An Improved FPTAS for 0-1 Knapsack

Ce Jin

Tsinghua University

0-1 Knapsack Problem

Given:

Knapsack capacity $W > 0$

n items
Each item i has *weight* $0 < w_i \leq W$
and *profit* $p_i > 0$

Find a subset of items $I \subseteq [n]$ such that:

- $w(I) := \sum_{i \in I} w_i \leq W$
- $p(I) := \sum_{i \in I} p_i$ is maximized

0-1 Knapsack Problem

Given:

Knapsack capacity $W > 0$

n items

Each item i has *weight* $0 < w_i \leq W$

and *profit* $p_i > 0$

Find a subset of items $I \subseteq [n]$ such that:

- $w(I) \coloneqq \sum_{i \in I} w_i \leq W$
- $p(I) \coloneqq \sum_{i \in I} p_i$ is maximized

A well-known NP-hard problem

FPTAS for 0-1 Knapsack

OPT = optimal total profit $p(I)$

For $\varepsilon > 0$, find a subset $I \subseteq [n]$ such that:

- $w(I) \leq W$
- $p(I) \geq \frac{\text{OPT}}{1+\varepsilon}$

Solvable in $\text{poly}\left(n, \frac{1}{\varepsilon}\right)$ time

Prior Work

FPTAS for 0-1 Knapsack:

- $\tilde{O}(n^3/\varepsilon)$ (textbook algorithm)
- $\tilde{O}(n + \varepsilon^{-4})$ [Ibarra and Kim, 1975]
- $\tilde{O}(n + \varepsilon^{-3})$ [Kellerer and Pferschy, 2004]
- $\tilde{O}(n + \varepsilon^{-2.5})$ [Rhee, 2015]
- $\boldsymbol{\tilde{O}(n + \varepsilon^{-2.4})}$ **[Chan, 2018]**

Prior Work

FPTAS for 0-1 Knapsack:

- $\tilde{O}(n^3/\varepsilon)$ (textbook algorithm)
- $\tilde{O}(n + \varepsilon^{-4})$ [Ibarra and Kim, 1975]
- $\tilde{O}(n + \varepsilon^{-3})$ [Kellerer and Pferschy, 2004]
- $\tilde{O}(n + \varepsilon^{-2.5})$ [Rhee, 2015]
- $\boldsymbol{\tilde{O}(n + \varepsilon^{-2.4})}$ **[Chan, 2018]**
- $\boldsymbol{\tilde{O}(n + \varepsilon^{-2.25})}$ **(this work)**

Conditional Lower Bound:

No $O\left((n + \varepsilon^{-1})^{2-\delta}\right)$ time algorithm,

unless $(\mathbf{min}, +)$ **convolution** has truly subquadratic algo [Cygan, Mucha, Węgrzycki, and Włodarczyk, 2017]

A Special Case

FPTAS for *Subset Sum* ($p_i = w_i$):

- $\tilde{O}\left(\min\left\{\boldsymbol{n} + \boldsymbol{\varepsilon}^{-\mathbf{2}}, n\varepsilon^{-1}\right\}\right)$
  [Kellerer, Mansini, Pferschy, and Speranza, 2003]

A Special Case

FPTAS for *Subset Sum* ($p_i = w_i$):

- $\tilde{O}\left(\min\{\boldsymbol{n} + \boldsymbol{\varepsilon}^{\boldsymbol{-2}}, n\varepsilon^{-1}\}\right)$
  [Kellerer, Mansini, Pferschy, and Speranza, 2003]

Our 0-1 Knapsack algorithm utilizes this result

- Our $\tilde{\boldsymbol{O}}(\boldsymbol{n} + \boldsymbol{\varepsilon}^{-2.25})$ algorithm builds on Chan’s $\tilde{\boldsymbol{O}}(\boldsymbol{n} + \boldsymbol{\varepsilon}^{-2.4})$ algorithm
- Two new ideas

  1. Extending Chan’s number-theoretic technique from two levels to multiple levels.

- Our $\tilde{\boldsymbol{O}}(\boldsymbol{n}+\boldsymbol{\varepsilon}^{-2.25})$ algorithm builds on Chan's $\tilde{\boldsymbol{O}}(\boldsymbol{n}+\boldsymbol{\varepsilon}^{-2.4})$ algorithm
- Two new ideas
  1. Extending Chan's number-theoretic technique from two levels to multiple levels.
  2. A greedy argument ⇒ less computation spent on cheap items (small *unit profit* p_i/w_i)

- Our $\tilde{O}(n + \varepsilon^{-2.25})$ algorithm builds on Chan's $\tilde{O}(n + \varepsilon^{-2.4})$ algorithm
- Two new ideas
  1. Extending Chan's number-theoretic technique from two levels to multiple levels.
  2. A greedy argument ⇒ less computation spent on cheap items (small *unit profit* p_i/w_i)

This talk

- Our $\widetilde{O}(n + \varepsilon^{-2.25})$ algorithm builds on Chan's $\widetilde{O}(n + \varepsilon^{-2.4})$ algorithm
- Two new ideas
  1. Extending Chan's number-theoretic technique from two levels to multiple levels.
  2. A greedy argument ⇒ less computation spent on cheap items (small *unit profit* p_i / w_i)

Preliminaries (based on [Chan, 2018])

Using Chan's Lemmas as blackboxes

This talk

$\widetilde{O}(n + \varepsilon^{-2.333})$ algo

Preliminaries

- Assume $n \leq \mathrm{poly}(\varepsilon^{-1})$.
- Too cheap items $\left(p_i < \frac{\varepsilon}{n} \max_j p_j\right)$ are discarded at the beginning (loss $\leq \varepsilon \cdot \mathrm{OPT}$)

So $\frac{\max p_j}{\min p_j} \leq \mathrm{poly}(\varepsilon^{-1})$

n items, capacity = W
weight $0 < w_i \leq W$ and profit $p_i > 0$

Preliminaries

- "Profit function" (defined over real $x \geq 0$)

$$f_I(x) = \max\{p(J) : J \subseteq I, w(J) \leq x\}$$

$n \leq \text{poly}(\varepsilon^{-1})$ items
profit $\max p_j / \min p_j \leq \text{poly}(\varepsilon^{-1})$
weight $0 < w_i \leq W$

Preliminaries

- "Profit function" (defined over real $x \geq 0$)
  $$f_I(x) = \max\{p(J): J \subseteq I, w(J) \leq x\}$$
- Task: compute a $(1+\varepsilon)$-approximation of profit function f_I,
  $$\widetilde{f_I}(x) \leq f_I(x) \leq (1+\varepsilon)\widetilde{f_I}(x), \forall x \geq 0$$

$n \leq \text{poly}(\varepsilon^{-1})$ items
profit $\max p_j / \min p_j \leq \text{poly}(\varepsilon^{-1})$
weight $0 < w_i \leq W$

Preliminaries

- "Profit function" (defined over real $x \geq 0$)
$$f_I(x) = \max\{p(J): J \subseteq I, w(J) \leq x\}$$
- Task: compute a $(1+\varepsilon)$-approximation of profit function f_I,
$$\widetilde{f_I}(x) \leq f_I(x) \leq (1+\varepsilon)\widetilde{f_I}(x), \forall x \geq 0$$
- For disjoint sets I, J of items,
$$f_{I \cup J}(x) = (f_I \oplus f_J)(x) \coloneqq \max_{0 \leq y \leq x} \big(f_I(y) + f_J(x-y)\big)$$

$n \leq \mathrm{poly}(\varepsilon^{-1})$ items
profit $\max p_j / \min p_j \leq \mathrm{poly}(\varepsilon^{-1})$
weight $0 < w_i \leq W$

