Part 12
Knapsack

Source: *Approximation Algorithms* (Vazirani, Springer Press)

Knapsack

Problem: Knapsack

- Given: n objects with weight $w_i \in \mathbb{Q}_+$ and profit $p_i \in \mathbb{Q}_+$, size $G \in \mathbb{Q}_+$
- Find: Subset of objects, maximizing the profit and not exceeding the weight bound:

$$OPT = \max_{I \subseteq \{1,\dots,n\}} \left\{ \sum_{i \in I} p_i \mid \sum_{i \in I} w_i \leq G \right\}$$

A dynamic program for Knapsack

Dynamic program:

(1) Assume restricted profits $p_i \in \{0, \dots, B\}$

(2) Compute table entries

$$\begin{aligned} T(i,b) &= \min_{I \subseteq \{1,\dots,i\}} \Big\{ \sum_{j \in I} w_j \mid \sum_{j \in I} p_j \geq b \Big\} \\ &= \text{minimum weight needed for a subset of the first } i \\ &\quad\ \text{objects to obtain a profit of at least } b \end{aligned}$$

using dynamic programming

$$T(i,b) = \min \Big\{ \underbrace{T(i-1,b)}_{\text{don't take } i}, \underbrace{T(i-1,b-p_i) + w_i}_{\text{take } i} \Big\} \ \forall i \ \forall p = 0, \dots, B$$

(3) Reconstruct I leading to $\max\{b \in \mathbb{N}_0 \mid T(n,b) \leq G\}$

Observation

The algorithm finds optimum solutions in time $O(n \cdot B)$.

The FPTAS

Algorithm:

(1) Scale profits s.t. $p_{\max} = n/\varepsilon$

(2) Round $p_i' := \lfloor p_i \rfloor$

(3) Compute and return optimum solution I for weights p_i'

Analysis of FPTAS

Theorem

Let $0 < \varepsilon \leq \frac{1}{2}$. The algo gives a $(1+2\varepsilon)$-apx in time $O(n^2/\varepsilon)$.

- W.l.o.g. $OPT \geq p_{\max} = n/\varepsilon$ (we can delete objects that even alone do not fit into the knapsack)
- Let I^* be optimum solution for original profits. Let OPT' be optimum value for profits p'. Then

$$\begin{aligned} OPT' \geq \sum_{i \in I^*} p'_i = \sum_{i \in I^*} \lfloor p_i \rfloor & \geq \sum_{i \in I^*} p_i - |I^*| \geq OPT - n \\ & \geq (1-\varepsilon) OPT \geq \frac{OPT}{1+2\varepsilon} \end{aligned}$$

- Let I be solution found by dynamic program:

$$\sum_{i \in I} p_i \geq \sum_{i \in I} p'_i = OPT' \geq \frac{OPT}{1+2\varepsilon}$$

- $B = \max\{p'_i\} \leq n/\varepsilon$ hence the running time is $O(n^2/\varepsilon)$

Part 13
Multi Constraint Knapsack

Source: Folklore

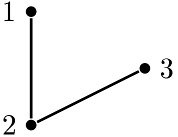
Multi Constraint Knapsack

Problem: Multi Constraint Knapsack (Mck)

- Given: n objects with profits $p_i \in \mathbb{Q}_+$ and k many budgets B_j. Object i has requirement $a_i^j \in \mathbb{Q}_+$ w.r.t. budget j.
- Find: Subset of objects, maximizing the profit and not exceeding any budget:

$$OPT = \max_{I \subseteq \{1,\dots,n\}} \left\{ \sum_{i \in I} p_i \mid \sum_{i \in I} a_i^j \leq B_j \ \forall j = 1, \dots, k \right\}$$

- For arbitrary k there is no $n^{1-\varepsilon}$-apx: Take an Independent Set instance $G = (V, E)$. For each edge $e = (u, v)$ add an "edge budget constraint" $a_u^e = a_v^e = 1, B_e = 1$. Then $OPT = OPT_{\text{IS}}$.



