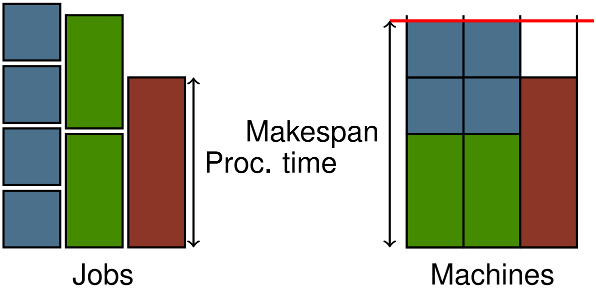
New Algorithmic Results for Scheduling via Intger Linear Programming

Klaus Jansen, University of Kiel

Joint Work with Sebastian Berndt, Lin Chen, Max Deppert, Kim-Manuel Klein, Lars Rohwedder, José Verschae and Gouchuan Zhang

Scheduling on Identical Machines $P||C_{max}$:

- Given: n jobs with processing times p_j
- and m machines
- Objective: Minimize makespan (maximum machine load)



Complexity

- Strongly NP-hard
- If $P \neq NP$, then there is no FPTAS (with running time polynomial in $\frac{1}{\varepsilon}$)
- If the *Exponential Time Hypothesis* holds, there is no EPTAS with running time $2^{(\frac{1}{\varepsilon})^{1-\delta}} + \text{poly}(n)$ [Chen, Jansen, Zhang '13]

Approximation Schemes

There is a PTAS with running time:

- $n^{O(\frac{1}{\varepsilon^2})}$ [Hochbaum & Shmoys '87]

There is an EPTAS with running time:

- $2^{2^{\widetilde{O}(\frac{1}{\varepsilon})}} + O(n \log n)$ [Alon et al. '98 & H. & S. '96]
- $2^{\widetilde{O}(\frac{1}{\varepsilon^2})} + O(n \log n)$ [Jansen '10]
- $2^{O(\frac{1}{\varepsilon} \log^4(\frac{1}{\varepsilon}))} + O(n \log n)$ [Jansen, Klein, Verschae '16]
- $2^{O(\frac{1}{\varepsilon} \log^2(\frac{1}{\varepsilon}))} + O(n)$ [Jansen, Rohwedder '19]
- $2^{O(\frac{1}{\varepsilon} \log(\frac{1}{\varepsilon}) \log\log(\frac{1}{\varepsilon}))} + O(n)$ [Berndt, Deppert, Jansen, Rohwedder '22]

General Strategy:

General scheme for designing a PTAS:

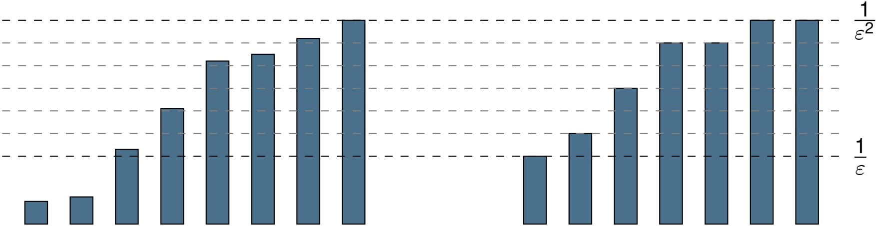
1. Guess the makespan T of the optimal solution.
2. Round instance $\rightsquigarrow (1+\varepsilon)$ multiplicative loss in objective.
3. Solve the rounded instance using an ILP formulation.

Rounding:

Lemma (Rounding and scaling)

$T = 1/\varepsilon^2$ and jobs sizes belong to $\Pi = \{\pi_1, \ldots, \pi_d\}$:

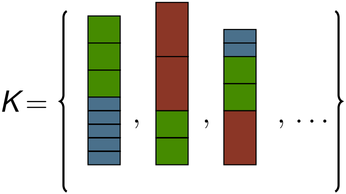
- $\Pi \subseteq \{\frac{1}{\varepsilon}, \frac{1}{\varepsilon}+1, \ldots, \frac{1}{\varepsilon^2}\}$ *and,* $\rightarrow$ *integer numbers*
- $|\Pi| = O(\frac{1}{\varepsilon}\log(\frac{1}{\varepsilon})) = \widetilde{O}(\frac{1}{\varepsilon})$. $\rightarrow$ *few sizes*



Configurations:

A *configuration* represents one possibility of assigning jobs from Π to a single machine.

