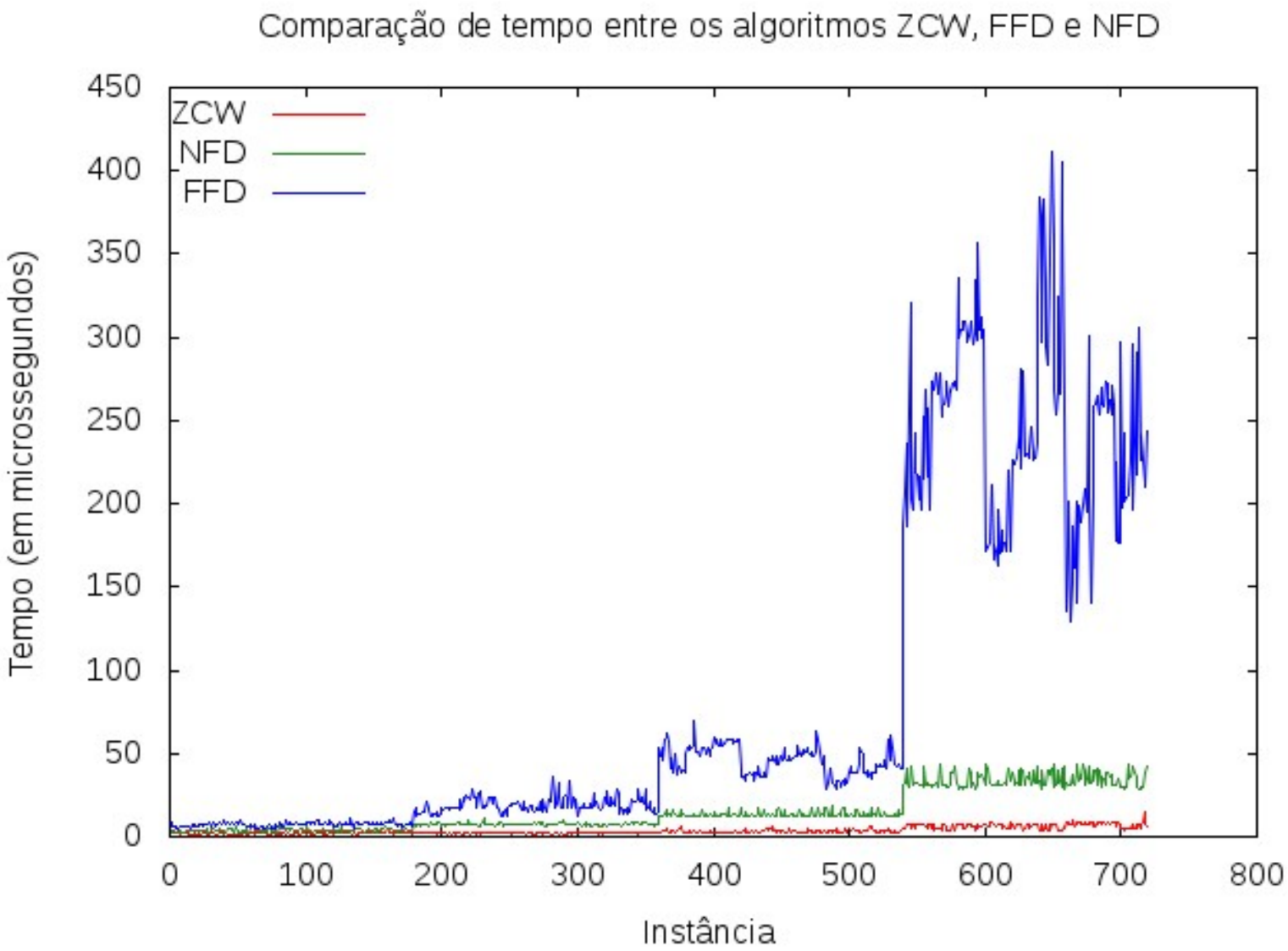
Comparação numérica entre FFD e ZCW

Durante as implementações, foi descoberto que havíamos implementado um outro algoritmo.