$\Rightarrow$

$$\begin{array}{lllll} \max & x_1 & +x_2 & +x_3 & \\ & 1x_1 & +1x_2 & +0x_3 & \leq 1 \\ & 0x_1 & +1x_2 & +1x_3 & \leq 1 \\ & & & x_i & \in \{0,1\} \end{array}$$

A PTAS for $k = O(1)$

Algorithm:

(1) Guess the $\lceil \frac{k}{\varepsilon} \rceil$ items I_{large} in the optimum solution with maximum profit

(2) Let x^* be optimum basic solution to the following LP

$$\begin{aligned} \max \sum_{i=1}^{n} x_i p_i \\ \sum_{i=1}^{n} a_i^j x_i &\leq B_j \quad \forall j = 1, \ldots, k \\ x_i &= 1 \quad \forall i \in I_{\text{large}} \\ x_i &= 0 \quad \forall i \notin I_{\text{large}} : p_i > \min\{p_j \mid j \in I_{\text{large}}\} \\ 0 \leq x_i &\leq 1 \quad \forall i = 1, \ldots, n \end{aligned}$$

(3) Output $I := \{i \mid x_i^* = 1\}$.

The Analysis

Theorem

For constant k the algorithm has polynomial running time. Furthermore $APX \geq (1-\varepsilon)OPT$.

- The produced solution is clearly feasible
- $LP \geq OPT$ (since we guess elements from OPT)
- Observation: $|\{i \mid 0 < x_i^* < 1\}| \leq k$ since x^* is a basic solution and appart from $0 \leq \ldots \leq 1$ there are only k constraints.
- For i with $0 < x_i^* < 1$ one has $p_i \leq \frac{\varepsilon}{k} OPT$

$$
\begin{aligned}
APX &\geq \sum_{i=1}^{n} \lfloor x_i^* \rfloor p_i \geq LP - \underbrace{\sum_{i:0<x_i^*<1} p_i}_{\leq k \cdot \frac{\varepsilon}{k} OPT} \\
&\geq OPT - k \cdot \frac{\varepsilon}{k} OPT = (1-\varepsilon) OPT
\end{aligned}
$$

□

Hardness of MultiConstraintKnapsack

Theorem

There is no FPTAS for MultiConstraintKnapsack *even for 2 budgets, unless* **NP** = **P**.

Problem: Partition

- Given: Numbers $a_1, \ldots, a_n \in \mathbb{N}$, $S := \sum_{i=1}^{n} a_i$, $m \in \{1, \ldots, n\}$
- Find: $I \subseteq \{1, \ldots, n\} : |I| = m, \sum_{i \in I} a_i = S/2$

- Recall: Partition is **NP**-hard.
- Define Mck instance with 2 constraints:

$$\begin{array}{rcll} \max \sum_{i=1}^{n} x_i & & & \\ \sum_{i=1}^{n} x_i a_i & \leq & S/2 & \\ \sum_{i=1}^{n} x_i (S - a_i) & \leq & S(m - \frac{1}{2}) & \\ x_i & \in & \{0, 1\} & \forall i = 1, \ldots, n \end{array}$$

Proof

- Claim: $\exists$ PARTITION solution $\Leftrightarrow OPT_{\text{MCK}} \geq m$
- $\Rightarrow$ Suppose $\exists I : |I| = m, \sum_{i \in I} a_i = S/2$. Then this is a MCK solution of value m since

$$\sum_{i \in I} (S - a_i) = mS - \sum_{i \in I} a_i = S(m - \frac{1}{2})$$

- $\Leftarrow$ Let I be MCK solution of value $\geq m$.

$$|I| \cdot S - \frac{S}{2} \overset{\text{1. constr.}}{\leq} |I| \cdot S - \underbrace{\sum_{i \in I} a_i}_{\leq S/2} = \sum_{i \in I} (S - a_i) \overset{\text{2. const.}}{\leq} m \cdot S - \frac{S}{2}$$

- Hence $|I| = m$. Then ineq. holds with "="
- Thus $\sum_{i \in I} a_i = S/2$. $\square$
- Now suppose for contradiction we would have an FPTAS for MCK: Then choose $\varepsilon := \frac{1}{n+1}$. Then the FPTAS would give an optimum solution for the instance resulting from the PARTITION reduction.

Part 14
Bin Packing

Source: *Combinatorial Optimization: Theory and Algorithms* (Korte, Vygen)

Bin Packing

Problem: BinPacking

- Given: Items with sizes $a_1, \ldots, a_n \in [0,1]$
- Find: Assign items to minimum number of bins of size 1.

$$OPT = \min\left\{k \mid \exists I_1 \dot{\cup} \ldots \dot{\cup} I_k = \{1,\ldots,n\} : \forall j : \sum_{i \in I_j} a_i \leq 1\right\}$$

- Define $\text{size}(I) = \sum_{i \in I} a_i$

First Fit

First Fit algorithm:

(1) Start with empty bins

(2) FOR $i = 1, \dots, n$ DO

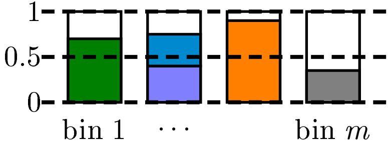
  (3) Assign item i to the bin B with least index such that $a_i + \sum_{j \in B} a_j \leq 1$

Lemma

Let m be the number of used bins. Then $m \leq 2\sum_{i=1}^n a_i + 1 \leq 2 \cdot OPT + 1$.

