# Approximation Algorithms for Integer Programming with Resource Augmentation

Guochuan Zhang

Zhejiang University

Joint with Hauke Brinkop (Kiel U.),Hua Chen (Zhejiang U. of Technology), Lin Chen (Zhejiang U.), and Klaus Jansen (Kiel U.)

STACS 2026, Grenoble

# The General Model

**Integer (Linear) Programming (a standard form)**

$$\min\{\mathbf{w}\mathbf{x} : H\mathbf{x} = \mathbf{b}, 0 \le \mathbf{x} \le \mathbf{u}, \mathbf{x} \in \mathbb{Z}^N\},$$

where $\mathbf{w}, \mathbf{u} \in \mathbb{Z}^N$, $\mathbf{b} \in \mathbb{Z}^M$, and $H \in \mathbb{Z}^{M \times N}$.

# The General Model

**Integer (Linear) Programming (a standard form)**

$$\min\{\mathbf{w}\mathbf{x} : H\mathbf{x} = \mathbf{b}, 0 \le \mathbf{x} \le \mathbf{u}, \mathbf{x} \in \mathbb{Z}^N\},$$

where $\mathbf{w}, \mathbf{u} \in \mathbb{Z}^N$, $\mathbf{b} \in \mathbb{Z}^M$, and $H \in \mathbb{Z}^{M \times N}$.

- NP-hard [Karp, 1972]

# The General Model

**Integer (Linear) Programming (a standard form)**

$$\min\{\mathbf{wx} : H\mathbf{x} = \mathbf{b}, 0 \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{Z}^N\},$$

where $\mathbf{w}, \mathbf{u} \in \mathbb{Z}^N$, $\mathbf{b} \in \mathbb{Z}^M$, and $H \in \mathbb{Z}^{M \times N}$.

- NP-hard [Karp, 1972]
- $(M \cdot \Delta)^{O(M^2)} \cdot \mathrm{poly}(N)$ [Papadimitriou, 1981]
  ($\Delta$ is the largest absolute entry in $H$ )

# The General Model

**Integer (Linear) Programming (a standard form)**

$$\min\{\mathbf{w}\mathbf{x} : H\mathbf{x} = \mathbf{b}, 0 \le \mathbf{x} \le \mathbf{u}, \mathbf{x} \in \mathbb{Z}^N\},$$

where $\mathbf{w}, \mathbf{u} \in \mathbb{Z}^N$, $\mathbf{b} \in \mathbb{Z}^M$, and $H \in \mathbb{Z}^{M \times N}$.

- NP-hard [Karp, 1972]
- $(M \cdot \Delta)^{O(M^2)} \cdot \mathrm{poly}(N)$ [Papadimitriou, 1981]
  ($\Delta$ is the largest absolute entry in $H$ )
- $2^{O(N^3)} \cdot \mathrm{poly}(M, \log \Delta)$ [Lenstra, 1983]

# The General Model

**Integer (Linear) Programming (a standard form)**

$$\min\{\mathbf{wx} : H\mathbf{x} = \mathbf{b}, 0 \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{Z}^N\},$$

where $\mathbf{w}, \mathbf{u} \in \mathbb{Z}^N$, $\mathbf{b} \in \mathbb{Z}^M$, and $H \in \mathbb{Z}^{M \times N}$.

- NP-hard [Karp, 1972]
- $(M \cdot \Delta)^{O(M^2)} \cdot \mathrm{poly}(N)$ [Papadimitriou, 1981]
  ($\Delta$ is the largest absolute entry in $H$ )
- $2^{O(N^3)} \cdot \mathrm{poly}(M, \log \Delta)$ [Lenstra, 1983]

- $(\sqrt{M}\Delta)^{O(M)} \cdot \mathrm{poly}(N)$ if $\mathbf{u} = \infty$ [Jansen et al., 2019]
- $N^{O(N)} \cdot \mathrm{poly}(M, \log \Delta)$ [Kannan, 1987]
  $(\log N)^{O(N)} \cdot \mathrm{poly}(M, \log \Delta)$ [Reis and Rothvoss, 2023]

# Remark

Even for $M = 1$, the running time is either pseudo-polynomial (depends on $\Delta$), or exponential in the number of variables.

# Remark

Even for $M = 1$, the running time is either pseudo-polynomial (depends on $\Delta$), or exponential in the number of variables.

**Question 1**

Can we solve an IP approximately in polynomial time?

- Preserve the optimality while slightly violating the constraints by $O(\varepsilon\Delta)$.

# Block-Structured IP

**$n$-fold IP & 2-stage stochastic IP**

$$H_1 = \begin{pmatrix} D^1 & D^2 & \cdots & D^n \\ A^1 & 0 & & 0 \\ 0 & A^2 & & 0 \\ & & \ddots & \\ 0 & 0 & & A^n \end{pmatrix}, \quad H_2 = \begin{pmatrix} B^1 & A^1 & 0 & & 0 \\ B^2 & 0 & A^2 & & 0 \\ \vdots & & & \ddots & \\ B^n & 0 & 0 & & A^n \end{pmatrix}$$

# Block-Structured IP

## $n$-fold IP & 2-stage stochastic IP

$$H_1 = \begin{pmatrix} D^1 & D^2 & \cdots & D^n \\ A^1 & 0 & & 0 \\ 0 & A^2 & & 0 \\ & & \ddots & \\ 0 & 0 & & A^n \end{pmatrix}, \quad H_2 = \begin{pmatrix} B^1 & A^1 & 0 & & 0 \\ B^2 & 0 & A^2 & & 0 \\ \vdots & & & \ddots & \\ B^n & 0 & 0 & & A^n \end{pmatrix}$$

## Known Results

For $n$-fold IP and 2-stage stochastic IP, there exist FPT algorithms parameterized by $s_A, s_B, s_C, s_D$, $t_A, t_B, t_C, t_D, \Delta$.