O primeiro algoritmo implementado para comparar com o ZCW foi o *Next-Fit* que possui uma ordem de complexidade de tempo $O(n)$.

Mas da mesma forma, os itens também foram ordenados no início do código com o *quick sort*, assim o algoritmo completo ficou com complexidade de tempo $O(n \log n)$.

Comparação numérica entre FFD, ZCW e NFD

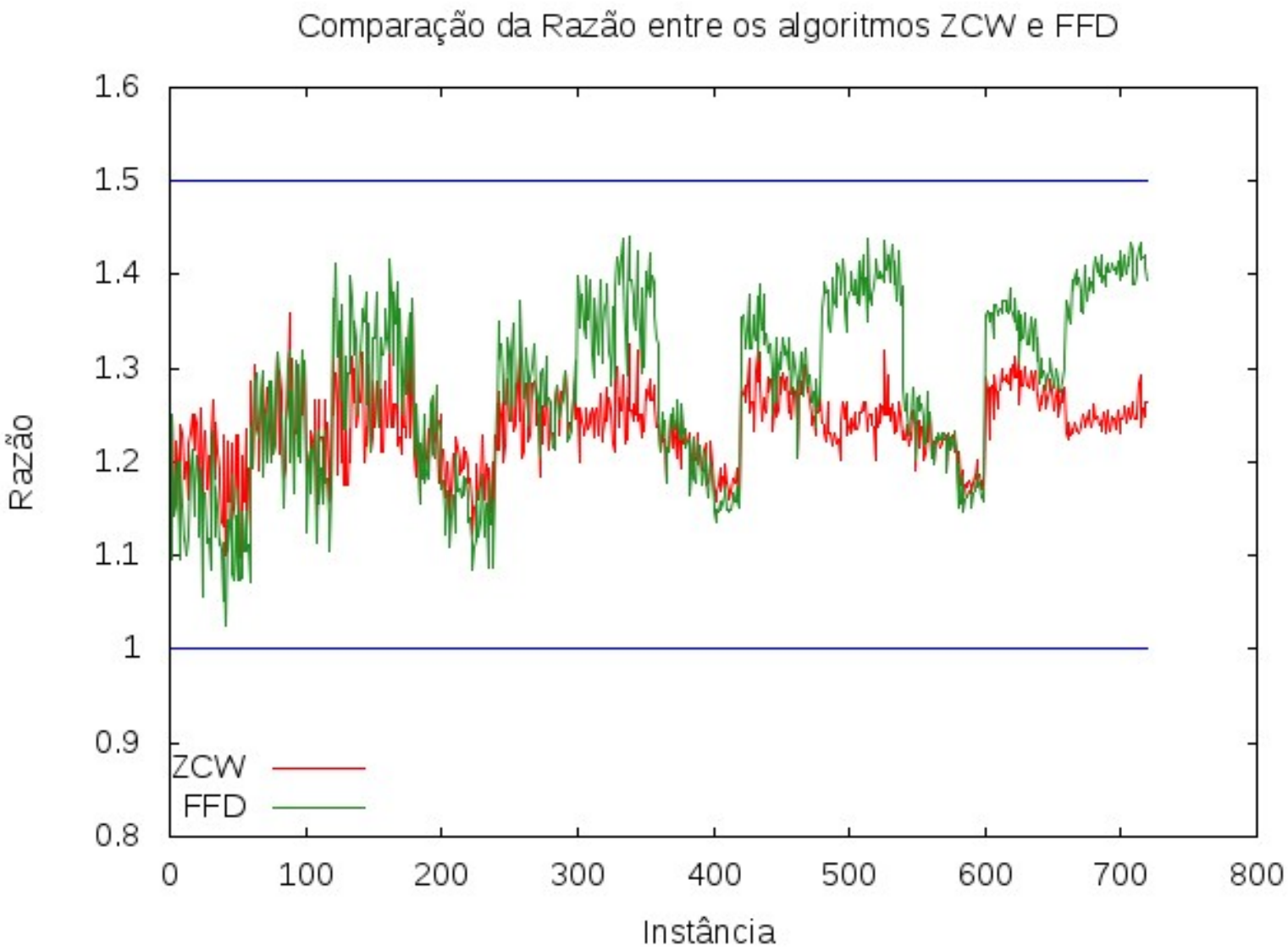


Comparação numérica entre FFD e ZCW

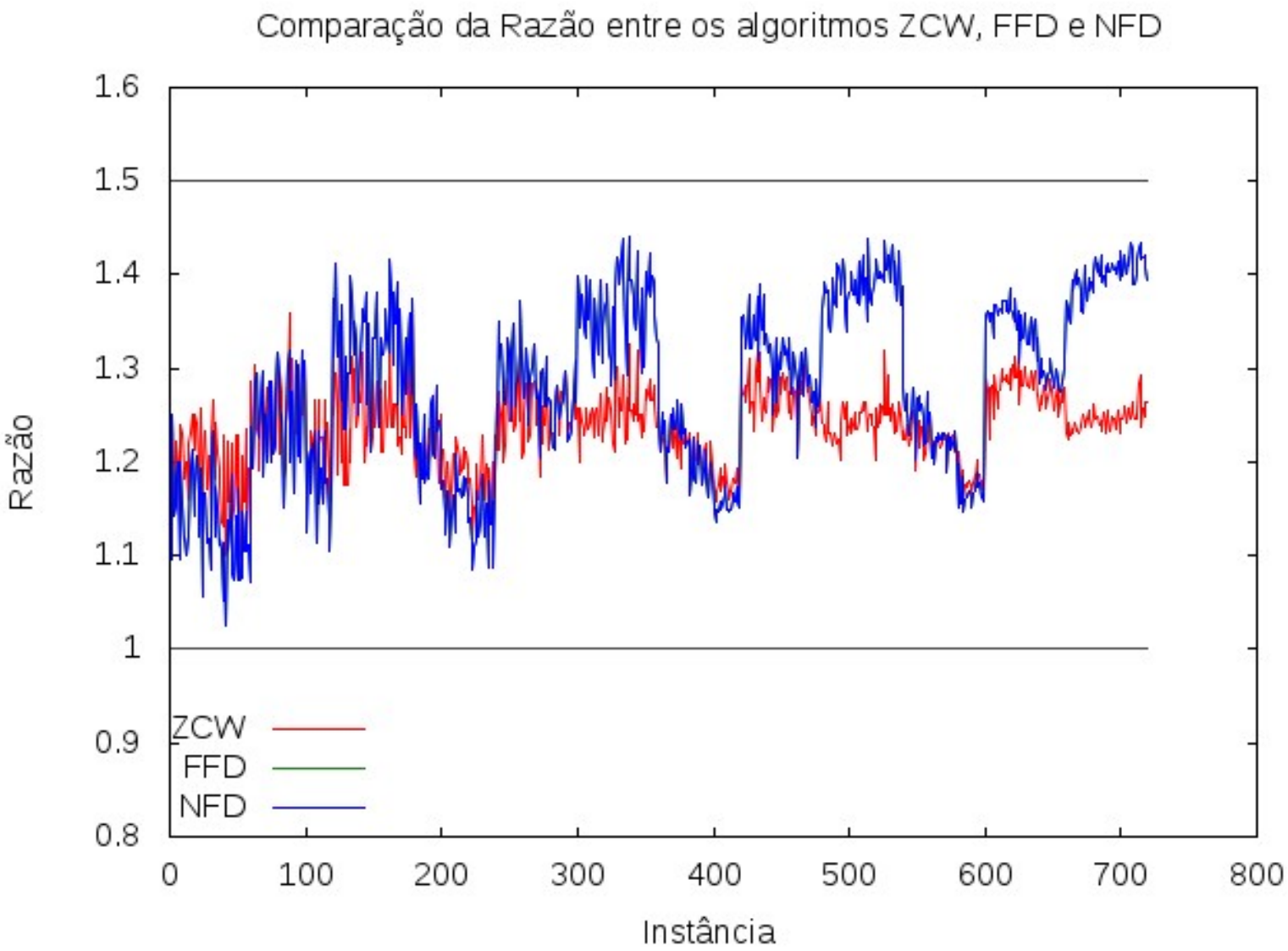
Para analisar estes resultados também foi calculada a média de tempo das instâncias com 50, 100, 200 e 500 itens dos dois algoritmos.

