Maximum-Feasible Subsystems: Algorithms and Complexity

Rajiv Raman

DIMAP, University of Warwick

July 15, 2010

Example

Given a system of equations/inequalities satisfy the largest number of them that can be satisfied simultaneously.

$$\begin{aligned} x_1 + x_2 &= 1 \\ x_1 &= 1 \\ x_2 &= 1 \end{aligned}$$

At most *two* of the equations above are feasible.

Maximum Feasible Subsystem

Given a matrix $A \in \mathbb{Q}^{m \times n}$, and $u \in \mathbb{Q}^m$, the maximum feasible subsystem problem is to find the largest subsystem of

$$Ax \diamond u$$

that is feasible, where $\diamond \in \{\leq, <, =\}$.

Applications

- Operations Research: Modeling real-life problems using LPs.
- Computational Geometry: Densest Hemisphere
- Machine Learning
- Maximum Acyclic Subgraphs
- Pricing
- Several Others

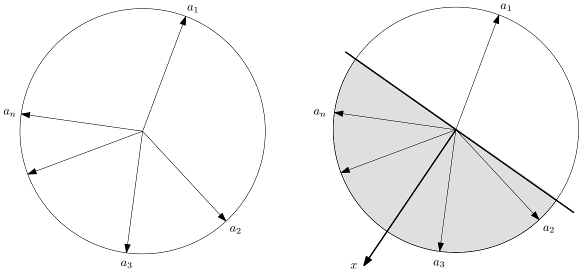
Application: Large LP models

- Modeling using large-scale LPs may lead to infeasible systems.
- Diagnosing infeasibility done by extracting a *minimal* infeasible system (Eg. CPLEX IIS[1] solver).
- This is the complementary problem, but much more difficult.

[1] Irreducible Infeasible System

Application: Densest Hemisphere

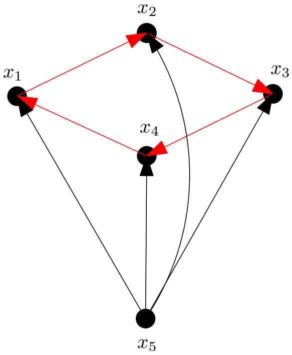
Given a set of points $\{a_1, \ldots, a_n\}$ on a sphere $\mathbb{S}^{n-1}$ in $\mathbb{R}^n$. The *Densest Hemisphere* problem is to find a halfspace that contains the *maximum number of points*.



$$\max\{a_i : a_i^T x \geq 0\}$$

Application:
Max Acyclic Subgraph

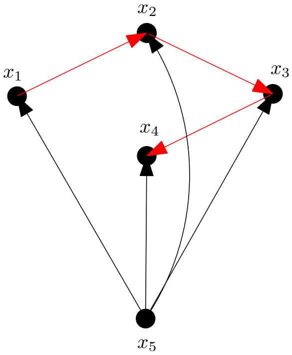
$D = (V, A)$ directed graph. Find $A' \subseteq A$ s.t. $D(V, A')$ is acyclic.



$$
\begin{aligned}
x_2 - x_1 &\geq 1 \\
x_3 - x_2 &\geq 1 \\
x_4 - x_3 &\geq 1 \\
x_1 - x_4 &\geq 1 \\
&\vdots
\end{aligned}
$$

Application:
Max Acyclic Subgraph

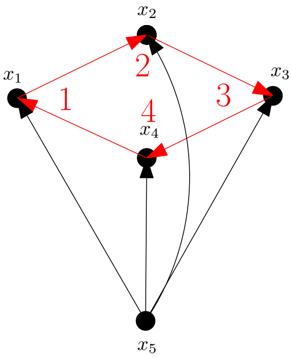
$D = (V, A)$ directed graph. Find $A' \subseteq A$ s.t. $D(V, A')$ is acyclic.



$$\begin{aligned} x_2 - x_1 &\geq 1 \\ x_3 - x_2 &\geq 1 \\ x_4 - x_3 &\geq 1 \\ x_1 - x_4 &\geq 1 \\ &\vdots \end{aligned}$$

Application: Max Acyclic Subgraph

$D = (V, A)$ directed graph. Find $A' \subseteq A$ s.t. $D(V, A')$ is acyclic.