FPT algorithm: running time $f(k) \cdot \text{poly}(|I|)$

# Running Time of FPT

**$n$-fold IP**

- $\Delta^{O(t_A(s_A s_D + s_A t_A))} \cdot n^3 |I|$ [Hemmecke et al., 2013]
- [Eisenbrand et al., 2018]
- [Eisenbrand et al., 2019] [Altmanová et al., 2019] [Jansen et al., 2019]
- [Jansen et al., 2020]
- $2^{O(s_A^2 s_D)}(s_D s_A \Delta)^{O(s_A^2 + s_A s_D^2)}(n t_A)^{1+o(1)}$ [Cslovjecsek et al., 2021]

# Running Time of FPT

**$n$-fold IP**

- $\Delta^{O(t_A(s_A s_D + s_A t_A))} \cdot n^3|I|$ [Hemmecke et al., 2013]
- [Eisenbrand et al., 2018]
- [Eisenbrand et al., 2019] [Altmanová et al., 2019] [Jansen et al., 2019]
- [Jansen et al., 2020]
- $2^{O(s_A^2 s_D)}(s_D s_A \Delta)^{O(s_A^2 + s_A s_D^2)}(nt_A)^{1+o(1)}$ [Cslovjecsek et al., 2021]

The running time is pseudo-polynomial (if $\Delta$ is not taken as a parameter).

# Running Time of FPT

**$n$-fold IP**

- $\Delta^{O(t_A(s_A s_D + s_A t_A))} \cdot n^3|I|$ [Hemmecke et al., 2013]
- [Eisenbrand et al., 2018]
- [Eisenbrand et al., 2019] [Altmanová et al., 2019] [Jansen et al., 2019]
- [Jansen et al., 2020]
- $2^{O(s_A^2 s_D)}(s_D s_A \Delta)^{O(s_A^2 + s_A s_D^2)}(n t_A)^{1+o(1)}$ [Cslovjecsek et al., 2021]

The running time is pseudo-polynomial (if $\Delta$ is not taken as a parameter).

It is NP-hard to determine whether an $n$-fold IP admits a feasible solution even if $A = (1, 1, \Delta)$ and $D = (1, 0, 0)$, where $\Delta \in \mathbb{Z}$ is part of the input.

[Chen, Chen and **Z**, 2022]

# Running Time of FPT

**$n$-fold IP**

- $\Delta^{O(t_A(s_A s_D + s_A t_A))} \cdot n^3 |I|$ [Hemmecke et al., 2013]
- [Eisenbrand et al., 2018]
- [Eisenbrand et al., 2019] [Altmanová et al., 2019] [Jansen et al., 2019]
- [Jansen et al., 2020]
- $2^{O(s_A^2 s_D)} (s_D s_A \Delta)^{O(s_A^2 + s_A s_D^2)} (n t_A)^{1+o(1)}$ [Cslovjecsek et al., 2021]

The running time is pseudo-polynomial (if $\Delta$ is not taken as a parameter).

It is NP-hard to determine whether an $n$-fold IP admits a feasible solution even if $A = (1, 1, \Delta)$ and $D = (1, 0, 0)$, where $\Delta \in \mathbb{Z}$ is part of the input.

[Chen, Chen and **Z**, 2022]

**Question 2**

Can we solve $n$-fold IP approximately in polynomial time?

- Preserve the optimality while slightly violating the constraints.

# Recall the Questions

**Question 1**

Can we solve an IP approximately in polynomial time?

- Preserve the optimality while slightly violating the constraints by $O(\varepsilon\Delta)$.

**Question 2**

Can we solve $n$-fold IP approximately in polynomial time?

- Preserve the optimality while slightly violating the constraints.

# To Question 1

**Theorem 1**

Given $\min\{\mathbf{w}\mathbf{x} : H\mathbf{x} = \mathbf{b},\ 0 \leq \mathbf{x} \leq \mathbf{u},\ \mathbf{x} \in \mathbb{Z}^N\}$ with $\mathsf{OPT}$, $H \in \mathbb{Q}^{M \times N}$. For arbitrarily small $\varepsilon > 0$, there exists an algorithm that returns a near-feasible solution $\tilde{\mathbf{x}}$:

- $f(M, \varepsilon) \cdot \mathrm{poly}(|I|)$ (polynomial time for constant number of rows)
- $\|H\tilde{\mathbf{x}} - \mathbf{b}\|_\infty \leq \varepsilon\Delta$ (slightly violating the constraints)
- $\mathbf{w}\tilde{\mathbf{x}} \leq \mathsf{OPT}$ (optimality)

Specifically, the running time is $2^{(\frac{M}{\varepsilon})^{\mathcal{O}(M)}} \cdot \mathrm{poly}(|I|)$.

# Application

- Knapsack with negative inputs
  (take the hard constraint as the objective)
    - Take $M = 1$, $\mathbf{u} = (1, 1, ..., 1)$, and $H\mathbf{x} = \sum_{j=1}^{n} v_j x_j$.
      IP is reduced to:

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n} w_j x_j \\
& \sum_{j=1}^{n} v_j x_j \geq V \\
& x_j \in \{0, 1\}, \qquad 1 \leq i \leq n
\end{aligned}
$$

# Application

- Knapsack with negative inputs
  (take the hard constraint as the objective)
  - Take $M = 1$, $\mathbf{u} = (1, 1, ..., 1)$, and $H\mathbf{x} = \sum_{j=1}^{n} v_j x_j$.
    IP is reduced to:

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n} w_j x_j \\
& \sum_{j=1}^{n} v_j x_j \geq V \\
& x_j \in \{0, 1\}, \qquad 1 \leq i \leq n
\end{aligned}
$$