Preliminaries

- "Profit function" (defined over real $x \geq 0$)
$$f_I(x) = \max\{p(J): J \subseteq I, w(J) \leq x\}$$
- Task: compute a $(1+\varepsilon)$-approximation of profit function f_I,
$$\widetilde{f_I}(x) \leq f_I(x) \leq (1+\varepsilon)\widetilde{f_I}(x), \forall x \geq 0$$
- For disjoint sets I, J of items,
$$f_{I \cup J}(x) = (f_I \oplus f_J)(x) := \max_{0 \leq y \leq x} \left(f_I(y) + f_J(x-y)\right)$$
- $\widetilde{f_I} \oplus \widetilde{f_J}$ is a $(1+\varepsilon)$-approximation of $f_{I \cup J}$

$n \leq \mathrm{poly}(\varepsilon^{-1})$ items
profit $\max p_j / \min p_j \leq \mathrm{poly}(\varepsilon^{-1})$
weight $0 < w_i \leq W$

Preliminaries

- A nondecreasing step function f has a $(1+\varepsilon)$-approx. with only $\tilde{O}(1/\varepsilon)$ steps (by rounding down to powers of $(1+\varepsilon)$)

$n \le \mathrm{poly}(\varepsilon^{-1})$ items
profit $\max p_j / \min p_j \le \mathrm{poly}(\varepsilon^{-1})$

Preliminaries

- A nondecreasing step function f has a $(1+\varepsilon)$-approx. with only $\tilde{O}(1/\varepsilon)$ steps (by rounding down to powers of $(1+\varepsilon)$)
- "Merging Lemma": Computing (a $(1+\varepsilon)$-approx. of) $f_1 \oplus \cdots \oplus f_m$ takes $\tilde{O}(m/\varepsilon^2)$ time.

($\log m$ depth binary tree. $\varepsilon' \coloneqq \varepsilon / \log m$)

$$f \oplus g \coloneqq \max_{0 \le y \le x} \big(f(y) + g(x-y)\big)$$

Preliminaries

- Divide items into $O(\log 1/\varepsilon)$ groups (group k: $p_i \in [2^k, 2^{k+1}]$)

$n \leq \mathrm{poly}(\varepsilon^{-1})$ items
profit $\max p_j / \min p_j \leq \mathrm{poly}(\varepsilon^{-1})$
weight $0 < w_i \leq W$
Merging $f_1 \oplus \cdots \oplus f_m$: $\tilde{O}(m/\varepsilon^2)$ time.

Preliminaries

- Divide items into $O(\log 1/\varepsilon)$ groups (group k: $p_i \in [2^k, 2^{k+1}]$)
- Compute all f_k and merge them in $\tilde{O}(1/\varepsilon^2)$ time

$n \le \mathrm{poly}(\varepsilon^{-1})$ items
profit $\max p_j / \min p_j \le \mathrm{poly}(\varepsilon^{-1})$
weight $0 < w_i \le W$
Merging $f_1 \oplus \cdots \oplus f_m$: $\tilde{O}(m/\varepsilon^2)$ time.

Preliminaries

- Divide items into $O(\log 1/\varepsilon)$ groups (group k: $p_i \in [2^k, 2^{k+1}]$)
- Compute all f_k and merge them in $\tilde{O}(1/\varepsilon^2)$ time
- Now assume $p_i \in [1,2]$

$n \leq \text{poly}(\varepsilon^{-1})$ items
profit $\max p_j / \min p_j \leq \text{poly}(\varepsilon^{-1})$
weight $0 < w_i \leq W$
Merging $f_1 \oplus \cdots \oplus f_m$: $\tilde{O}(m/\varepsilon^2)$ time.

Preliminaries

- Simple greedy (sort by unit profit $\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \cdots$) approximates with additive error $d \leq \max p_i = O(1)$

profit $p_i \in [1,2]$
weight $0 < w_i \leq W$

Preliminaries

- Simple greedy (sort by unit profit $\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \cdots$) approximates with additive error $d \leq \max p_i = O(1)$
- For $f(w) \geq \Omega(\varepsilon^{-1})$, this is $1 + O(\varepsilon)$ multiplicative approx.

profit $p_i \in [1,2]$
weight $0 < w_i \leq W$

Preliminaries

- Simple greedy (sort by unit profit $\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \cdots$) approximates with additive error $d \leq \max p_i = O(1)$
- For $f(w) \geq \Omega(\varepsilon^{-1})$, this is $1 + O(\varepsilon)$ multiplicative approx.
- Only need to $1 + O(\varepsilon)$ approximate $\mathbf{min}\{\boldsymbol{B}, \boldsymbol{f_I}\}$ for $\boldsymbol{B} = \boldsymbol{O}(\boldsymbol{\varepsilon}^{-1})$!

profit $p_i \in [1,2]$
weight $0 < w_i \leq W$

Preliminaries

- Round p_i down to $1, 1+\varepsilon, 1+2\varepsilon, \dots, 2-\varepsilon$
  $(1+\varepsilon)$ multiplicative error
- Only $m = O(1/\varepsilon)$ different p_i's!

$p_i \in [1,2]$

Preliminaries

- Round p_i down to $1, 1+\varepsilon, 1+2\varepsilon, \ldots, 2-\varepsilon$
  $(1+\varepsilon)$ multiplicative error
- Only $m = O(1/\varepsilon)$ different p_i's!
- Collect all items with the **same profit $\boldsymbol{p}$**. Then f_p can be computed by simple greedy (sort $w_1 \le w_2 \le \cdots$)

$p_i \in [1,2]$

Recap

$p_i \in [1,2]$

Profit functions $f_1, \ldots, f_m$ obtained by simple greedy (one for every p_i) $(m = O(1/\varepsilon)$)

Task: $1 + O(\varepsilon)$ approximate

$$\min\{B, f_1 \oplus \cdots \oplus f_m\} \quad (B = O(\varepsilon^{-1}))$$

Lemma: Merging $f_1 \oplus \cdots \oplus f_m$ in $\tilde{O}(m/\varepsilon^2)$ time.

(Immediately gives $\tilde{O}(n + \varepsilon^{-3})$ algo)

Chan's results

$\tilde{O}\left(\varepsilon^{-1}\sqrt{B}m\right)$ algo
(faster when ***B* small**)

↓

$\tilde{O}(\varepsilon^{-4/3}n + \varepsilon^{-2})$ algo
(faster when ***n* small**)

n items with $m = O(\varepsilon^{-1})$ distinct profit values $p_i \in [1,2]$
Task: $1 + O(\varepsilon)$ approximate

$$\min\{B, f_1 \oplus \cdots \oplus f_m\}$$

$(B = O(\varepsilon^{-1}))$

A Greedy Lemma

If
- the items can be divided into two groups H, L with a large enough gap between their unit profits,

$$\max_{\ell \in L} p_\ell / w_\ell \leq (1-\alpha) \cdot \min_{h \in H} p_h / w_h$$