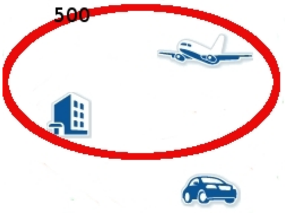


$$\begin{aligned} 2 - 1 &\geq 1 \\ 3 - 2 &\geq 1 \\ 4 - 3 &\geq 1 \\ 1 - 4 &\geq 1 \\ &\vdots \end{aligned}$$

Application: Pricing

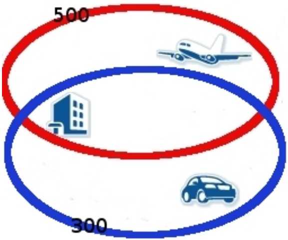
- $E = \{e_1, \ldots, e_m\}$ of items to sell.

Application: Pricing



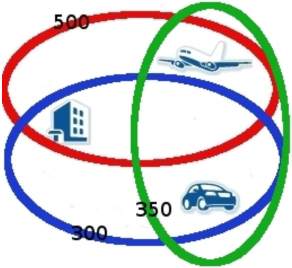
- $E = \{e_1, \ldots, e_m\}$ of items to sell.
- Buyer interested in a subset.
- Buyer i has budget B_i.

Application: Pricing



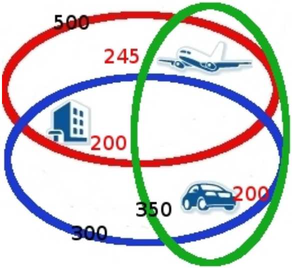
- A collection $E = \{e_1, \ldots, e_m\}$ of items to sell.
- Buyers interested in subsets of items.
- Each buyer i has a budget B_i.

Application: Pricing



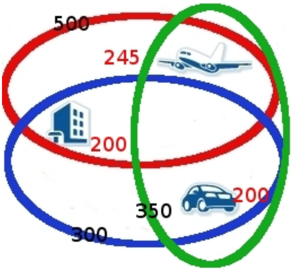
- A collection $E = \{e_1, \ldots, e_m\}$ of items to sell.
- Buyers interested in subsets of items.
- Each buyer i has a budget B_i.

Application: Pricing



- A collection $E = \{e_1, \ldots, e_m\}$ of items to sell.
- Buyers interested in subsets of items.
- Each buyer i has a budget B_i.
- Buyer buys if the total price is within her budget.

Application: Pricing



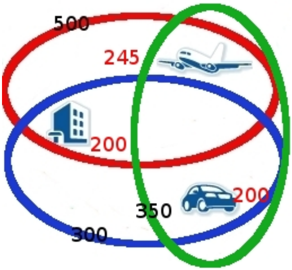
$200 + 245 \quad < \quad 500$ Buys

$200 + 200 \quad > \quad 300$ Does not Buy

$200 + 245 \quad > \quad 350$ Does not buy

Total Profit : $200 + 245 = 445.$

Application: Pricing



$200 + 245 \quad < \quad 500$ Buys

$200 + 200 \quad > \quad 300$ Does not Buy

$200 + 245 \quad > \quad 350$ Does not buy

Total Profit : $200 + 245 = 445$.

Objective

Set prices to maximize profit.

Pricing

- Assume buyers are *single minded*, i.e., each buyer is interested in exactly *one* subset.
- Assume we have infinitely many copies of each item.
- Also considered:
  - Buyers have different valuations on different subsets.
  - Finite copies of each item.

Connection to MFS

Lemma

There exists a 2-approximate solution S to Pricing such that each buyer i in S spends at least $B_i/2$.

Proof.

Let OPT be the optimal solution. $OPT = OPT_< + OPT_\geq$. If Lemma is not true, we can *double* the price of each item, and $OPT_<$ is still feasible, leading to a contradiction. □

With a loss of a factor of 2, we can assume each buyer spends at least $1/2$ her budget.

Connection to MFS

Define a variable p_e = Price of item $e \in E$. Then, for each buyer $i \in I$, we can write:

$$B_i/2 \leq \sum_{e \in I_i} p_e \leq B_i \text{ of weight } B_i$$

A *Weighted Maximum Feasible Subsystem Problem:* Maximize the *weight* of the satisfied constraints.