- Multidimensional knapsack with resource augmentation

# Turn to Block-Structured IP

When $H$ is $n$-fold,

$$H = \begin{pmatrix} D^1 & D^2 & \cdots & D^n \\ A^1 & 0 & & 0 \\ 0 & A^2 & & 0 \\ & & \ddots & \\ 0 & 0 & & A^n \end{pmatrix}$$

$$H\mathbf{x} = \mathbf{b}$$

$$\Rightarrow$$

$$\sum_{i=1}^{n} D^i \mathbf{x}^i = \mathbf{b}^0$$
$$A^i \mathbf{x}^i = \mathbf{b}^i, \qquad \forall 1 \le i \le n$$

# To Question 2

**Theorem 2 (Approximating $n$-fold)**

Given $\min\{\mathbf{w}\mathbf{x} : \sum_{i=1}^{n} D^i\mathbf{x}^i = \mathbf{b}^0, A^i\mathbf{x}^i = \mathbf{b}^i, 1 \le i \le n, \mathbf{0} \le \mathbf{x} \le \mathbf{u}, \mathbf{x} \in \mathbb{Z}^{nt}\}$ with $\mathsf{OPT}$, where $A^i \in \mathbb{Q}_{\ge 0}^{s_A \times t_A}$, $D^i \in \mathbb{Q}_{\ge 0}^{s_D \times t_D}$, and $t_A = t_D = t$.
Let $\varepsilon > 0$ be arbitrarily small. **Then**, there exists an algorithm that returns a near-feasible solution $\tilde{\mathbf{x}}$:

- $f(s_A, s_D, t, \varepsilon) \cdot \mathrm{poly}(|I|)$ (running time)
- $(1-\varepsilon)\mathbf{b}^0 \le \sum_{i=1}^{n} D^i\tilde{\mathbf{x}}^i \le (1+\varepsilon)\mathbf{b}^0$,
  $(1-\varepsilon)\mathbf{b}^i \le A^i\tilde{\mathbf{x}}^i \le (1+\varepsilon)\mathbf{b}^i$, $1 \le i \le n$ (constraints)
- $\mathbf{w}\tilde{\mathbf{x}} \le \mathsf{OPT}$ (optimality)

Running time: $2^{2^{(s_D/\varepsilon)^{\mathcal{O}(s_D)} \cdot (s_A t/\varepsilon)^{\mathcal{O}(t)}}} \cdot \mathrm{poly}(|I|)$

**Theorem 3 (Approximating multichoice IP)**

Given $\min\{\mathbf{w}\mathbf{x} : \sum_{i=1}^{n} D^i \mathbf{x}^i = \mathbf{b}^0, \mathbf{x}^i \in \mathcal{P}^i, 1 \leq i \leq n, \mathbf{x} \in \mathbb{Z}^{nt}\}$ with $\mathsf{OPT}$, where $D^i \in \mathbb{Q}^{s\times t}$ and $\mathcal{P}^i$ is an arbitrary set of integer vectors. Let $\varepsilon > 0$ be arbitrarily small, $\kappa = \max_{\mathbf{y}\in\cup_{i=1}^{n}\mathcal{P}^i} \|\mathbf{y}\|_\infty$. **Then**, there exists an algorithm that returns a near-feasible solution $\tilde{\mathbf{x}}$:

- $f(s,t,\kappa,\varepsilon)\cdot \mathrm{poly}(|I|)$ (running time)
- $\|\sum_{i=1}^{n} D^i\tilde{\mathbf{x}}^i - \mathbf{b}^0\|_\infty \leq \varepsilon\Delta$ (constraints)
- $\mathbf{w}\tilde{\mathbf{x}} \leq \mathsf{OPT}$ (optimality)

where $\Delta = \max_i \|D^i\|_\infty$.

Running time: $2^{(\frac{s\kappa^t}{\varepsilon})^{\mathcal{O}(s\kappa^t)}} \cdot \mathrm{poly}(|I|)$

# Application

## Unrelated machine scheduling $Rm||C_{\max}$

- $n$ jobs and $m$ machines
- $p_{ih}$: the processing time of job $i$ on machine $h$
- Goal: Minimizing the makespan

# Application

**Unrelated machine scheduling $Rm||C_{\max}$**

- $n$ jobs and $m$ machines
- $p_{ih}$: the processing time of job $i$ on machine $h$
- Goal: Minimizing the makespan

– Let $x_h^i \in \{0,1\}$ denote whether job $i$ is assigned to machine $h$.
$\left\{\mathbf{x} : \sum_{i=1}^n p_{ih} x_h^i \leq T, \forall 1 \leq h \leq m, \mathbf{x}^i \in \mathcal{P}^i, 1 \leq i \leq n, \mathbf{x} \in \mathbb{Z}^{nt}\right\}$,
where $\mathcal{P}^i := \{\mathbf{x}^i : x_1^i + x_2^i + \cdots + x_m^i = 1, x_h^i \in \{0,1\}\}$.

# Application

**Unrelated machine scheduling $Rm||C_{\max}$**

- $n$ jobs and $m$ machines
- $p_{ih}$: the processing time of job $i$ on machine $h$
- Goal: Minimizing the makespan

– Let $x_h^i \in \{0,1\}$ denote whether job $i$ is assigned to machine $h$.
$\left\{\mathbf{x} : \sum_{i=1}^{n} p_{ih} x_h^i \le T, \forall 1 \le h \le m, \mathbf{x}^i \in \mathcal{P}^i, 1 \le i \le n, \mathbf{x} \in \mathbb{Z}^{nt}\right\}$,
where $\mathcal{P}^i := \{\mathbf{x}^i : x_1^i + x_2^i + \cdots + x_m^i = 1, x_h^i \in \{0,1\}\}$.