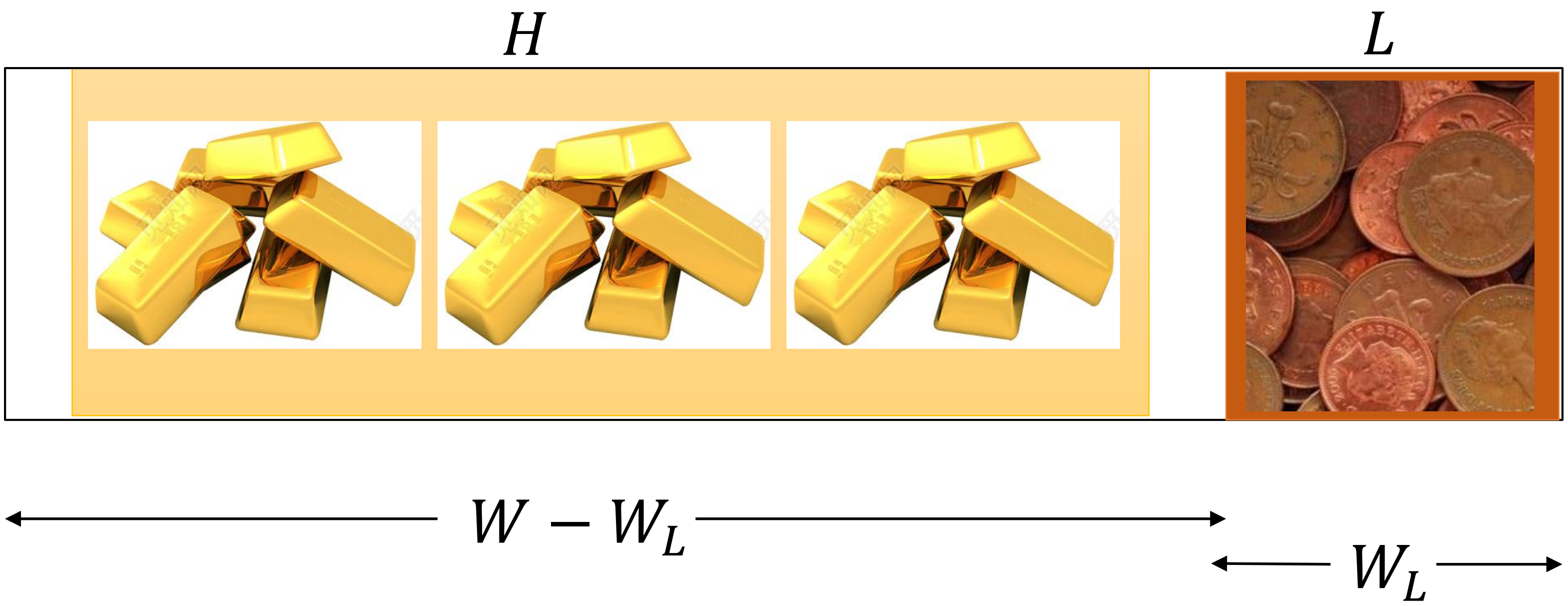
- and group H is large enough,

$$\sum_{h \in H} w_h > W$$

Then

- In an optimal solution, L-items contribute total profit $\leq 2/\alpha$

$p_i \in [1,2]$
capacity = W



- Suppose the optimal solution is

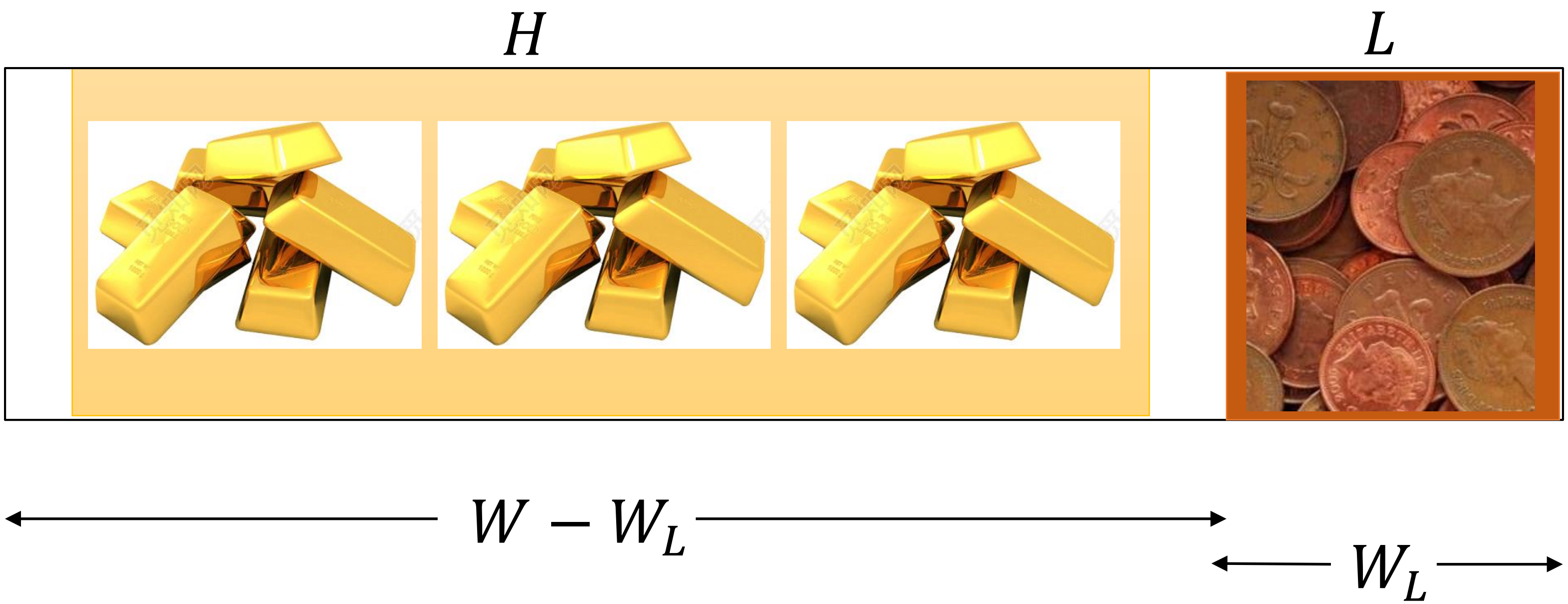
$$f_{H\cup L}(W) = f_H(W - W_L) + f_L(W_L)$$

$p_i \in [1,2]$

capacity = W

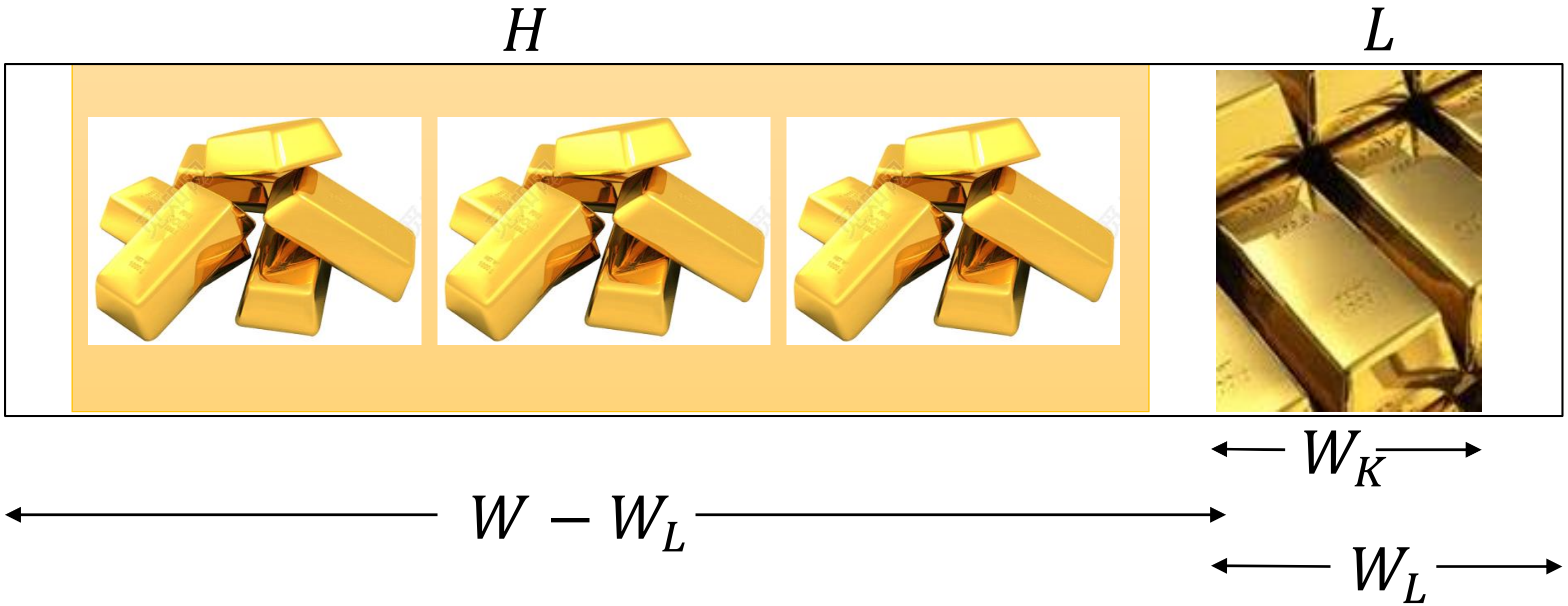
$\max_{\ell \in L} p_\ell / w_\ell \leq (1-\alpha) \cdot \min_{h \in H} p_h / w_h$