Example (The set of configurations)

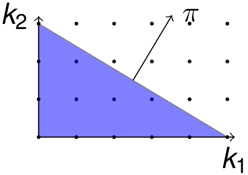


Configurations:

Knapsack polytope

$$P = \{k \in \mathbb{R}_{\geq 0}^{|\Pi|} : k^t \cdot \pi \leq T\}$$

Polyhedral view



Configurations:

Set of configurations

$$K := P \cap \mathbb{Z}_{\geq 0}^{|\Pi|}$$

Observation 1

$$|K| \leq (T+1)^{|\Pi|} = 2^{O(\frac{1}{\varepsilon}\log^2(\frac{1}{\varepsilon}))} = 2^{\widetilde{O}(\frac{1}{\varepsilon})}.$$

Integer Programming Formulation

Observation 2:

The vector $(x_k)_{k \in K}$ belongs to the system

$$\left.\begin{aligned} \sum_{k \in K} x_k &= m \\ \sum_{k \in K} k_i x_k &= n_i \quad \text{for all } \pi_i \in \Pi \\ x &\in \mathbb{Z}_{\geq 0}^K \end{aligned}\right\}$$

\# of constraints $= \widetilde{O}(\frac{1}{\varepsilon})$

\# variables $= 2^{\widetilde{O}(\frac{1}{\varepsilon})}$

Solving the ILP, first Approach:

Method [Alon et al. ’98] and [Hochbaum & Shmoys ’97] uses

Theorem [Kannan ’87 / Lenstra ’83]

An integer program with N variables can be solved in time $2^{\widetilde{O}(N)}\, s$ (where s is the length of the input).

In our case $N = |K| = 2^{\widetilde{O}(\frac{1}{\varepsilon})}$ and thus the running time is

$$2^{\widetilde{O}(N)} \log(n) = 2^{2^{\widetilde{O}(\frac{1}{\varepsilon})}} \log(n) \leftarrow \text{doubly exponential!}$$

Main Idea: Try to reduce the number of variables.

Solving the ILP, second Approach:

Guess the support [Jansen '10]

Theorem [Eisenbrand & Shmonin '06]

There is an optimum sol. x^* for $\{c^t x : Ax = b, x \geq 0, x \text{ integer}\}$ s.t. $|\text{support}(x^*)| \leq O(M(\log(M \cdot \Delta))$ where

- M = number of constraints,
- Δ = largest coefficient in A, c.

In our case:

- $M = |\Pi| = \widetilde{O}(\frac{1}{\varepsilon})$, and $\Delta = \frac{1}{\varepsilon}$
- $|\text{support}(x^*)| \leq \widetilde{O}(\frac{1}{\varepsilon})$

Solving the ILP, second Approach:

Guess the support [Jansen '10]

Algorithm:

1. Try each possible support: there are $\widetilde{O}(\frac{1}{\varepsilon}) \cdot \binom{|K|}{\widetilde{O}(\frac{1}{\varepsilon})} = 2^{\widetilde{O}(\frac{1}{\varepsilon^2})}$ many.
2. Solve ILP restricted to guessed variables with Kannan's algorithm (running time $2^{\widetilde{O}(\frac{1}{\varepsilon})} \log(n)$)
3. Total running time: $2^{\widetilde{O}(\frac{1}{\varepsilon^2})} \log(n)$.

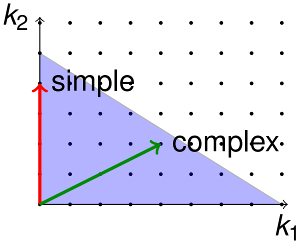
Solving the ILP, third Approach:

Understanding the Optimum

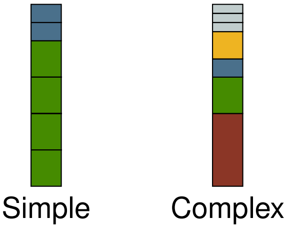
Definition

A configuration k is *complex* if it contains more than $\log(T+1)$ different sizes; o.w. it is *simple*.