– Observe $D^i \in \mathbb{Z}^{m \times m}$ and $\kappa = 1$. Theorem 3 implies a PTAS:

- $2^{(\frac{m}{\varepsilon})^{\mathcal{O}(m)}} \cdot n^{\mathcal{O}(1)}$ (running time)
- a feasible schedule $\tilde{\mathbf{x}}$: makespan $\le T^* + \varepsilon \cdot \max_{i,h} p_{ih}$

# Application

**Unrelated machine scheduling $Rm||C_{\max}$**

- $n$ jobs and $m$ machines
- $p_{ih}$: the processing time of job $i$ on machine $h$
- Goal: Minimizing the makespan

– Let $x_h^i \in \{0,1\}$ denote whether job $i$ is assigned to machine $h$.
$\left\{\mathbf{x} : \sum_{i=1}^n p_{ih} x_h^i \leq T, \forall 1 \leq h \leq m, \mathbf{x}^i \in \mathcal{P}^i, 1 \leq i \leq n, \mathbf{x} \in \mathbb{Z}^{nt}\right\}$,
where $\mathcal{P}^i := \{\mathbf{x}^i : x_1^i + x_2^i + \cdots + x_m^i = 1, x_h^i \in \{0,1\}\}$.

– Observe $D^i \in \mathbb{Z}^{m\times m}$ and $\kappa = 1$. Theorem 3 implies a PTAS:

- $2^{(\frac{m}{\varepsilon})^{\mathcal{O}(m)}} \cdot n^{\mathcal{O}(1)}$ (running time)
- a feasible schedule $\tilde{\mathbf{x}}$: makespan $\leq T^* + \varepsilon \cdot \max_{i,h} p_{ih}$
- Best-known previous result: $(1/\varepsilon)^{\mathcal{O}(m)}$-time for makespan $(1+\varepsilon)T^*$ [Jansen et al., 2010]

# Application

**Generalized assignment $Rm|c_{ih}|C_{\max}$**

- $n$ jobs and $m$ machines
- $p_{ih}$: the processing time of job $i$ on machine $h$
- $c_{ih}$: the cost of scheduling job $i$ on machine $h$
- $C$: cost budget for scheduling all jobs
- Goal: Minimizing the makespan subject to the cost budget

– Theorem 3 implies a PTAS:
  - $2^{(\frac{m}{\varepsilon})^{\mathcal{O}(m)}} \cdot n^{\mathcal{O}(1)}$ (running time)
  - a feasible schedule $\tilde{\mathbf{x}}$: makespan $\leq T^* + \varepsilon \cdot \max_{i,h} p_{ih}$

# Application

**Generalized assignment $Rm|c_{ih}|C_{\max}$**

- $n$ jobs and $m$ machines
- $p_{ih}$: the processing time of job $i$ on machine $h$
- $c_{ih}$: the cost of scheduling job $i$ on machine $h$
- $C$: cost budget for scheduling all jobs
- Goal: Minimizing the makespan subject to the cost budget

– Theorem 3 implies a PTAS:

  - $2^{(\frac{m}{\varepsilon})^{\mathcal{O}(m)}} \cdot n^{\mathcal{O}(1)}$ (running time)
  - a feasible schedule $\tilde{\mathbf{x}}$: makespan $\leq T^* + \varepsilon \cdot \max_{i,h} p_{ih}$
  - Best-known prior result: $\mathcal{O}(n(n/\varepsilon)^m)$-time for makespan $(1+\varepsilon)T^*$ [Angel et al., 2001]

# Proof Sketch of Theorem 1

**Theorem 1**

Given $\min\{\mathbf{w}\mathbf{x} : H\mathbf{x} = \mathbf{b},\ 0 \leq \mathbf{x} \leq \mathbf{u},\ \mathbf{x} \in \mathbb{Z}^N\}$ with $\mathsf{OPT}$, $H \in \mathbb{Q}^{M \times N}$. For arbitrarily small $\varepsilon > 0$, there exists an algorithm that returns a near-feasible solution $\tilde{\mathbf{x}}$:

- $f(M, \varepsilon) \cdot \mathrm{poly}(|I|)$ (running time)
- $\|H\tilde{\mathbf{x}} - \mathbf{b}\|_\infty \leq \varepsilon\Delta$ (constraints)
- $\mathbf{w}\tilde{\mathbf{x}} \leq \mathsf{OPT}$ (optimality)

# Proof Sketch of Theorem 1

**Theorem 1**

Given $\min\{\mathbf{w}\mathbf{x} : H\mathbf{x} = \mathbf{b},\ 0 \le \mathbf{x} \le \mathbf{u},\ \mathbf{x} \in \mathbb{Z}^N\}$ with $\mathsf{OPT}$, $H \in \mathbb{Q}^{M\times N}$. For arbitrarily small $\varepsilon > 0$, there exists an algorithm that returns a near-feasible solution $\tilde{\mathbf{x}}$:

- $f(M, \varepsilon) \cdot \mathrm{poly}(|I|)$ (running time)
- $\|H\tilde{\mathbf{x}} - \mathbf{b}\|_\infty \le \varepsilon\Delta$ (constraints)
- $\mathbf{w}\tilde{\mathbf{x}} \le \mathsf{OPT}$ (optimality)

Let $H = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N)$ where $\mathbf{h}_j \in [-\Delta, \Delta]^M$. So $H\mathbf{x} = \sum \mathbf{h}_i x_i$.

# Proof Sketch of Theorem 1

**Theorem 1**

Given $\min\{\mathbf{w}\mathbf{x} : H\mathbf{x} = \mathbf{b},\, 0 \le \mathbf{x} \le \mathbf{u},\, \mathbf{x} \in \mathbb{Z}^N\}$ with $\mathsf{OPT}$, $H \in \mathbb{Q}^{M \times N}$. For arbitrarily small $\varepsilon > 0$, there exists an algorithm that returns a near-feasible solution $\tilde{\mathbf{x}}$:

- $f(M, \varepsilon) \cdot \mathrm{poly}(|I|)$ (running time)
- $\|H\tilde{\mathbf{x}} - \mathbf{b}\|_\infty \le \varepsilon\Delta$ (constraints)
- $\mathbf{w}\tilde{\mathbf{x}} \le \mathsf{OPT}$ (optimality)

Let $H = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N)$ where $\mathbf{h}_j \in [-\Delta, \Delta]^M$. So $H\mathbf{x} = \sum \mathbf{h}_i x_i$.