$\sum_{h \in H} w_h > W$



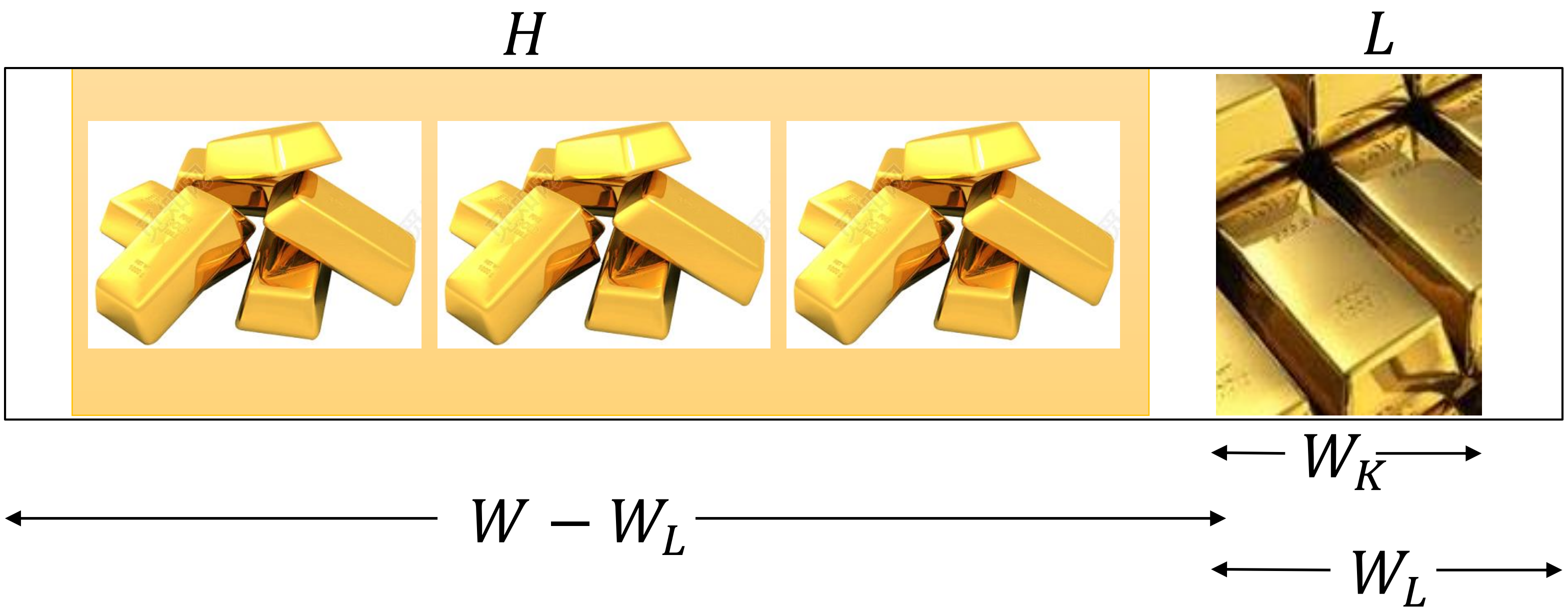
- Remove all L-items, and insert more H-items (denoted by K) to fill in the W_L space

$p_i \in [1,2]$
capacity = W
$\max_{\ell \in L} p_\ell / w_\ell \leq (1-\alpha) \cdot \min_{h \in H} p_h / w_h$
$\sum_{h \in H} w_h > W$



- Remove all L-items, and insert more H-items (denoted by K) to fill in the W_L space
- Until $W_L - W_K < \max_{h \in H} w_h$

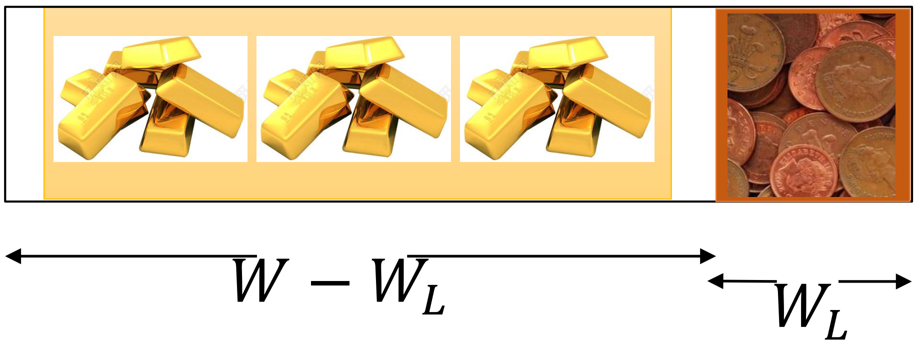
$p_i \in [1,2]$
capacity = W
$\max_{\ell \in L} p_\ell / w_\ell \leq (1 - \alpha) \cdot \min_{h \in H} p_h / w_h$
$\boldsymbol{\sum_{h \in H} w_h > W}$



- Remove all L-items, and insert more H-items (denoted by K) to fill in the W_L space
- Until $W_L - W_K < \max_{h \in H} w_h \leq \mathbf{2/q}$

$\boldsymbol{p_i} \in [\mathbf{1, 2}]$
capacity = W
$\max_{\ell \in L} p_\ell / w_\ell \leq (1-\alpha) \cdot \min_{h \in H} \boldsymbol{p_h / w_h} = (1-\alpha) \cdot \boldsymbol{q}$
$\sum_{h \in H} w_h > W$

tot profit $\geq f_H\,(W - W_L) + qW_K$



optimal sol:

$f_{H\cup L}(W) = f_H(W - W_L) + f_L(W_L)$

$p_i \in [1,2]$

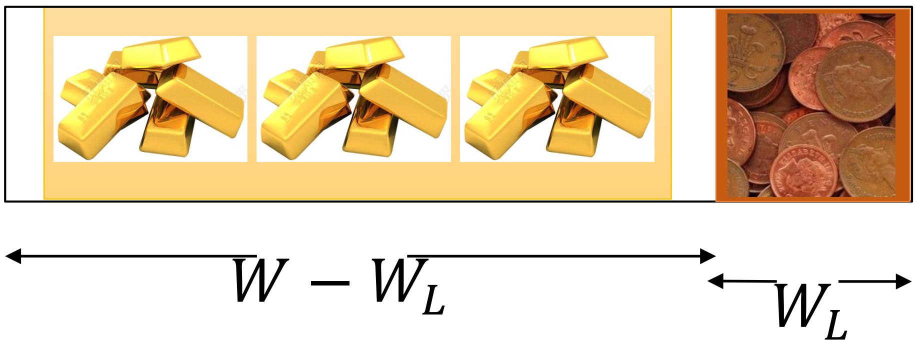
capacity = W

$\max_{\ell\in L} p_\ell / w_\ell \leq (1-\alpha)\cdot \min_{h\in H} p_h/w_h = (1-\alpha)\cdot q$