Example $(\log(T+1)=1)$



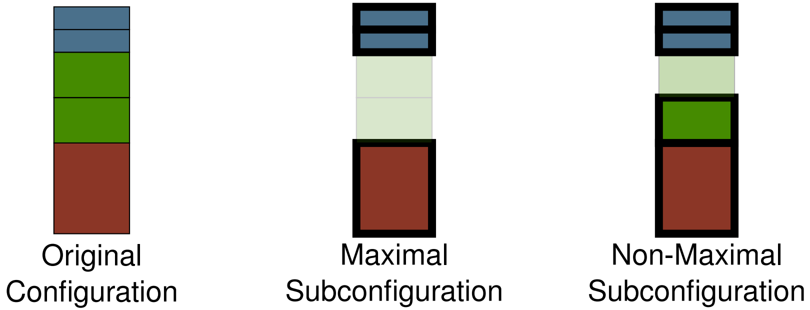
Example $(\log(T+1)=3)$



Solving the ILP, third Approach:

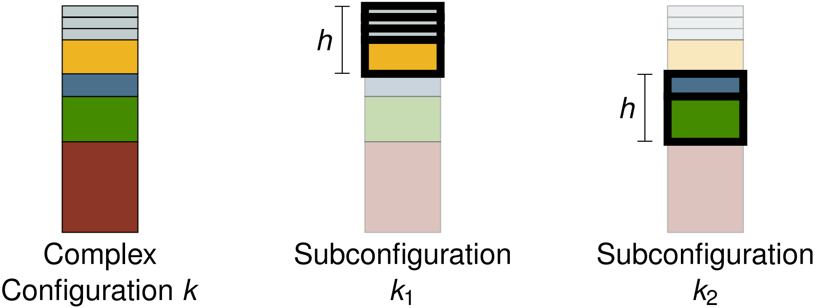
Understanding the Optimum

A “subconfiguration” $k' \leq k$ of configuration k is called *maximal* if it contains all possible jobs of each taken size.



Lemma

Every complex conf. $k \in K$ contains two maximal disjoint subconfigurations k_1, k_2 s.t. the total size of k_1 and k_2 coincide.



Lemma

Every complex conf. $k \in K$ *contains two maximal disjoint subconfigurations* k_1, k_2 *s.t.* $\pi \cdot k_1 = \pi \cdot k_2$.

Proof.

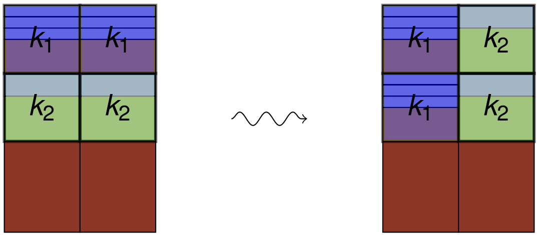
- Let $C > \log(T + 1)$ be the number of sizes (colors) in k.
- Number of maximal subconfigurations $= 2^C > T + 1$.
- Total size of each configuration is in $\{0, 1, 2, \ldots, T\}$.
- Pigeonhole principle $\Rightarrow$ there are two maximal subconfigurations of same total size.

$\square$

Solving the ILP, third Approach:

Lemma (Sparsification Lemma (informal))

If a complex configuration is taken twice in a solution, then we can replace it by two other "less complex" configurations.

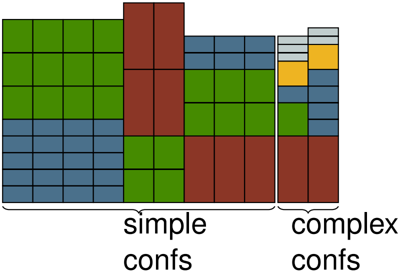


Solving the ILP, third Approach:

Theorem (Thin solutions)

If the ILP is feasible, then there is a solution x^* such that:

- *At most $\widetilde{O}(\frac{1}{\varepsilon})$ machines get complex configurations.*
- *Each complex configuration is used at most once.*
- $|support(x^*)| \leq O(|\Pi| \log(|\Pi| T)) = \widetilde{O}(\frac{1}{\varepsilon})$.