- All but $m-1$ bins must be filled with $\geq \frac{1}{2}$ (otherwise we would not have opened a new bin):

$$\sum_{i=1}^n a_i \geq \frac{1}{2}(m-1)$$



- Hence $m \leq 2\sum_{i=1}^n a_i + 1$.

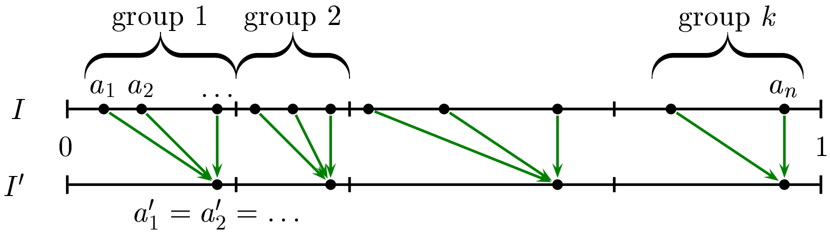
Linear Grouping

- INPUT: Instance $I = (a_1, \ldots, a_n)$, $k \in \mathbb{N}$
- OUTPUT: Instance $I' = (a'_1, \ldots, a'_n)$ with $a'_i \geq a_i$ and $\leq k$ different item sizes

(1) Sort $a_1 \leq a_2 \leq \ldots \leq a_n$

(2) Partition items into k consecutive groups of $\lceil n/k \rceil$ items (the last group might have less items)

(3) Let a'_i be the size of the largest item in i's group

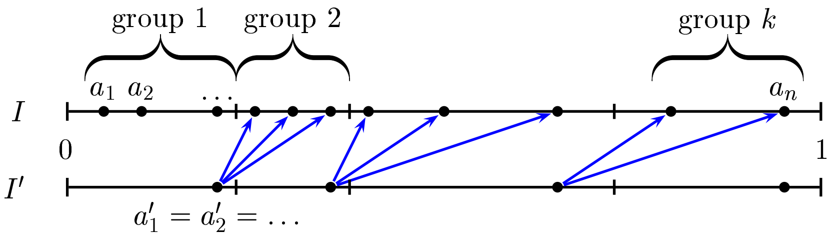


Linear Grouping (2)

Lemma

$OPT(I') \leq OPT(I) + \lceil n/k \rceil$.

- Consider solution $OPT(I)$. Assign item a'_i of group j to a space for item in group $j+1$
- Assign largest $\lceil n/k \rceil$ items to their own bin



An asymptotic PTAS

Algorithm of Fernandez de la Vega & Lueker:

(1) Let $I = \{i \mid a_i > \varepsilon\}$ be set of large items (other items are small)
(2) Apply linear grouping with $k = 1/\varepsilon^2$ groups to $I \to I'$
(3) Compute an optimum distribution of I'
(4) Distribute the small items over the used bins using First Fit

Lemma

The algorithm runs in polynomial time and uses at most $(1 + 2\varepsilon)OPT + 1$ bins.

- Let $b_1, \ldots, b_{1/\varepsilon^2}$ different item sizes in I'.
- Possible bin configurations $\mathcal{P} = \{p \in \{0, \ldots, 1/\varepsilon\}^{1/\varepsilon^2} \mid b^T p \leq 1\}$. $|\mathcal{P}| \leq (1/\varepsilon^2)^{1/\varepsilon}$.
- Solution is described by $(n_p)_{p \in \mathcal{P}}$ (n_p = how many times shall I pack a bin with configuration p?), $n_p \in \{0, \ldots, n\}$
- $\leq n^{(1/\varepsilon^2)^{1/\varepsilon}}$ possibilities for $(n_p)_{p \in \mathcal{P}}$.

An asymptotic PTAS (2)

- We need $OPT(I') + \#$ of bins additionally opened for the small items
- Note that

$$OPT(I') \leq OPT(I) + \lceil |I| \cdot \varepsilon^2 \rceil \leq OPT(I) + \lceil \varepsilon \cdot OPT(I) \rceil = (1+2\varepsilon) \cdot OPT$$

using $OPT(I) \geq \sum_{i \in I} a_i \geq \varepsilon \cdot |I|$ and $OPT \geq OPT(I)$.