$\sum_{h\in H} w_h > W$

$W_L - W_K < 2/q$

tot profit $\geq f_H\,(W - W_L) + qW_K$



optimal sol:

$$f_{H \cup L}(W) = f_H(W - W_L) + f_L(W_L)$$

$$\boldsymbol{qW_K \leq f_L(W_L)}$$

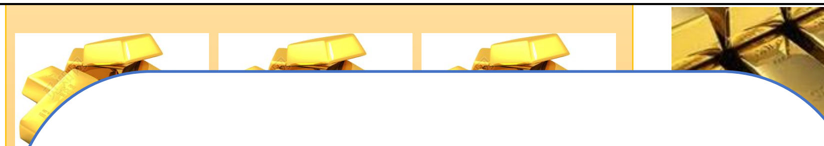
$p_i \in [1,2]$

capacity = W

$\max_{\ell \in L} p_\ell / w_\ell \leq (1-\alpha) \cdot \min_{h \in H} \boldsymbol{p_h / w_h} = (1-\alpha) \cdot \boldsymbol{q}$

$\sum_{h \in H} w_h > W$

$W_L - W_K < 2/q$



$$(1-\alpha)\cdot q\cdot W_L \geq f_L(W_L)$$
$$f_L(W_L) \geq qW_K$$
$$qW_K > qW_L - 2$$
$$\downarrow$$
$$(1-\alpha)\cdot qW_L > qW_L - 2$$

tot profit $\geq f_H\ (W - W_L) + qW_K$

optimal sol:
$f_{H\cup L}(W) = f_H(W - W_L) + f_L(W_L)$

$$\boldsymbol{qW_K \leq f_L(W_L)}$$

$p_i \in [1,2]$
capacity = W
$\max_{\ell\in L} p_\ell/w_\ell \leq (1-\alpha)\cdot \boldsymbol{\min_{h\in H} p_h/w_h} = (1-\alpha)\cdot \boldsymbol{q}$
$\sum_{h\in H} w_h > W$
$\boldsymbol{W_L - W_K < 2/q}$

$$(1-\alpha)\cdot q \cdot W_L \geq f_L(W_L)$$
$$f_L(W_L) \geq qW_K$$
$$qW_K > qW_L - 2$$
$$\downarrow$$
$$(1-\alpha)\cdot qW_L > qW_L - 2$$
$$\downarrow$$
$$\alpha q W_L < 2$$

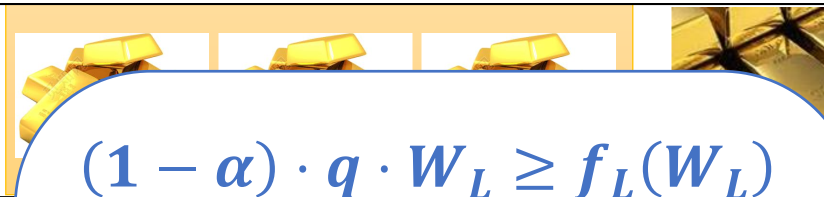
tot profit $\geq f_H\,(W - W_L) + qW_K$

optimal sol:

$f_{H\cup L}(W) = f_H(W - W_L) + f_L(W_L)$

$$\boldsymbol{qW_K \leq f_L(W_L)}$$

$p_i \in [1,2]$
capacity = W
$\max_{\ell\in L} p_\ell/w_\ell \leq (1-\alpha)\cdot \boldsymbol{\min_{h\in H} p_h/w_h} = (1-\alpha)\cdot \boldsymbol{q}$
$\sum_{h\in H} w_h > W$
$\boldsymbol{W_L - W_K < 2/q}$



$$(1-\alpha)\cdot q\cdot W_L \geq f_L(W_L)$$

$$f_L(W_L) \geq qW_K$$

$$qW_K > qW_L - 2$$

$$\downarrow$$

$$(1-\alpha)\cdot qW_L > qW_L - 2$$

$$\downarrow$$

$$\alpha q W_L < 2$$

$$\downarrow$$

$$f_L(W_L) < 2/\alpha$$

tot profit $\geq f_H\ (W - W_L) + qW_K$

optimal sol:

$f_{H\cup L}(W) = f_H(W - W_L) + f_L(W_L)$

$$qW_K \leq f_L(W_L)$$

$p_i \in [1,2]$

capacity = W

$\max_{\ell\in L} p_\ell / w_\ell \leq (1-\alpha)\cdot \min_{h\in H} p_h/w_h = (1-\alpha)\cdot q$

$\sum_{h\in H} w_h > W$

$W_L - W_K < 2/q$

Recap

If: $\max_{\ell \in L} p_\ell / w_\ell \leq (1-\alpha) \cdot \min_{h \in H} p_h / w_h$

$\sum_{h \in H} w_h > W$

Then: In optimal solution of $f_{H \cup L}(W)$, L-items contribute total profit $\leq 2/\alpha$

Task: $1 + O(\varepsilon)$ approximate $\min\{B,\ f_I\}, B = O(\varepsilon^{-1})$

$\tilde{O}\left(\varepsilon^{-1}\sqrt{B}m\right)$ algo (Chan)

(faster when ***B* small**)

$\tilde{O}(\varepsilon^{-4/3}n + \varepsilon^{-2})$ algo (Chan)

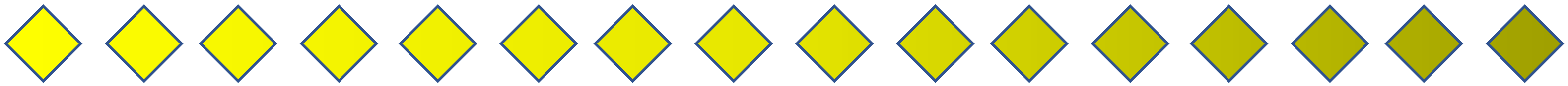
(faster when ***n* small**)

$m = O(\varepsilon^{-1})$ distinct values

$p_i \in [1,2]$

Improved Algorithm

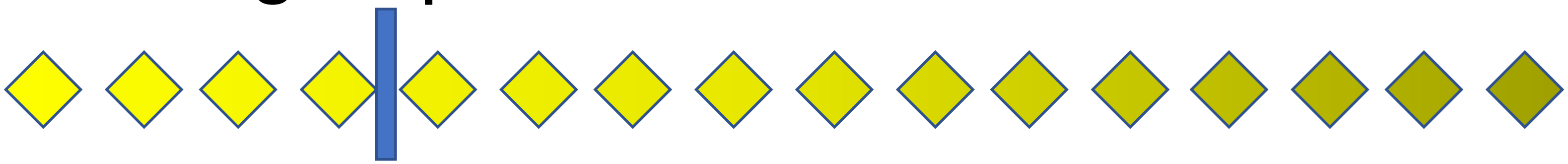
- Sort the items by p_i/w_i, and divide into three groups