Lemma

The number of simple configurations in K is $2^{O(\log^2(\frac{1}{\varepsilon}))} = 2^{\widetilde{O}(1)}$.

Proof.

Let $D = \log(T+1)$ and $T = 1/\varepsilon^2$.

$$\begin{aligned}
\#\text{ simple conf} &\leq \sum_{i=0}^{D} \binom{|\Pi|}{i} \times (T+1)^i \\
&\leq (D+1)|\Pi|^D \times (T+1)^D \\
&\leq (\frac{1}{\varepsilon}\log(\frac{1}{\varepsilon}))^{O(\log(\frac{1}{\varepsilon}))} \\
&\leq 2^{O(\log^2(\frac{1}{\varepsilon}))} \leq 2^{\widetilde{O}(1)}.
\end{aligned}$$

□

Solving the ILP, third Approach:

Algorithm

Part 1: Complex Configurations.

1. Guess jobs assigned to complex configurations and number of complex machines.
2. Solve that subinstance optimally with a dynamic program.

Solving the ILP: Third Approach

Algorithm

Part 2: Remaining Instance.

1. Guess the (simple!) configurations in support:

$$\text{\# possibilities} \leq \binom{2^{\widetilde{O}(1)}}{\widetilde{O}(\frac{1}{\varepsilon})} = 2^{\widetilde{O}(\frac{1}{\varepsilon})}$$

2. For each possibility solve the ILP restricted to those variables with Kannan's algorithm.

Total running time: $2^{\widetilde{O}(\frac{1}{\varepsilon})} \log(n)$

Main Result:

Algorithm

Theorem [Jansen, Klein, Verschae '16]

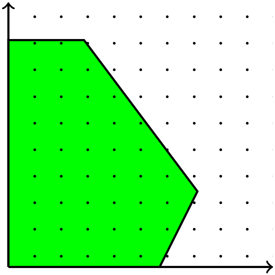
The minimum makespan problem on identical machines admits an EPTAS with running time

$$2^{O(\frac{1}{\varepsilon}\log^4(\frac{1}{\varepsilon}))} + O(n\log n) = 2^{\widetilde{O}(\frac{1}{\varepsilon})} + O(n\log n).$$

New Techniques via ILPs

$$\max c^t x$$
$$Ax = b$$
$$x \in \mathbb{Z}^n_{\geq 0}$$

where $A \in \mathbb{Z}^{M \times N}$, $b \in \mathbb{Z}^M$, $c \in \mathbb{Z}^N$.



Considered case

M (#constraints) is a constant, entries of A are small ($\leq \Delta$).

Pseudo-Polynomial Algorithms for ILPs

Known Algorithms

There is an algorithm for ILPs with running time:

- $(M(\Delta + \|b\|_\infty))^{O(M^2)}$ [Papadimitrou '81]
- $N \cdot O(M\Delta)^{2M} \cdot \|b\|_\infty^2$. [Eisenbrand & Weismantel '18]

Theorem [Jansen & Rohwedder ITCS 19]

ILPs can be solved in time $O(M\Delta)^{2M} \cdot (1 + \log(\|b\|_\infty))/\log(\Delta) + O(NM)$. Moreover, improving the exponent to $2M - \delta$ is equivalent to finding a truly subquadratic algorithm for $(\min, +)$-convolution.

Feasibility Problem

Theorem [Jansen & Rohwedder, ITCS 19]

Algorithm for feasibility with running time: $O(M\Delta)^{M} \cdot \log(\Delta) \cdot \log(\Delta + \|b\|_\infty) + O(NM)$. Improving exponent to $M - \delta$ would contradict the Strong Exponential Time Hypothesis (SETH).