- Suppose we need to open an additional bin for small items. Let m be total number of used bins. Then all but one bin are filled to $\geq 1 - \varepsilon$. Hence

$$OPT \geq \sum_{i=1}^{m} a_i \geq (1-\varepsilon) \cdot (m-1)$$

and

$$m \leq \frac{OPT}{1-\varepsilon} + 1 \leq (1+2\varepsilon)OPT + 1$$

Section 14.1
The algorithm of Karmarkar & Karp

The Algorithm of Karmarkar & Karp

Theorem (Karmarkar, Karp '82)

One can compute a BinPacking *solution with* $OPT + O(\log^2 n)$ *many bins in polynomial time.*

- Assume $a_i \geq \delta := \frac{1}{n}$ (again one can distribute items that are smaller than $\frac{1}{n}$ after distributing the large items.

The Gilmore-Gomory LP-relaxation

- Let $b_i \in \mathbb{N}$ now the number of items of size a_i
- n = number of different item sizes
- $m := \sum_{i=1}^{n} b_i$ = total number of items
- $\mathcal{P} = \{p \in \mathbb{Z}_+^n \mid a^T p \leq 1\}$ set of feasible patterns
- Variable x_p = # of bins packed with pattern p

Primal

$$\begin{aligned} \min \mathbf{1}^T x & & & (P(\mathcal{P})) \\ \sum_{p \in \mathcal{P}} x_p p & \geq & b & \\ x & \geq & \mathbf{0} & \end{aligned}$$

- # var. exponential
- # constr. polynomial

Dual

$$\begin{aligned} \max y^T b & & & (D(\mathcal{P})) \\ p^T y & \leq & 1 & \quad \forall p \in \mathcal{P} \\ y & \geq & \mathbf{0} & \end{aligned}$$

- # var. polynomial
- # constr. exponential

Idea: Solve the dual with Ellipsoid!

Example

- Item sizes $a_1 = 0.3, a_2 = 0.4$
- # of items $b_1 = 31, b_2 = 7$
- Set of patterns $\mathcal{P} = \left\{ \binom{0}{1}, \binom{0}{2}, \binom{1}{1}, \binom{2}{1}, \binom{1}{0}, \binom{2}{0}, \binom{3}{0} \right\}$

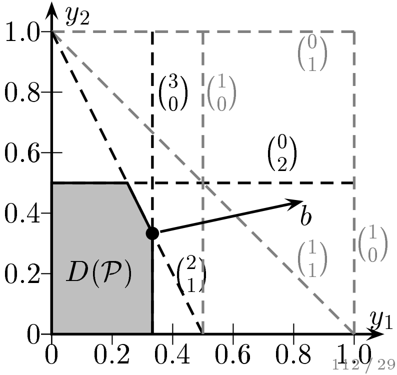
Primal

$$\min \mathbf{1}^T x$$
$$\begin{pmatrix} 0 & 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 2 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} x \geq \begin{pmatrix} 31 \\ 7 \end{pmatrix}$$
$$x \geq \mathbf{0}$$

- Opt basic solution is $x = (0, 0, 0, 7, 0, 0, \frac{17}{3})$

Dual

$$\max 31 y_1 + 7 y_2$$
$$\begin{pmatrix} 0 & 1 \\ 0 & 2 \\ 1 & 1 \\ 2 & 1 \\ 1 & 0 \\ 2 & 0 \\ 3 & 0 \end{pmatrix} y \leq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$
$$y \geq \mathbf{0}$$



Weak Separation Problem

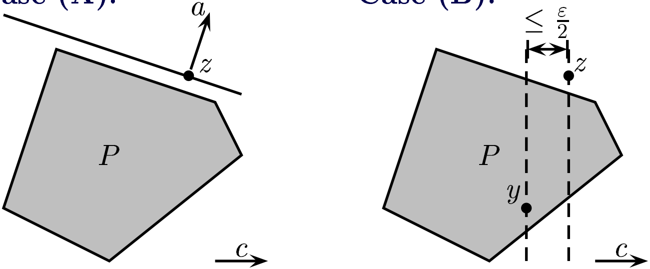
ε-Weak Separation Oracle for $P \subseteq \mathbb{R}^n$, obj.fct. $c \in \mathbb{Q}^n$

INPUT: Vector $z \in \mathbb{Q}^n$

OUTPUT: One of the following

- *Case (A):* Vector a with $a^T x \leq a^T z \; \forall x \in P$
- *Case (B):* Point $y \in P$ with $c^T y \geq c^T z - \frac{\varepsilon}{2}$

Case (A): **Case (B):**



- If $z \in P$, just return z ($\rightarrow$ case (B)).