$p_i \in [1,2]$

Improved Algorithm

- Sort the items by p_i/w_i, and divide into three groups

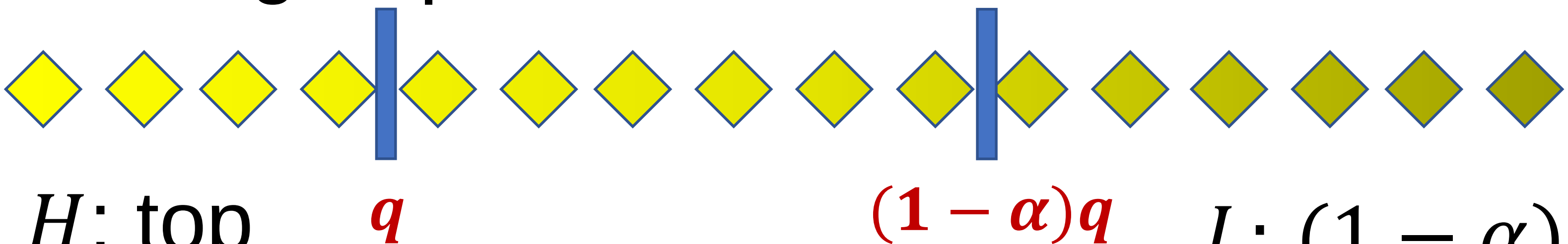


H: top $\Theta(\varepsilon^{-1})$ items

$p_i \in [1,2]$

Improved Algorithm

- Sort the items by p_i/w_i, and divide into three groups



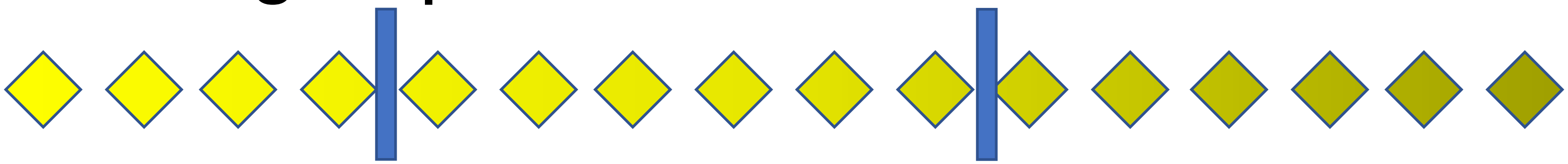
H: top $\Theta(\varepsilon^{-1})$ items

L: $(1-\alpha)$ multiplicative gap

$p_i \in [1,2]$

Improved Algorithm

- Sort the items by p_i/w_i, and divide into three groups



q $(1-\alpha)q$

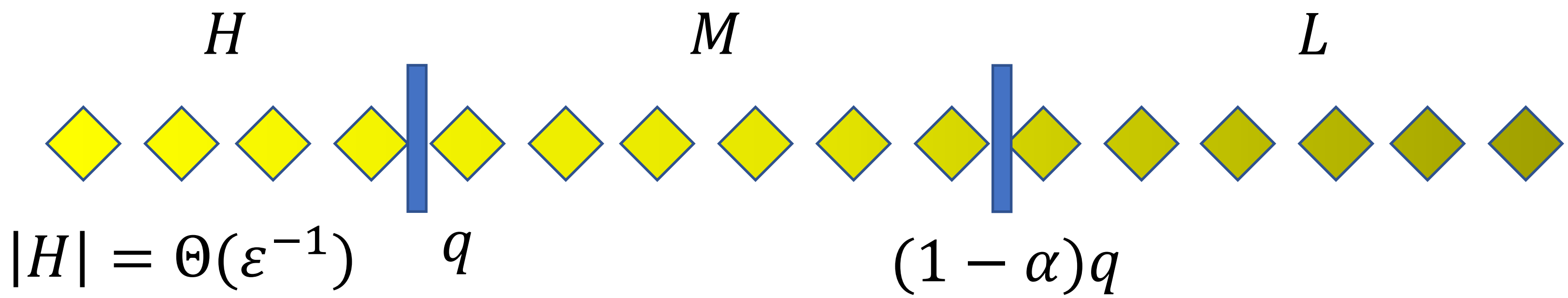
H: top $\Theta(\varepsilon^{-1})$ items

M: items in between

L: $(1-\alpha)$ multiplicative gap

$p_i \in [1,2]$

Improved Algorithm



Task: $1 + O(\varepsilon)$ approximate $\min\{O(\varepsilon^{-1}),\ f_I\}$

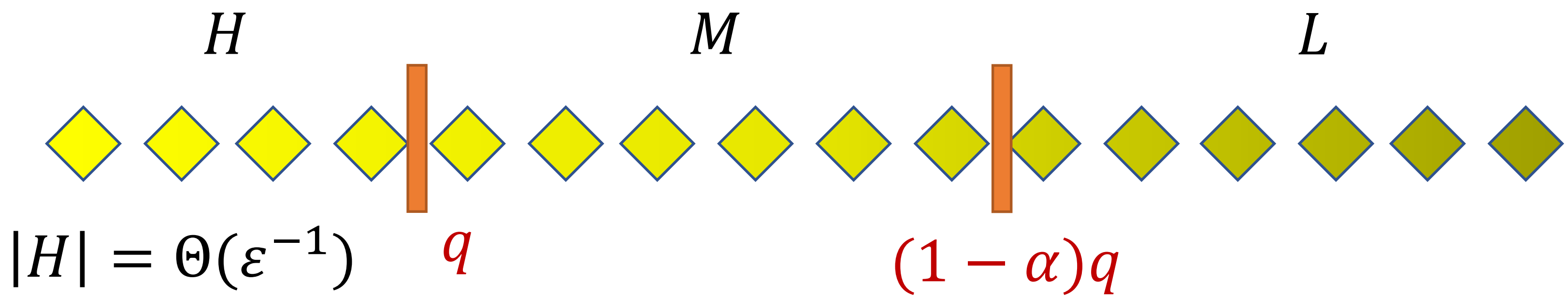
If: $\max_{\ell \in L} p_\ell / w_\ell \leq (1-\alpha) \cdot \min_{h \in H} p_h / w_h$

$\sum_{h \in H} w_h > W$

Then: In optimal solution of $f_{H \cup L}(W)$, L-items contribute total profit $\leq 2/\alpha$

$p_i \in [1,2]$

Improved Algorithm



Task: $1 + O(\varepsilon)$ approximate $\min\{O(\varepsilon^{-1}),\ f_I\}$

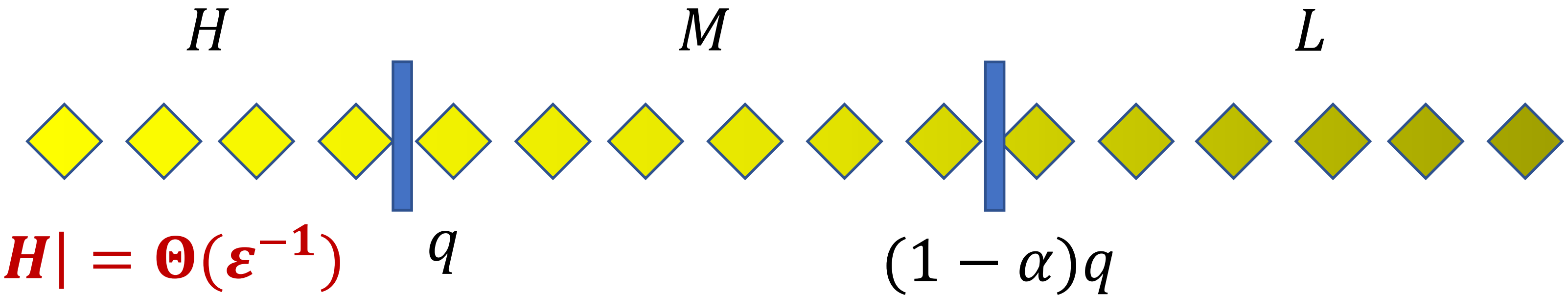
If: $\max_{\ell \in L} p_\ell / w_\ell \leq (1-\alpha) \cdot \min_{h \in H} p_h / w_h$ ✓

$\sum_{h \in H} w_h > W$

Then: In optimal solution of $f_{H \cup L}(W)$, L-items contribute total profit $\leq 2/\alpha$

$p_i \in [1,2]$

Improved Algorithm



$|H| = \Theta(\varepsilon^{-1})$

Task: $1 + O(\varepsilon)$ approximate $\min\{O(\varepsilon^{-1}),\ f_I\}$

(If $f_I(W) < O(\varepsilon^{-1})$ then $W < w(H)$)