Application $P||C_{max}$

Configuration ILP for large jobs

$$\begin{array}{ll} \sum_{k \in K} x_k = m & \\ \sum_{k \in K} k_i x_k = n_i & \forall \pi_i \in \Pi \\ x_k \in \mathbb{Z}_{\geq 0} & \forall k \in K \end{array}$$

has $M + 1 = O(\frac{1}{\epsilon}\log(\frac{1}{\epsilon}))$ constraints and $N = |K| = 2^{O(\frac{1}{\epsilon})}$ many variables. The value $\Delta = \max_{k,i} k_i \leq \frac{1}{\epsilon}$ and $\|b\|_\infty \leq n$.

New result: Including preprocessing $O(n + \frac{1}{\epsilon}\log(\frac{1}{\epsilon}))$, we get:

$$2^{O(\frac{1}{\epsilon}\log^2(\frac{1}{\epsilon}))} + O(n).$$

Main Underlying Idea

Theorem [Steinitz]

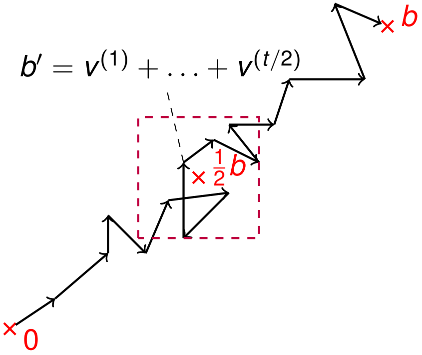
Let $\|\cdot\|$ be a norm in $\mathbb{R}^M$ and $v^{(1)}, \ldots, v^{(t)} \in \mathbb{R}^M$ with $\|v^{(i)}\| \leq 1$ $\forall i$ and $v^{(1)} + \cdots + v^{(t)} = 0$. Then there is a permutation $\pi \in S_t$ with $\|\sum_{i=1}^{j} v^{(\pi(i))}\| \leq M$ for all $j = 1, \ldots, t$.

Corollary

Let $v^{(1)}, \ldots, v^{(t)}$ denote columns of matrix A with $\sum_{i=1}^{t} v^{(i)} = b$ and entries bounded by Δ. Then there exists a permutation $\pi \in S_t$ such that for all $j \in \{1, \ldots, t\}$

$$\left\| \sum_{i=1}^{j} v^{(\pi(i))} - j \cdot b/t \right\|_\infty \leq 2M\Delta.$$

Our First Approach



Let $v^{(1)} + \ldots + v^{(t)} = b$ be columns corresponding to an optimal solution of (IP).

Equivalent:

$v^{(1)} + \ldots + v^{(t/2)}$ is optimal for

$$\{\max c^t x, Ax = b', x \in \mathbb{Z}^N_{\geq 0}\}$$

and $v^{(t/2+1)} + \ldots + v^{(t)}$ is for

$$\{\max c^t x, Ax = b - b', x \in \mathbb{Z}^N_{\geq 0}\}.$$

If ordered via Steinitz Lemma, b' and $b - b'$ are not far from $\frac{1}{2}b$.

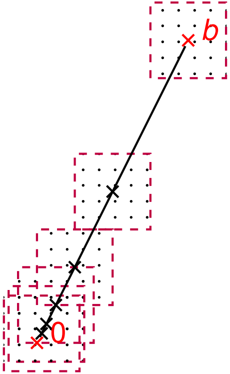
Dynamic Program

Solve for every $i = 0, 1, \ldots, \ell = O(M \log(M\Delta) + \log(\|b\|_\infty))$ and every $b' \in \mathbb{Z}^M$ with

$$\left\| b' - \frac{1}{2^{\ell-i}} b \right\|_\infty \leq 4M\Delta$$

the problem

$$\begin{aligned} \max\ & c^t x \\ Ax &= b' \\ \|x\|_1 &= 2^i \\ x &\in \mathbb{Z}^N_{\geq 0}. \end{aligned}$$



Original problem for $i = \ell$ and $b' = b$.