Grötschel-Lovász-Schrijver Algorithm

- INPUT: $c \in \mathbb{Q}^n, x_0 \in \mathbb{Q}^n, \varepsilon, r, R \in \mathbb{Q}_+$: $B(x_0, r) \subseteq P \subseteq B(x_0, R)$
- OUTPUT: $y^* \in P$ with $c^T y^* \geq OPT_f - \varepsilon$

(1) Ellipsod $E_0 := B(x_0, R)$ with center $z_0 := x_0$, $y^* := x_0$

(2) FOR $t = 0, \ldots, poly$ DO

  (4) Submit z_t to ε-weak separation oracle

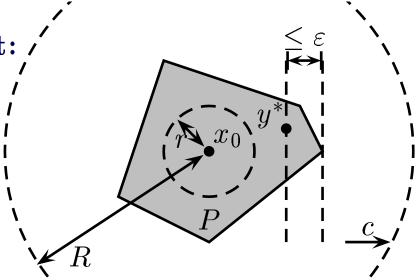
  (5) *Case (A) → a:* Compute $E_{t+1} \supseteq E_t \cap \{x \mid a^T x \leq a^T z_t\}$

  (6) *Case (B) → $y \in P$:*

    (7) IF $c^T y > c^T y^*$ THEN $y^* := y$

    (8) Compute $E_{t+1} \supseteq E_t \cap \{x \mid c^T x \geq c^T z_t\}$

Input/ Output:



Grötschel-Lovász-Schrijver Algorithm

- INPUT: $c \in \mathbb{Q}^n, x_0 \in \mathbb{Q}^n, \varepsilon, r, R \in \mathbb{Q}_+$: $B(x_0, r) \subseteq P \subseteq B(x_0, R)$
- OUTPUT: $y^* \in P$ with $c^T y^* \geq OPT_f - \varepsilon$

(1) Ellipsod $E_0 := B(x_0, R)$ with center $z_0 := x_0$, $y^* := x_0$

(2) FOR $t = 0, \ldots, poly$ DO

  (4) Submit z_t to ε-weak separation oracle

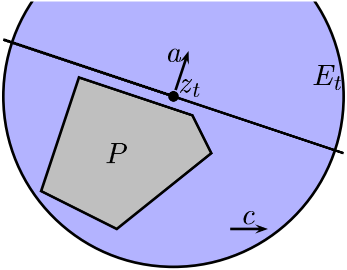
  (5) *Case (A)* $\rightarrow a$: Compute $E_{t+1} \supseteq E_t \cap \{x \mid a^T x \leq a^T z_t\}$

  (6) *Case (B)* $\rightarrow y \in P$:

    (7) IF $c^T y > c^T y^*$ THEN $y^* := y$

    (8) Compute $E_{t+1} \supseteq E_t \cap \{x \mid c^T x \geq c^T z_t\}$

Case (A):



Grötschel-Lovász-Schrijver Algorithm

- INPUT: $c \in \mathbb{Q}^n, x_0 \in \mathbb{Q}^n, \varepsilon, r, R \in \mathbb{Q}_+$: $B(x_0, r) \subseteq P \subseteq B(x_0, R)$
- OUTPUT: $y^* \in P$ with $c^T y^* \geq OPT_f - \varepsilon$

(1) Ellipsod $E_0 := B(x_0, R)$ with center $z_0 := x_0$, $y^* := x_0$

(2) FOR $t = 0, \ldots, poly$ DO

  (4) Submit z_t to ε-weak separation oracle

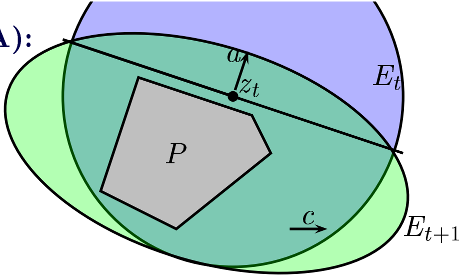
  (5) *Case (A)* $\rightarrow$ *a:* Compute $E_{t+1} \supseteq E_t \cap \{x \mid a^T x \leq a^T z_t\}$

  (6) *Case (B)* $\rightarrow y \in P$*:*

    (7) IF $c^T y > c^T y^*$ THEN $y^* := y$

    (8) Compute $E_{t+1} \supseteq E_t \cap \{x \mid c^T x \geq c^T z_t\}$

Case (A):