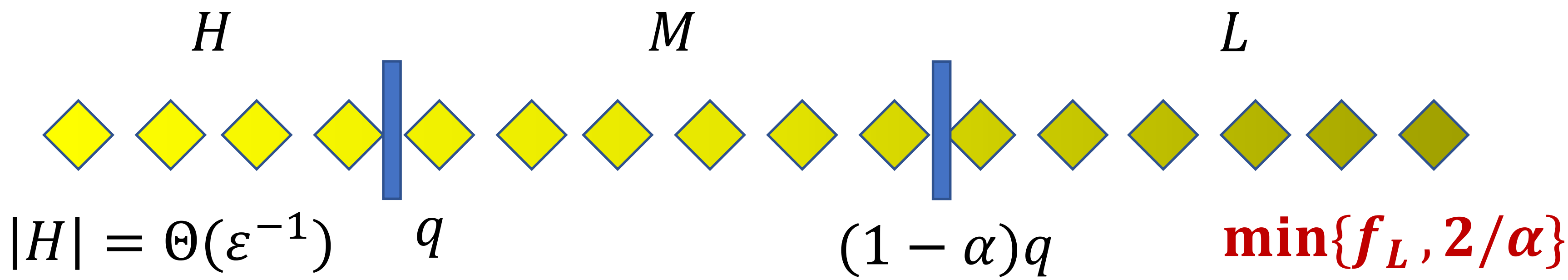
If: $\max_{\ell \in L} p_\ell / w_\ell \le (1-\alpha) \cdot \min_{h \in H} p_h / w_h$ ✓

$\sum_{h \in H} w_h > W$ ✓

Then: In optimal solution of $f_{H \cup L}(W)$, L-items contribute total profit $\le 2/\alpha$

$p_i \in [1, 2]$

Improved Algorithm



Task: $1 + O(\varepsilon)$ approximate $\min\{O(\varepsilon^{-1}),\ f_I\}$

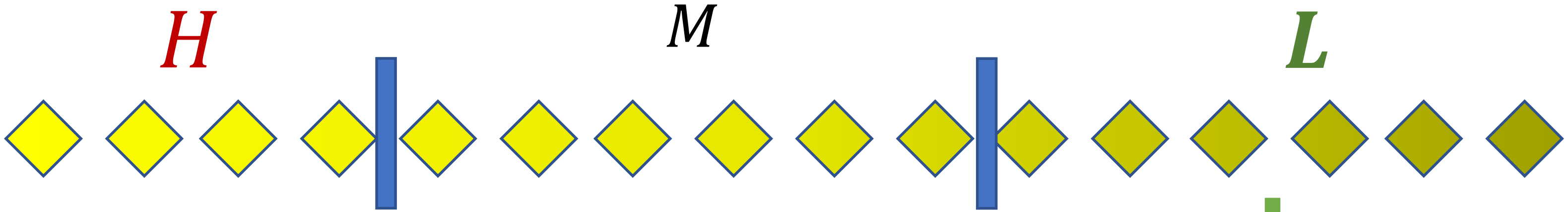
If: $\max_{\ell \in L} p_\ell / w_\ell \leq (1-\alpha) \cdot \min_{h \in H} p_h / w_h$

$\sum_{h \in H} w_h > W$

Then: In optimal solution of $f_{H \cup L}(W)$, L-items contribute total profit $\leq 2/\alpha$

$p_i \in [1,2]$

Improved Algorithm



$|H| = \Theta(\varepsilon^{-1})$ $\quad q \quad$ $(1-\alpha)q$ $\quad \min\{f_L, 2/\alpha\}$

$\tilde{O}(\varepsilon^{-4/3}n + \varepsilon^{-2})$ algo (Chan)

(faster when ***n* small**)

$\tilde{O}(\varepsilon^{-1}\sqrt{B}m)$ algo (Chan)

(faster when ***B* small**)

$n \coloneqq |H| \approx 1/\varepsilon$

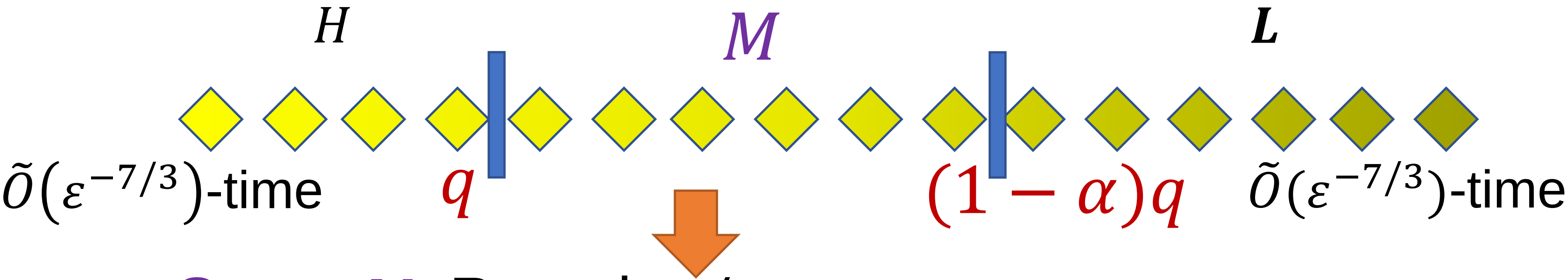
$m \coloneqq O(\varepsilon^{-1})$

$B \coloneqq 2/\alpha$

$\tilde{O}(\varepsilon^{-7/3})$ $\qquad$ $\tilde{O}(\varepsilon^{-7/3})$

Let $\alpha = \varepsilon^{2/3}$

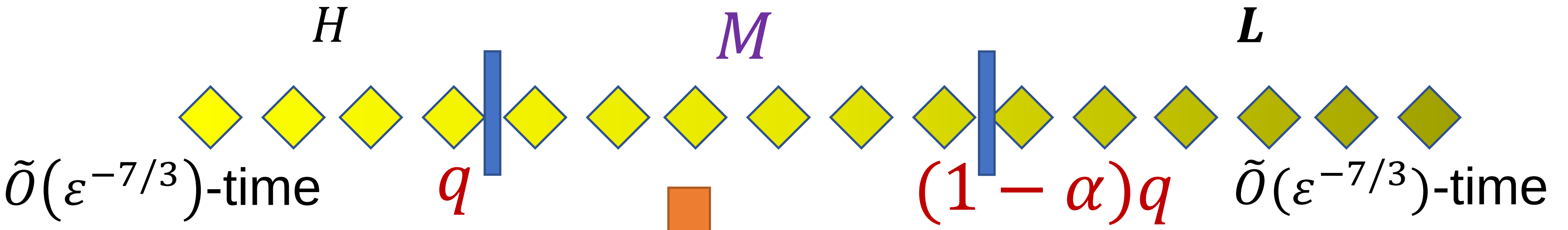
Improved Algorithm



Group M: Round p_i/w_i
down to powers of $(1+\varepsilon)$

Let $\alpha = \varepsilon^{2/3}$

Improved Algorithm

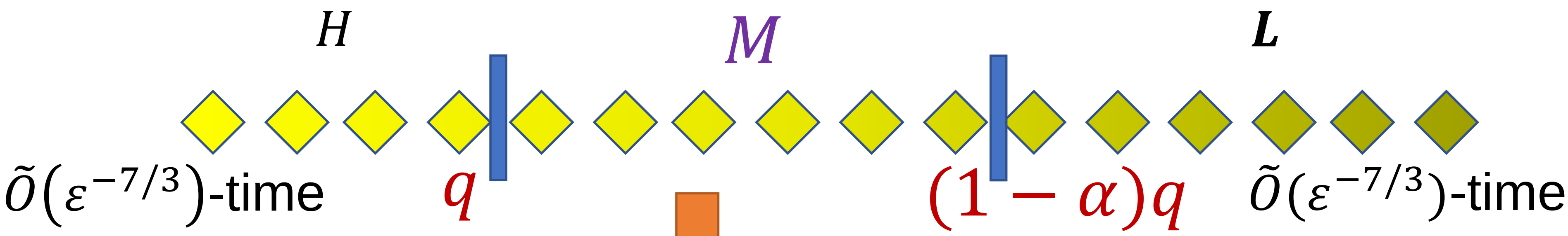


Group M: Round p_i/w_i
down to powers of $(1+\varepsilon)$

Only $\log_{1+\varepsilon}\frac{1}{1-\alpha} \approx \alpha/\varepsilon = \boldsymbol{\varepsilon^{-1/3}}$
distinct values of p_i/w_i