- $\mathbf{h}_i$ and $\mathbf{h}_j$ are of the same type if $\|\mathbf{h}_i - \mathbf{h}_j\|_\infty \le f(\varepsilon)\Delta$.
  There are only $\mathcal{O}_{M,\varepsilon}(1)$ distinct types of vectors.

# Proof Sketch of Theorem 1

**Theorem 1**

Given $\min\{\mathbf{w}\mathbf{x} : H\mathbf{x} = \mathbf{b},\ 0 \leq \mathbf{x} \leq \mathbf{u},\ \mathbf{x} \in \mathbb{Z}^N\}$ with $\mathsf{OPT}$, $H \in \mathbb{Q}^{M\times N}$. For arbitrarily small $\varepsilon > 0$, there exists an algorithm that returns a near-feasible solution $\tilde{\mathbf{x}}$:

- $f(M, \varepsilon) \cdot \mathrm{poly}(|I|)$ (running time)
- $\|H\tilde{\mathbf{x}} - \mathbf{b}\|_\infty \leq \varepsilon\Delta$ (constraints)
- $\mathbf{w}\tilde{\mathbf{x}} \leq \mathsf{OPT}$ (optimality)

Let $H = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N)$ where $\mathbf{h}_j \in [-\Delta, \Delta]^M$. So $H\mathbf{x} = \sum \mathbf{h}_i x_i$.

- $\mathbf{h}_i$ and $\mathbf{h}_j$ are of the same type if $\|\mathbf{h}_i - \mathbf{h}_j\|_\infty \leq f(\varepsilon)\Delta$.
  There are only $\mathcal{O}_{M,\varepsilon}(1)$ distinct types of vectors.
- Relax $x_i$'s, but introduce one integer variable for each type.

# Proof Sketch of Theorem 1

**Theorem 1**

Given $\min\{\mathbf{w}\mathbf{x} : H\mathbf{x} = \mathbf{b},\ 0 \le \mathbf{x} \le \mathbf{u},\ \mathbf{x} \in \mathbb{Z}^N\}$ with $\mathsf{OPT}$, $H \in \mathbb{Q}^{M \times N}$. For arbitrarily small $\varepsilon > 0$, there exists an algorithm that returns a near-feasible solution $\tilde{\mathbf{x}}$:

- $f(M, \varepsilon) \cdot \mathrm{poly}(|I|)$ (running time)
- $\|H\tilde{\mathbf{x}} - \mathbf{b}\|_\infty \le \varepsilon\Delta$ (constraints)
- $\mathbf{w}\tilde{\mathbf{x}} \le \mathsf{OPT}$ (optimality)

Let $H = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N)$ where $\mathbf{h}_j \in [-\Delta, \Delta]^M$. So $H\mathbf{x} = \sum \mathbf{h}_i x_i$.

- $\mathbf{h}_i$ and $\mathbf{h}_j$ are of the same type if $\|\mathbf{h}_i - \mathbf{h}_j\|_\infty \le f(\varepsilon)\Delta$. There are only $\mathcal{O}_{M,\varepsilon}(1)$ distinct types of vectors.
- Relax $x_i$'s, but introduce one integer variable for each type.
- Solve the mixed-IP, and round the fractional variables.

# Proof Sketch of Theorem 1

**Theorem 1**

Given $\min\{\mathbf{w}\mathbf{x} : H\mathbf{x} = \mathbf{b},\ 0 \le \mathbf{x} \le \mathbf{u},\ \mathbf{x} \in \mathbb{Z}^N\}$ with $\mathsf{OPT}$, $H \in \mathbb{Q}^{M\times N}$. For arbitrarily small $\varepsilon > 0$, there exists an algorithm that returns a near-feasible solution $\tilde{\mathbf{x}}$:

- $f(M, \varepsilon) \cdot \mathrm{poly}(|I|)$ (running time)
- $\|H\tilde{\mathbf{x}} - \mathbf{b}\|_\infty \le \varepsilon\Delta$ (constraints)
- $\mathbf{w}\tilde{\mathbf{x}} \le \mathsf{OPT}$ (optimality)

Let $H = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N)$ where $\mathbf{h}_j \in [-\Delta, \Delta]^M$. So $H\mathbf{x} = \sum \mathbf{h}_i x_i$.

- $\mathbf{h}_i$ and $\mathbf{h}_j$ are of the same type if $\|\mathbf{h}_i - \mathbf{h}_j\|_\infty \le f(\varepsilon)\Delta$. There are only $\mathcal{O}_{M,\varepsilon}(1)$ distinct types of vectors.
- Relax $x_i$'s, but introduce one integer variable for each type.
- Solve the mixed-IP, and round the fractional variables.

**Challenge:**
make sure only a few variables can take fractional values.

# Proof Sketch of Theorem 1

- Cut $[-\Delta, \Delta]^M$ into $\mathcal{O}_{\delta,M}(1)$ boxes.

# Proof Sketch of Theorem 1

- Cut $[-\Delta, \Delta]^M$ into $\mathcal{O}_{\delta,M}(1)$ boxes.
- Divide all $\mathbf{h}_j$'s into $\mathcal{O}_{\delta,M}(1)$ different boxes. In the $k$-th box, we find a vector $(\mathbf{v}_k)$. Define $\tilde{\mathbf{h}}_j = \mathbf{h}_j - \mathbf{v}_k$, and then $\|\tilde{\mathbf{h}}_j\|_\infty \leq \delta\Delta$.