Second Approach via Discrepancy

Definition

For a matrix $A \in \mathbb{Z}^{M \times N}$ the **discrepancy** is

$$disc(A) = \min_{z \in \{0,1\}^N} \left\| A(z - (\frac{1}{2}, \ldots, \frac{1}{2})^T) \right\|_\infty .$$

The **hereditary discrepancy** of a matrix $A \in \mathbb{Z}^{M \times N}$ is

$$herdisc(A) = \max_{I \subseteq \{1,\ldots,N\}} disc(A_I)$$

where A_I denotes A restricted to the columns in I.

Results for Discrepancy

Theorem [Spencer '85]
For every matrix $A \in \mathbb{R}^{M \times N}$ with biggest absolute value of an entry bounded by Δ

$$herdisc(A) \leq 6\sqrt{M}\Delta.$$

Theorem [Beck, Fiala '81]
For every matrix $A \in \mathbb{R}^{M \times N}$, where the $\|.\|_1$ norm of any column of A is at most t it holds

$$herdisc(A) < t.$$

Discrepancy instead of Steinitz

Main Idea: Split the $\|.\|_1$ norm of a solution.

Lemma

Let $x \in \mathbb{Z}_{\geq 0}^N$. Then there is a vector $z \in \mathbb{Z}_{\geq 0}^N$ with $z_i \leq x_i$ for all $i = 1, \ldots, N$ and

$$\left\| A(z - \frac{x}{2}) \right\|_\infty \leq herdisc(A).$$

Furthermore, if $\|x\|_1 > 1$ then there is a vector z' as above with $1/6\|x\|_1 \leq \|z'\|_1 \leq 5/6\|x\|_1$

$$\left\| A(z' - \frac{x}{2}) \right\|_\infty \leq 2 \cdot herdisc(A).$$

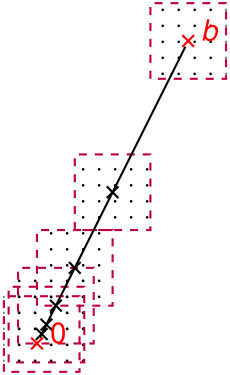
Modified Dynamic Program

Let x^* be an optimum solution with $\|x^*\|_1 \leq K$ and let H be an upper bound for *herdisc*(A).
Solve for every $i = 0, \ldots, \ell = \lceil \log_{6/5}(K) \rceil$ and every $b' \in \mathbb{Z}^M$ with

$$\left\| b' - \frac{1}{2^{\ell - i}} b \right\|_\infty \leq 4H$$

the problem

$$\max c^t x$$
$$Ax = b'$$
$$\|x\|_1 \leq (6/5)^i$$
$$x \in \mathbb{Z}^N_{\geq 0}.$$



Original problem for $i = \ell$ and $b' = b$.

Improved Results

Let H be an upper bound for *herdisc*(A).

Theorem [Jansen & Rohwedder, MOR 22]
ILPs can be solved in time $O(H)^{2M} \cdot M \cdot \log(M\Delta)/\log(\Delta) + LP$.

Theorem [Jansen & Rohwedder, MOR 22]
The feasibility problem for ILPs can be solved in time:
$O(H)^{M} \cdot \log(\Delta) \cdot M \cdot \log(M\Delta) + LP$.

Improved Algorithm for $P||C_{max}$

Using a specific rounding for large jobs and replacing configurations k with $\|k\|_1 > 2\log(1/\epsilon)$ we obtain a **modified Configuration ILP** with matrix A' for $P||C_{max}$ with

$$herdisc(A') \leq O(\log(1/\epsilon)).$$

This implies

Theorem [Berndt, Deppert, J., Rohwedder ALENEX 22]

The minimum makespan problem on identical machines admits an EPTAS with running time

$$2^{O(\frac{1}{\epsilon}\log(\frac{1}{\epsilon})\log\log(\frac{1}{\epsilon}))} + O(n).$$