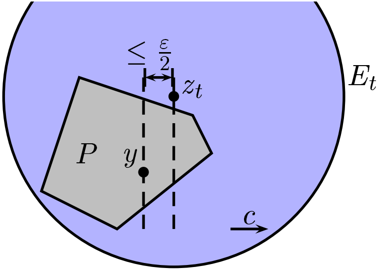


Grötschel-Lovász-Schrijver Algorithm

- INPUT: $c \in \mathbb{Q}^n, x_0 \in \mathbb{Q}^n, \varepsilon, r, R \in \mathbb{Q}_+$: $B(x_0, r) \subseteq P \subseteq B(x_0, R)$
- OUTPUT: $y^* \in P$ with $c^T y^* \geq OPT_f - \varepsilon$

(1) Ellipsod $E_0 := B(x_0, R)$ with center $z_0 := x_0$, $y^* := x_0$
(2) FOR $t = 0, \ldots, poly$ DO
  (4) Submit z_t to ε-weak separation oracle
  (5) *Case (A)* $\rightarrow a$: Compute $E_{t+1} \supseteq E_t \cap \{x \mid a^T x \leq a^T z_t\}$
  (6) *Case (B)* $\rightarrow y \in P$:
    (7) IF $c^T y > c^T y^*$ THEN $y^* := y$
    (8) Compute $E_{t+1} \supseteq E_t \cap \{x \mid c^T x \geq c^T z_t\}$

Case (B):



Grötschel-Lovász-Schrijver Algorithm

- INPUT: $c \in \mathbb{Q}^n, x_0 \in \mathbb{Q}^n, \varepsilon, r, R \in \mathbb{Q}_+$: $B(x_0, r) \subseteq P \subseteq B(x_0, R)$
- OUTPUT: $y^* \in P$ with $c^T y^* \geq OPT_f - \varepsilon$

(1) Ellipsod $E_0 := B(x_0, R)$ with center $z_0 := x_0$, $y^* := x_0$

(2) FOR $t = 0, \ldots, poly$ DO

  (4) Submit z_t to ε-weak separation oracle

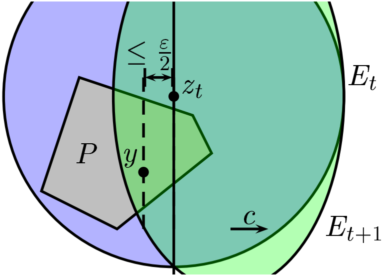
  (5) *Case (A)* $\rightarrow a$*:* Compute $E_{t+1} \supseteq E_t \cap \{x \mid a^T x \leq a^T z_t\}$

  (6) *Case (B)* $\rightarrow y \in P$*:*

    (7) IF $c^T y > c^T y^*$ THEN $y^* := y$

    (8) Compute $E_{t+1} \supseteq E_t \cap \{x \mid c^T x \geq c^T z_t\}$

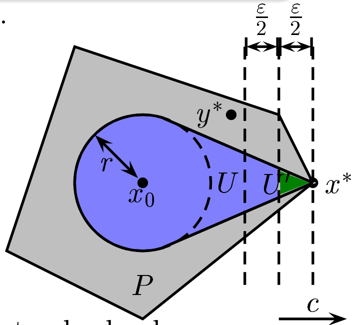
Case (B):



Analysis

> **Theorem**
>
> *Let $OPT_f = \max\{c^T x \mid x \in P\}$. The GLS algorithm finds a $y^* \in P$ with $c^T y^* \geq OPT_f - \varepsilon$.*

- Suppose for contradiction this is false.
- Let $x^* \in P$ be opt. sol.; φ input size.
- Inequalities from case (A) never cut points from P
- Ineq. from case (B) never cut points better than $OPT_f - \frac{\varepsilon}{2}$ (otherwise we would have found a suitable y^*)
- Let $U := \text{conv}\{B(x_0, r), x^*\}$ and $U' = \{x \in U \mid c^T x \geq OPT_f - \frac{\varepsilon}{2}\}$. By standard volume bounds: $\text{vol}(U') \geq (\frac{1}{2})^{poly(\varphi)}$. But $U' \subseteq E_t \; \forall t$. After $poly(\varphi)$ many it. $\text{vol}(E_t) = (1 - \frac{\Theta(1)}{n})^t \cdot \text{vol}(E_0) < \text{vol}(U')$. Contradiction!