# Proof Sketch of Theorem 1

- Cut $[-\Delta, \Delta]^M$ into $\mathcal{O}_{\delta,M}(1)$ boxes.
- Divide all $\mathbf{h}_j$'s into $\mathcal{O}_{\delta,M}(1)$ different boxes. In the $k$-th box, we find a vector $(\mathbf{v}_k)$. Define $\tilde{\mathbf{h}}_j = \mathbf{h}_j - \mathbf{v}_k$, and then $\|\tilde{\mathbf{h}}_j\|_\infty \leq \delta\Delta$.

$$H\mathbf{x} = \sum_{j=1}^{n} \mathbf{h}_j x_j = \sum_{k=1}^{\mathcal{O}_{\delta,M}(1)} \sum_{j \in I_k} (\mathbf{v}_k + \tilde{\mathbf{h}}_j) x_j$$

# Proof Sketch of Theorem 1

- Cut $[-\Delta, \Delta]^M$ into $\mathcal{O}_{\delta,M}(1)$ boxes.
- Divide all $\mathbf{h}_j$'s into $\mathcal{O}_{\delta,M}(1)$ different boxes. In the $k$-th box, we find a vector $(\mathbf{v}_k)$. Define $\tilde{\mathbf{h}}_j = \mathbf{h}_j - \mathbf{v}_k$, and then $\|\tilde{\mathbf{h}}_j\|_\infty \leq \delta\Delta$.

$$H\mathbf{x} = \sum_{j=1}^{n} \mathbf{h}_j x_j = \sum_{k=1}^{\mathcal{O}_{\delta,M}(1)} \sum_{j \in I_k} (\mathbf{v}_k + \tilde{\mathbf{h}}_j) x_j$$

- Let $y_k := \sum_{j \in I_k} x_j$,

$$\sum_{k=1}^{\mathcal{O}_{\delta,M}(1)} \mathbf{v}_k y_k + \sum_{k=1}^{\mathcal{O}_{\delta,M}(1)} \sum_{j \in I_k} \tilde{\mathbf{h}}_j x_j = \mathbf{b}$$

# Proof Sketch of Theorem 1

- Cut $[-\Delta, \Delta]^M$ into $\mathcal{O}_{\delta,M}(1)$ boxes.
- Divide all $\mathbf{h}_j$'s into $\mathcal{O}_{\delta,M}(1)$ different boxes. In the $k$-th box, we find a vector $(\mathbf{v}_k)$. Define $\tilde{\mathbf{h}}_j = \mathbf{h}_j - \mathbf{v}_k$, and then $\|\tilde{\mathbf{h}}_j\|_\infty \leq \delta\Delta$.

$$H\mathbf{x} = \sum_{j=1}^{n} \mathbf{h}_j x_j = \sum_{k=1}^{\mathcal{O}_{\delta,M}(1)} \sum_{j \in I_k} (\mathbf{v}_k + \tilde{\mathbf{h}}_j) x_j$$

- Let $y_k := \sum_{j \in I_k} x_j$,

$$\sum_{k=1}^{\mathcal{O}_{\delta,M}(1)} \mathbf{v}_k y_k + \sum_{k=1}^{\mathcal{O}_{\delta,M}(1)} \sum_{j \in I_k} \tilde{\mathbf{h}}_j x_j = \mathbf{b}$$

- Solve the MIP $\implies y_k^* \in \mathbb{Z}$, $x_j^* \in \mathbb{R}$

# Proof Sketch of Theorem 1

$$
\begin{aligned}
\text{(LP)} \quad \min \quad & \sum_{j=1}^{n} w_j x_j \\
& \sum_{k=1}^{(2/\delta)^m} \sum_{j \in I_k} \tilde{\mathbf{h}}_j x_j = \sum_{k=1}^{(2/\delta)^m} \sum_{j \in I_k} \tilde{\mathbf{h}}_j x_j^* && \text{(1a)} \\
& \sum_{j \in I_k} x_j = y_k^*, \quad \forall 1 \leq k \leq (2/\delta)^m && \text{(1b)} \\
& l_j \leq x_j \leq u_j, x_j \in \mathbb{R}, \quad \forall j \in [n]
\end{aligned}
$$

# Proof Sketch of Theorem 1

$$
\begin{aligned}
\text{(LP)} \quad \min \quad & \sum_{j=1}^{n} w_j x_j \\
& \sum_{k=1}^{(2/\delta)^m} \sum_{j \in I_k} \tilde{\mathbf{h}}_j x_j = \sum_{k=1}^{(2/\delta)^m} \sum_{j \in I_k} \tilde{\mathbf{h}}_j x_j^* && \text{(1a)} \\
& \sum_{j \in I_k} x_j = y_k^*, \quad \forall 1 \le k \le (2/\delta)^m && \text{(1b)} \\
& l_j \le x_j \le u_j, x_j \in \mathbb{R}, \quad \forall j \in [n]
\end{aligned}
$$

- (1b) contains many constraints, but have a simple structure. We can show that only $\mathcal{O}(M)$ $x_j$'s take a fractional value in a vertex solution.