Let $\alpha = \varepsilon^{2/3}$

Improved Algorithm



Group M: Round p_i/w_i down to powers of $(1+\varepsilon)$

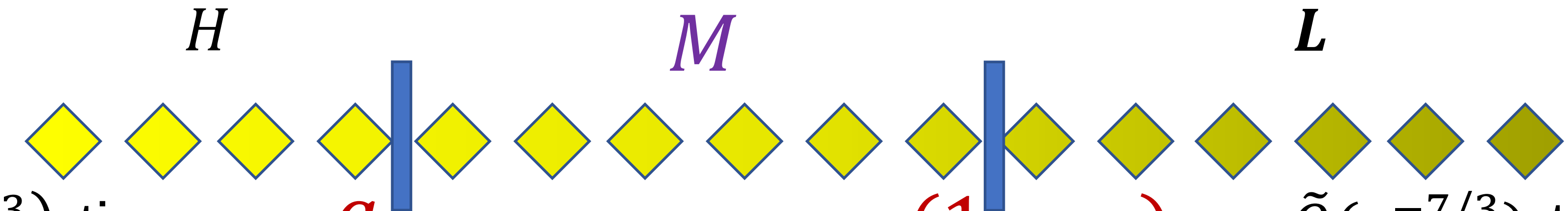
Only $\log_{1+\varepsilon}\frac{1}{1-\alpha} \approx \alpha/\varepsilon = \boldsymbol{\varepsilon^{-1/3}}$ distinct values of p_i/w_i

Items with the same p_i/w_i (profit ∝ weight): a ***Subset Sum*** instance

$\tilde{O}(n+\varepsilon^{-2})$ time [KMPS03]

Let $\boldsymbol{\alpha = \varepsilon^{2/3}}$

Improved Algorithm



$\tilde{O}(\varepsilon^{-7/3})$-time q $(1-\alpha)q$ $\tilde{O}(\varepsilon^{-7/3})$-time

Group M: Round p_i/w_i down to powers of $(1+\varepsilon)$

Only $\log_{1+\varepsilon}\frac{1}{1-\alpha} \approx \alpha/\varepsilon = \boldsymbol{\varepsilon^{-1/3}}$ distinct values of p_i/w_i

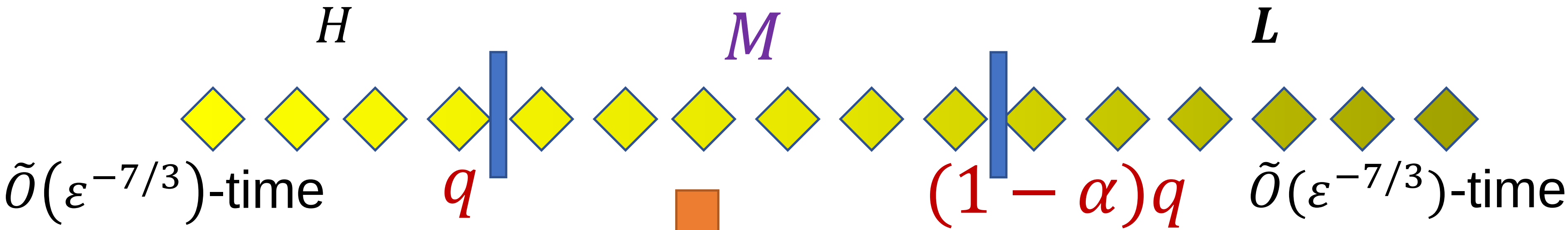
Items with the same p_i/w_i (profit ∝ weight): a ***Subset Sum*** instance

$\tilde{O}(n+\varepsilon^{-2})$ time [KMPS03]

Merge $\boldsymbol{\varepsilon^{-1/3}}$ groups

Let $\boldsymbol{\alpha = \varepsilon^{2/3}}$

Improved Algorithm



Group M: Round p_i/w_i down to powers of $(1+\varepsilon)$

$n + \boldsymbol{\varepsilon^{-1/3}} \cdot \varepsilon^{-2}$

Only $\log_{1+\varepsilon} \frac{1}{1-\alpha} \approx \alpha/\varepsilon = \boldsymbol{\varepsilon^{-1/3}}$ distinct values of p_i/w_i

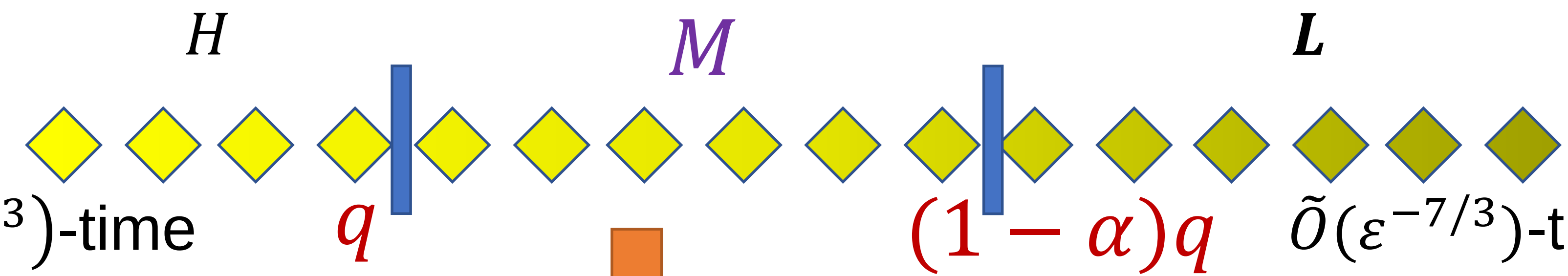
Items with the same p_i/w_i (profit ∝ weight): a ***Subset Sum*** instance

Merge $\boldsymbol{\varepsilon^{-1/3}}$ groups

$\tilde{O}(n+\varepsilon^{-2})$ time [KMPS03]

Let $\alpha = \varepsilon^{2/3}$

Improved Algorithm



Group M: Round p_i/w_i down to powers of $(1+\varepsilon)$

$n + \varepsilon^{-1/3} \cdot \varepsilon^{-2}$

Only $\log_{1+\varepsilon} \frac{1}{1-\alpha} \approx \alpha/\varepsilon = \varepsilon^{-1/3}$ distinct values of p_i/w_i

Total time:
$\tilde{O}(n + \varepsilon^{-7/3})$

Items with the same p_i/w_i (profit ∝ weight): a ***Subset Sum*** instance

Merge $\varepsilon^{-1/3}$ groups

$\tilde{O}(n + \varepsilon^{-2})$ time [KMPS03]

Let $\alpha = \varepsilon^{2/3}$

Further improvement

$\tilde{O}\left(\varepsilon^{-1}\sqrt{B}m\right)$ (Chan)

(faster when ***B* small**)

$\tilde{O}(\varepsilon^{-4/3}n + \varepsilon^{-2})$ (Chan)

(faster when ***n* small**)

extending Chan's techniques from **two levels to multiple levels**

$\tilde{O}\left(\varepsilon^{-4/3}B^{1/3}m^{2/3}\right)$ $(m^2 \gg \varepsilon^{-2}/B)$

(faster when ***B* small**)

$\tilde{O}(\varepsilon^{-3/2}n^{3/4} + n + \varepsilon^{-2})$

(faster when ***n* small**)

Greedy argument (this talk)

$\tilde{O}(n + \varepsilon^{-9/4})$

Open problem

Subset Sum:

$\tilde{O}\left(\min\left\{\boldsymbol{n} + \boldsymbol{\varepsilon}^{-2}, n\varepsilon^{-1}\right\}\right)$

[Kellerer, Mansini, Pferschy, and Speranza, 2003]

Unbounded Knapsack (each item has infinitely many copies) which easily reduces to 0-1 Knapsack:

$\tilde{O}(n + \varepsilon^{-2})$ [Jansen and Kraft, 2015]

Improve 0-1 Knapsack to $\tilde{O}(n + \varepsilon^{-2})$ time?

Thank you!