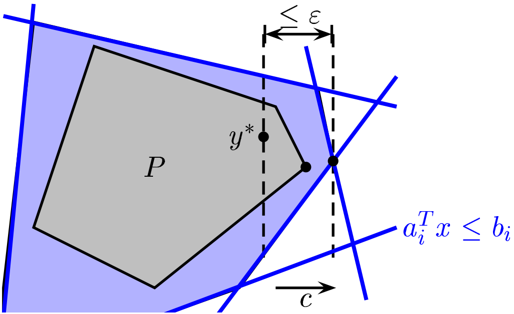


A useful observation

Observation

Consider a run of the GLS algorithm for $P \subseteq \mathbb{R}^n$ which yields $y^* \in P$. Let $a_1^T x \leq b_1, \ldots, a_N^T x \leq b_N$ be the inequalities which the oracle are returned for Case (A).

- Each $a_i^T x \leq b_i$ is feasible for P
- $c^T y^* \geq \max\{c^T x \mid a_i^T x \leq b_i \; \forall i = 1, \ldots, N\} - \varepsilon$



Solving $D(\mathcal{P})$

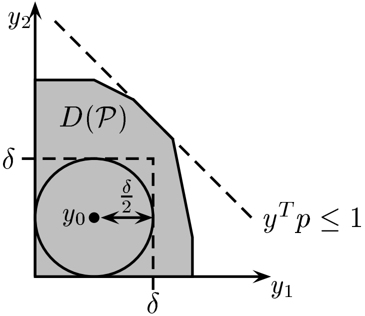
Lemma

Suppose $a_i \geq \delta$. Then we can find a feasible solution y^* to $D(\mathcal{P})$ of value $\geq OPT_f - 1$ in time polynomial in $n, m, \frac{1}{\delta}$.

- Apply GLS algo for $\varepsilon := 1$. Choose $y_0 = (\frac{\delta}{2}, \dots, \frac{\delta}{2})$.

$$B\left(y_0, \frac{\delta}{2}\right) \overset{(\delta,\dots,\delta)^T p \leq 1}{\subseteq} D(\mathcal{P}) \subseteq B(y_0, n)$$

- We use $\sum_{i=1}^{n} p_i \leq \frac{1}{\delta}$ for any feasible pattern $p \in \mathcal{P}$ since $a_i \geq \delta$

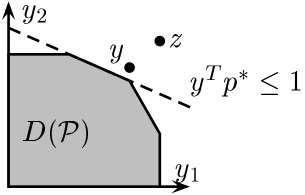


Solving $D(\mathcal{P})$ (2)

- We solve ε-weak separation problem for $z \in \mathbb{Q}^n$.
- If $z_i < 0 \rightarrow$ Case (A) (inequality $z_i \geq 0$ violated)
- If $z_i > 1 \rightarrow$ Case (A) (inequality $z^T e_i \leq 1$ violated)
- Round z down to nearest multiple of $\frac{1}{2m}$ and term this vector y. Solve $p^* = \text{argmax}\{y^T p \mid p \in \mathcal{P}\}$ (KNAPSACK with profits from $0, 1 \cdot \frac{1}{2m}, 2 \cdot \frac{1}{2m}, \ldots, 1$)

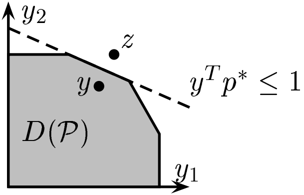
Case $y^T p^* > 1$:

- Then $z^T p^* \geq y^T p^* > 1$ $\rightarrow$ Case (A).



Case $y^T p^* \leq 1$:

- Then $y \in D(\mathcal{P})$. And $z^T b - y^T b \leq m \cdot \frac{1}{2m} = \frac{1}{2} = \frac{\varepsilon}{2}$. $\rightarrow$ Case (B)



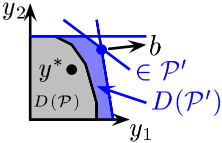
- GLS yields a solution y^* mit $b^T y^* \geq OPT_f - 1$. □

Finding a near optimal basic solution for $P(\mathcal{P})$

Theorem

Suppose $a_i \geq \delta$. Then we can find a basic solution x^* for $P(\mathcal{P})$ of value $\leq OPT_f + 1$ in time polynomial in $n, m, \frac{1}{\delta}$.