# Proof Sketch of Theorem 1

$$
\begin{aligned}
\text{(LP)} \quad \min \quad & \sum_{j=1}^{n} w_j x_j \\
& \sum_{k=1}^{(2/\delta)^m} \sum_{j \in I_k} \tilde{\mathbf{h}}_j x_j = \sum_{k=1}^{(2/\delta)^m} \sum_{j \in I_k} \tilde{\mathbf{h}}_j x_j^* && \text{(1a)} \\
& \sum_{j \in I_k} x_j = y_k^*, \quad \forall 1 \leq k \leq (2/\delta)^m && \text{(1b)} \\
& l_j \leq x_j \leq u_j, x_j \in \mathbb{R}, \quad \forall j \in [n]
\end{aligned}
$$

- (1b) contains many constraints, but have a simple structure. We can show that only $\mathcal{O}(M)$ $x_j$'s take a fractional value in a vertex solution.
- Round these $O(M)$ fractional variables up or down at the cost of introducing an error of

$$M\delta\Delta \leq \varepsilon\Delta$$

□

# Proof Sketch of Theorem 3

**Theorem 3 (Approximating multichoice IP)**

Given $\min\{\mathbf{w}\mathbf{x} : \sum_{i=1}^{n} D^i\mathbf{x}^i = \mathbf{b}^0, \mathbf{x}^i \in \mathcal{P}_i,\ 1 \le i \le n, \mathbf{x} \in \mathbb{Z}^{nt}\}$ with $\mathsf{OPT}$, where $D_i \in \mathbb{Q}^{s\times t}$ and $\mathcal{P}_i$ is an arbitrary set of integer vectors. Let $\varepsilon > 0$ be arbitrarily small, $\kappa = \max_{\mathbf{y}\in\cup_{i=1}^{n}\mathcal{P}_i} \|\mathbf{y}\|_\infty$. **Then**, there exists an algorithm that returns a near-feasible solution $\tilde{\mathbf{x}}$:

- $f(s,t,\kappa,\varepsilon)\cdot \mathrm{poly}(|I|)$ (running time)
- $\|\sum_{i=1}^{n} D_i\tilde{\mathbf{x}}^i - \mathbf{b}^0\|_\infty \le \varepsilon\Delta$ (constraints)
- $\mathbf{w}\tilde{\mathbf{x}} \le \mathsf{OPT}$ (optimality)

Each $\mathbf{x}^i$ can take $\mathcal{O}_{s,t,\kappa}(1)$ different possible values (constant number of choices). Let $\mathcal{D}^i = (D^i\mathbf{x}^i[1], D^i\mathbf{x}^i[2], ...)$, which is fixed.

# Proof Sketch of Theorem 3

**Theorem 3 (Approximating multichoice IP)**

Given $\min\{\mathbf{w}\mathbf{x} : \sum_{i=1}^{n} D^i \mathbf{x}^i = \mathbf{b}^0, \mathbf{x}^i \in \mathcal{P}_i,\ 1 \le i \le n, \mathbf{x} \in \mathbb{Z}^{nt}\}$ with $\mathsf{OPT}$, where $D_i \in \mathbb{Q}^{s \times t}$ and $\mathcal{P}_i$ is an arbitrary set of integer vectors. Let $\varepsilon > 0$ be arbitrarily small, $\kappa = \max_{\mathbf{y} \in \cup_{i=1}^{n} \mathcal{P}_i} \|\mathbf{y}\|_\infty$. **Then**, there exists an algorithm that returns a near-feasible solution $\tilde{\mathbf{x}}$:

- $f(s, t, \kappa, \varepsilon) \cdot \mathrm{poly}(|I|)$ (running time)
- $\| \sum_{i=1}^{n} D_i \tilde{\mathbf{x}}^i - \mathbf{b}^0 \|_\infty \le \varepsilon\Delta$ (constraints)
- $\mathbf{w}\tilde{\mathbf{x}} \le \mathsf{OPT}$ (optimality)

Each $\mathbf{x}^i$ can take $\mathcal{O}_{s,t,\kappa}(1)$ different possible values (constant number of choices). Let $\mathcal{D}^i = (D^i\mathbf{x}^i[1], D^i\mathbf{x}^i[2], \ldots)$, which is fixed.

- $\|\mathcal{D}^i\|_\infty = \mathcal{O}_{s,t,\kappa,\varepsilon}(\Delta)$. Cut the box $[-\|\mathcal{D}^i\|_\infty, \|\mathcal{D}^i\|_\infty]^s$ into $\mathcal{O}_{s,t,\kappa,\varepsilon}(1)$ small boxes and introduce an integer variable each.

# Proof Sketch of Theorem 3

**Theorem 3 (Approximating multichoice IP)**

Given $\min\{\mathbf{w}\mathbf{x} : \sum_{i=1}^{n} D^i \mathbf{x}^i = \mathbf{b}^0, \mathbf{x}^i \in \mathcal{P}_i, 1 \le i \le n, \mathbf{x} \in \mathbb{Z}^{nt}\}$ with $\mathsf{OPT}$, where $D_i \in \mathbb{Q}^{s \times t}$ and $\mathcal{P}_i$ is an arbitrary set of integer vectors. Let $\varepsilon > 0$ be arbitrarily small, $\kappa = \max_{\mathbf{y} \in \cup_{i=1}^{n} \mathcal{P}_i} \|\mathbf{y}\|_\infty$. **Then**, there exists an algorithm that returns a near-feasible solution $\tilde{\mathbf{x}}$:

- $f(s, t, \kappa, \varepsilon) \cdot \mathrm{poly}(|I|)$ (running time)
- $\| \sum_{i=1}^{n} D_i \tilde{\mathbf{x}}^i - \mathbf{b}^0 \|_\infty \le \varepsilon \Delta$ (constraints)
- $\mathbf{w}\tilde{\mathbf{x}} \le \mathsf{OPT}$ (optimality)

Each $\mathbf{x}^i$ can take $\mathcal{O}_{s,t,\kappa}(1)$ different possible values (constant number of choices). Let $\mathcal{D}^i = (D^i\mathbf{x}^i[1], D^i\mathbf{x}^i[2], \ldots)$, which is fixed.

- $\|\mathcal{D}^i\|_\infty = \mathcal{O}_{s,t,\kappa,\varepsilon}(\Delta)$. Cut the box $[-\|\mathcal{D}^i\|_\infty, \|\mathcal{D}^i\|_\infty]^s$ into $\mathcal{O}_{s,t,\kappa,\varepsilon}(1)$ small boxes and introduce an integer variable each.
- Relax $x_i$'s, and solve the mixed-IP.