	Tempo ZCW	Tempo NFD	Tempo FFD
50	1,80	4.25	7.07
100	2,09	7.33	17.60
200	3,36	14.39	51.85
500	6,68	34.36	240.37

Comparação numérica entre FFD, ZCW e NFD



Comparação numérica entre FFD, ZCW e NFD



Conclusões

Neste trabalho apresentamos dois algoritmos de aproximação para o problema de empacotamento unidimensional, a saber, ZCW e FFD, e mostramos que a razão de aproximação de ambos é igual a $\frac{3}{2}$.

Na comparação teórica entre os dois algoritmos, analisando a complexidade de tempo e espaço, concluimos que ZCW possui melhor desempenho.

Conclusões

Por fim, em relação à comparação numérica, foi notado que os três algoritmos tiveram seus resultados para cada instância respeitando a razão de aproximação.

Em relação a comparação de tempo, os três algoritmos também respeitaram suas ordens de complexidade teórica.

Assim, o algoritmo que mostrou melhor desempenho nas instâncias testadas foi o ZCW.

Referências

- G. Zhang, X. Cai, C. K. Wong. Linear time-approximation algorithms for bin packing. *Operations Research Letters* 26, pp. 217-222, 2000.
- D. S. Johnson. Fast algorithm for Bin Packing. *Journal of Computer and System Sciences* 8, pp. 272-314, 1974.
- D. S. Johnson, A. Demerson, J. D. Ullman, M. R. Garey, R. L. Graham. Worst-case perfomance bounds for simple one-dimensional packing algorithm. *SIAM J. Comput* 3, pp. 299-325, 1974.
- B. Xia, Z. Tan. Tigther bounds of the First Fit algorithm for the bin-packing problem. *SIAM J. Comput* 3, pp. 299-325, 1974.
- R. E. Korf. A new algorithm for optimal bin packing. *In Eighteenth national conference on Artificial intelligence*, pp. 731-736, 2002.

Referências

- D. Simchi-Levi. New worst-case results for the bin-packing problem. *Naval Research Logistic* 41, pp. 579-585, 1994.
- M. Yue. A simple proof of the inequality $FFD(L) \leq \frac{11}{9}OPT(L) + 1, \forall L$, for the FFD bin-packing algorithm. *Acta. Mathematicae Applicatae Sinica* 7, pp 321-331, 1991.
- G. Dósa, J. Sgall. First Fit bin packing: A tight analysis. *Symposium on Theoretical Aspects of Computer Science* 30, pp. 538-549, 2013.
- E.G. Coffman, M.R. Garey, D.S. Johnson, Approximation algorithms for bin packing: a survey, in: D. Hochbaum (Ed.), *Approximation Algorithms for NP-Hard Problems*, PWS Publishing Company, Boston, pp. 46–93, 1996.
- C. G. Fernandes, F. K. Miyazawa, M. R Cerioli, P. Feofiloff. Uma Introdução Sucinta a Algoritmos de Aproximação, IMPA, 23rd Brazilian Mathematics Colloquium, Rio de Janeiro, Brazil, 2001.

Referências

A. Scholl, R. Klein, C. Jürgens. Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem, *Computers & Operations Research* 24, pp 627-645, 1997.

D. S. Johnson. Near Optimal Bin Packing Algorithms. Massachusetts Institute of Technology, Dept. of Mathematics, pp. 398-399.1973.

J.K. Lenstra, D.B. Shmoys, Computing near-optimal schedules, in: P. Chretienne et al., (Eds.), Scheduling Theory and Its Application, Wiley, New York, 1995.