- Run GLS to obtain sol. y^* to $D(\mathcal{P})$ with $b^T y^* \geq OPT_f - 1$
- Let $y^T p \leq 1$, $p \in \mathcal{P}'$ be inequalities returned by oracle for case (A). $\mathcal{P}' \subseteq \mathcal{P}$ has polynomial size and



$$D(\mathcal{P}) \overset{y^* \text{ valid for } D(\mathcal{P})}{\geq} b^T y^* \geq D(\mathcal{P}') - 1 \qquad (1)$$

- Compute optimum basic solution x^* for $P(\mathcal{P}')$ in poly-time.

$$\mathbf{1}^T x^* = P(\mathcal{P}') \overset{\text{duality}}{=} D(\mathcal{P}') \overset{(1)}{\leq} D(\mathcal{P}) + 1 \overset{\text{duality}}{=} P(\mathcal{P}) + 1$$

- x^* is also a (non-optimal) basic solution for $P(\mathcal{P})$

Geometric Grouping

- INPUT: Instance $I = (a_1, \ldots, a_n)$, $size(I) = \sum_{i=1}^{n} a_i b_i \leq n$, $a_i \geq \delta$
- OUTPUT: Rounded up instance I' with $n/2$ diff. item sizes $OPT_f(I') \leq OPT_f(I)$ plus waste of $O(\log \frac{1}{\delta})$

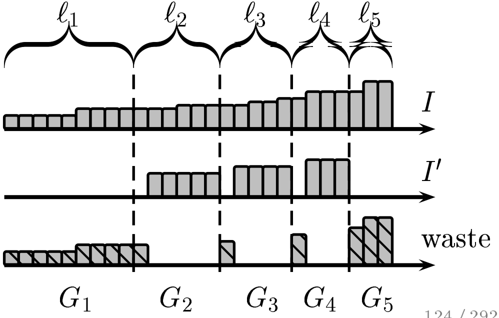
(1) Sort items w.r.t. sizes $e_1 \leq e_2 \leq \ldots \leq e_m$ (a_i appears b_i times)

(2) Let $G_1 = \{e_1, \ldots, e_{\ell_1}\}$ be minimal set of items with $\sum_{i \in G_1} e_i \geq 2$, then continue with $G_2, \ldots$. Let $\ell_i := |G_i|$ be number of items in G_i

(3) Remove first and last group → waste

(4) From G_i throw away smallest $\ell_i - \ell_{i+1}$ items → waste

(5) Round up items in G_i to largest item → I'



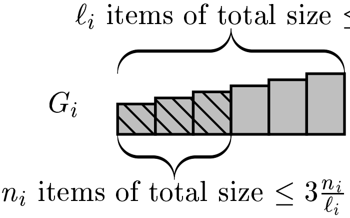
Geometric Grouping (2)

Lemma

Size of waste is $O(\log \frac{1}{\delta})$.

- Size of 1st and last group is $O(1)$
- Consider group G_i. Total size of items in G_i is ≤ 3.
- Num of groups is $\leq n/2$. Cleary $\frac{2}{\delta} \geq \ell_1 \geq \ell_2 \geq \ldots$.
- The $n_i := \ell_i - \ell_{i+1}$ smallest items in G_i have size $\leq 3\frac{n_i}{\ell_i}$.

$$\text{waste} \leq 3\sum_i \frac{n_i}{\ell_i} \leq 3\sum_{j=1}^{\ell_1} \frac{1}{j} \overset{\ell_1 \leq 2/\delta}{=} O(\log \frac{1}{\delta})$$



The algorithm

Algorithm:

(1) Compute a basic solution x to $P(\mathcal{P})$ with $\mathbf{1}^T x \leq OPT_f + 1$
(2) Buy $\lfloor x_p \rfloor$ times pattern p, let I be remaining instance
(3) Apply geometric grouping to I (with n different item sizes) $\rightarrow I'$ (with $n/2$ different item sizes)
(4) Recurse

Theorem

One has $APX \leq OPT_f + O(\log^2 n)$.

- Since x is basic solution, $|\{p \mid x_p > 0\}| \leq n$.
- After (2) $size(I) \leq \sum_p (x_p - \lfloor x_p \rfloor) \leq n$.
- Let x^t be solution x in iteration t. We buy $\sum_p \lfloor x_p^t \rfloor$ bins, but OPT_f decreases by the same quantity.
- We pay in total $OPT_f +$ total waste. We have $O(\log n)$ recursions; in each recursion we have a waste of $O(\log \frac{1}{\delta}) = O(\log n)$. □

State of the art

- Computing OPT exactly is **NP**-hard even if the numbers a_i are unary encoded (i.e. BinPacking is strongly **NP**-hard).

Open question

One can compute a Bin Packing solution with $\leq OPT + 1$ bins in poly-time?

Mixed Integer Roundup Conjecture

One has $OPT \leq \lceil OPT_f \rceil + 1$.