# Proof Sketch of Theorem 3

**Theorem 3 (Approximating multichoice IP)**

Given $\min\{\mathbf{w}\mathbf{x} : \sum_{i=1}^{n} D^i\mathbf{x}^i = \mathbf{b}^0, \mathbf{x}^i \in \mathcal{P}_i,\ 1 \le i \le n, \mathbf{x} \in \mathbb{Z}^{nt}\}$ with $\mathsf{OPT}$, where $D_i \in \mathbb{Q}^{s\times t}$ and $\mathcal{P}_i$ is an arbitrary set of integer vectors. Let $\varepsilon > 0$ be arbitrarily small, $\kappa = \max_{\mathbf{y}\in\cup_{i=1}^{n}\mathcal{P}_i} \|\mathbf{y}\|_\infty$. **Then**, there exists an algorithm that returns a near-feasible solution $\tilde{\mathbf{x}}$:

- $f(s,t,\kappa,\varepsilon)\cdot \mathrm{poly}(|I|)$ (running time)
- $\|\sum_{i=1}^{n} D_i\tilde{\mathbf{x}}^i - \mathbf{b}^0\|_\infty \le \varepsilon\Delta$ (constraints)
- $\mathbf{w}\tilde{\mathbf{x}} \le \mathsf{OPT}$ (optimality)

Each $\mathbf{x}^i$ can take $\mathcal{O}_{s,t,\kappa}(1)$ different possible values (constant number of choices). Let $\mathcal{D}^i = (D^i\mathbf{x}^i[1], D^i\mathbf{x}^i[2], ...)$, which is fixed.

- $\|\mathcal{D}^i\|_\infty = \mathcal{O}_{s,t,\kappa,\varepsilon}(\Delta)$. Cut the box $[-\|\mathcal{D}^i\|_\infty, \|\mathcal{D}^i\|_\infty]^s$ into $\mathcal{O}_{s,t,\kappa,\varepsilon}(1)$ small boxes and introduce an integer variable each.
- Relax $x_i$'s, and solve the mixed-IP.
- Round the fractional variables.

# Proof Sketch of Theorem 3

**Theorem 3 (Approximating multichoice IP)**

Given $\min\{\mathbf{w}\mathbf{x} : \sum_{i=1}^{n} D^i\mathbf{x}^i = \mathbf{b}^0, \mathbf{x}^i \in \mathcal{P}_i, 1 \le i \le n, \mathbf{x} \in \mathbb{Z}^{nt}\}$ with $\mathsf{OPT}$, where $D_i \in \mathbb{Q}^{s\times t}$ and $\mathcal{P}_i$ is an arbitrary set of integer vectors. Let $\varepsilon > 0$ be arbitrarily small, $\kappa = \max_{\mathbf{y}\in\cup_{i=1}^{n}\mathcal{P}_i} \|\mathbf{y}\|_\infty$. **Then**, there exists an algorithm that returns a near-feasible solution $\tilde{\mathbf{x}}$:

- $f(s,t,\kappa,\varepsilon)\cdot \mathrm{poly}(|I|)$ (running time)
- $\|\sum_{i=1}^{n} D_i\tilde{\mathbf{x}}^i - \mathbf{b}^0\|_\infty \le \varepsilon\Delta$ (constraints)
- $\mathbf{w}\tilde{\mathbf{x}} \le \mathsf{OPT}$ (optimality)

Each $\mathbf{x}^i$ can take $\mathcal{O}_{s,t,\kappa}(1)$ different possible values (constant number of choices). Let $\mathcal{D}^i = (D^i\mathbf{x}^i[1], D^i\mathbf{x}^i[2], \ldots)$, which is fixed.

- $\|\mathcal{D}^i\|_\infty = \mathcal{O}_{s,t,\kappa,\varepsilon}(\Delta)$. Cut the box $[-\|\mathcal{D}^i\|_\infty, \|\mathcal{D}^i\|_\infty]^s$ into $\mathcal{O}_{s,t,\kappa,\varepsilon}(1)$ small boxes and introduce an integer variable each.
- Relax $x_i$'s, and solve the mixed-IP.
- Round the fractional variables.

  Make sure only a few variables can take fractional values.

# Proof Sketch of Theorem 3

- The LP derived from the mixed-IP can be written as $A_\Sigma \mathbf{z} = \mathbf{b}_A$ and $C\mathbf{z} = \mathbf{b}_C$, where
  - $C\mathbf{z} = \mathbf{b}_C$ consists of a few constraints
  - $A_\Sigma = \begin{pmatrix} A_1 & & \\ & \ddots & \\ & & A_\rho \end{pmatrix}$ where

$$A_k = \begin{array}{l} \left(\begin{array}{ccc|ccc|c|ccc} 1 & & & 1 & & & & 1 & & \\ & \ddots & & & \ddots & & \cdots & & \ddots & \\ & & 1 & & & 1 & & & & 1 \\ \hline 1 & \cdots & 1 & & & & & & & \\ & & & 1 & \cdots & 1 & & & & \\ & & & & & & \ddots & & & \\ & & & & & & & 1 & \cdots & 1 \end{array}\right) \begin{array}{l} \left.\vphantom{\begin{array}{c}1\\ \ddots\\ 1\end{array}}\right\} |I_k| \text{ rows} \\ \left.\vphantom{\begin{array}{c}1\\ 1\\ \ddots\\ 1\end{array}}\right\} \tau \text{ rows} \end{array} \\ \quad \underbrace{\hphantom{1 \cdots 1}}_{|I_k| \text{ columns}} \quad \underbrace{\hphantom{1 \cdots 1}}_{|I_k| \text{ columns}} \qquad \underbrace{\hphantom{1 \cdots 1}}_{|I_k| \text{ columns}} \end{array}$$

- $|I_k|$ may depend on $n$ and is huge. But we can leverage the specific block structure of $A_\Sigma$ to bound the number of variables taking a fractional value in a vertex solution.

Thanks!