Rounding scheme

1. let T be a makespan guess
2. discard *small* jobs with $p_j \leq \varepsilon T$ and *huge* jobs with $p_j \geq (1 - 2\varepsilon)T$
3. we know $p_j \in (\varepsilon T, (1 - 2\varepsilon)T)$
4. split this into $\log(1/\varepsilon)$ *growing intervals* $I_i = [2^i \varepsilon T, 2^{i+1} \varepsilon T)$
   for $\varepsilon = 1/6$, $T = 1$ we get the growing intervals
   $I_0 = [1/6, 1/3), I_1 = [1/3, 2/3)$
5. split these intervals into $1/\varepsilon$ many equally sized intervals
   the growing intervals $[1/6, 1/3)$ and $[1/3, 2/3)$ are split into smaller intervals with the following boundaries:

$$\frac{1}{6}, \ \frac{1}{6} + \frac{1}{36}, \ \frac{1}{6} + \frac{2}{36}, \ \frac{1}{6} + \frac{3}{36}, \ \frac{1}{6} + \frac{4}{36}, \ \frac{1}{6} + \frac{5}{36},$$
$$\frac{1}{3}, \ \frac{1}{3} + \frac{1}{18}, \ \frac{1}{3} + \frac{2}{18}, \ \frac{1}{3} + \frac{3}{18}, \ \frac{1}{3} + \frac{4}{18}, \ \frac{1}{3} + \frac{5}{18}.$$

6. round remaining job processing times to the next lower boundary
7. two boundaries (of equal parity) of a growing interval sum up to a boundary in the next growing interval, e.g.
   $(1/6 + 1/36) + (1/6 + 3/36) = 1/3 + 2/18$

Simplification of the Configuration ILP

$$\sum_{k \in \mathcal{K}} x_k = m$$

$$\sum_{k \in \mathcal{K}} x_k \cdot k_{i,j} = n_{i,j} \quad \forall (i,j)$$

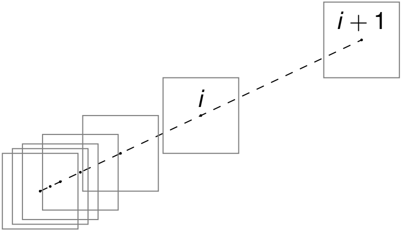
$$x_k \in \mathbb{Z}_{\geq 0} \quad \forall k \in \mathcal{K}$$

For each feasible configuration k with $\|k\|_1 > 2 \cdot \log(1/\varepsilon)$:

$\Rightarrow$ There is a growing interval i and two indices j, j' of the *same parity* with $k_{i,j} \geq 1$ and $k_{i,j'} \geq 1$ if $j \neq j'$ and $k_{i,j} \geq 2$ otherwise. *Decrement* each $k_{i,j}$ and $k_{i,j'}$ by one while *incrementing* $k_{i+1,\frac{j+j'}{2}}$ by one without breaking the feasibility of k or even altering its total load. This decreases $\|k\|_1$.

Convolution via FFT

- Think about the dynamic table entries as coefficients of multivariate polynomials
- Given the i-th dynamic table, we can compute the $(i+1)$-th dynamic table by computing the polynomial product of the i-th table with itself



- Can be done efficiently using the Fast Fourier Transformation (FFT)

Compute $O(\log(n))$ many FFTs on input of size $(\log(1/\varepsilon))^{O(1/\varepsilon\log(1/\varepsilon))}$ which yields total running time

$$2^{O(1/\varepsilon\log(1/\varepsilon)\log\log(1/\varepsilon))}\log(n)+O(n).$$

Implementation

- The code is available on: `github.com/made4this/BDJR`
- Implemented in C++ for $\varepsilon \approx 17.29\%$
- Parallelization with OpenMP in version 5.0
- FFT computations with FFTW3 in version 3.3.8
- 9000 experimental instances first used by Kedia in 1971
- Experiments computed in the HPC Linux Cluster of Kiel University using 16 cpu cores and 100GB of memory per instance.

Conclusion

Recent Work

- Improvements of our implementation for $P||C_{max}$.
- Improved EPTAS for $Q||C_{max}$ with support bound for integral variables in MILP formulation.
- Parameterized algorithm for $Q||C_{max}$ with d item sizes and maximum processing time p_{max}.

Conclusion

Open Questions

- Improved EPTAS for $P||C_{max}$ with running time $2^{O(1/\epsilon)} + O(n)$.
- Improve lower bound of EPTASs for $P||C_{max}$.