Related Work

- [Amaldi,Kann 95]
  - $Ax \diamond u$, $\diamond \in \{\leq, =\}$ NP-hard even if $A_{ij} \in \{-1, 1\}, \{-1, 0, 1\}$
  - 2-approximation for $Ax \leq b$.
  - Also studied variations where some inequalities must be satisfied in any feasible solution.
- $Ax = b$
  - [Feige, Reichman 07] $Ax = b$ is hard to approximate beyond $m^{1-\epsilon}$ for any $\epsilon > 0$.
  - [Guruswami,Raghavendra 07] $Ax = b$ with $|A_i| \leq 3$ is hard to approximate beyond $m^{1-\epsilon}$ for any $\epsilon > 0$.
- [Amaldi,Kann 98] Complexity of Minimizing unsatisfied constraints.

MFS with 0/1 Matrices

Given a matrix $A_{m\times n}$, $A_{ij} \in \{0,1\}$, and $l, u \in \mathbb{Q}^m$, and a weight function $w : \{1,\ldots,m\} \to \mathbb{Q}$, find a maximum weight feasible subsystem of

$$\begin{aligned} l_i \leq \quad A_i^T x \quad &\leq u_i \;\; (w_i) \\ x \quad &\geq 0 \end{aligned}$$

- Assume wlog. $\min\{l_i : l_i \neq 0\} = 1$.
- Let $L = \max\{l_1, \ldots, l_m\}$.

Bi-criteria Approximation

Let S be a feasible solution.

- $\alpha \geq 1$: Approximation factor.
- $\beta \geq 1$: Relaxation factor.

(α, β)-approximation

- $|S| \geq \frac{OPT}{\alpha}$, and
- For each $i \in S$, $l_i \leq A_i^T x \leq \beta u_i$.

i.e., We satisfy at least an α-fraction of the inequalities, while violating each inequality by a factor of β.

Our Results

0/1-Matrices	Bi-criteria approximation	Hardness Results.
Interval Matrices	Bi-criteria approximation	Hardness Results.

An $(O(\log nL), 1+\epsilon)$-approximation

$$\begin{aligned} l_1 &\leq A_1^T x \leq u_1 \\ l_2 &\leq A_2^T x \leq u_2 \\ &\vdots \\ l_m &\leq A_m^T x \leq u_m \end{aligned}$$

- Group into sets on $\frac{l_i}{|A_i|}$
- Solve each group separately.
- Return the best group.

An $(O(\log nL/\epsilon), 1+\epsilon)$-approximation

Constraint	$\frac{l_i}{\lvert A_i \rvert}$
$0 \le A_i^T x \le u_i$	$(0, (1+\epsilon)]$
$l_i \le A_i^T x \le u_i$	$((1+\epsilon), (1+\epsilon)^2]$
	$\vdots$
	$((1+\epsilon)^{i-1}, (1+\epsilon)^i]$
$l_i \le A_i^T x \le u_i$	$((1+\epsilon)^{h-1}(1+\epsilon)^h]$

- $L_{min} = \min\{\frac{l_i}{|A_i|}\}$ $(\ge 1/n)$
- $L_{max} = \max\{\frac{l_i}{|A_i|}\}$ $(\le L)$
- $h = \lceil \log_{1+\epsilon} \frac{L_{max}}{L_{min}} \rceil \le \lceil \log_{1+\epsilon} nL \rceil$
- Set $x = (1+\epsilon)^i$.
- Satisfies G_i, with $\beta = (1+\epsilon)$.
- Hence $\alpha \le h+1$

Hardness of Approximation

But this algorithm is almost the best possible.

Theorem

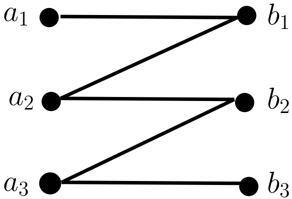
Unless $NP \subseteq DTIME(2^{n^{\epsilon}})$ for some $\epsilon > 0$, it is impossible to obtain a better than $O(\alpha, \beta)$-approximation, where $\alpha \cdot \beta = O(\log^{\mu} n)$, for some $\mu > 0$.

Hence, obtaining a better than $\left(O(\log^{\mu} n), O(1)\right)$-approximation algorithm is hard.

Dependence on L

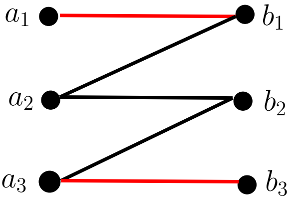
- $(\alpha, \beta) = (\log nL, 1 + \epsilon)$
- If L is poly(m, n), polylog approximation.
- What if $L = \Omega(2^{poly(m,n)})$
- Then MFS is hard to approximate beyond $\left(\left(\frac{L}{\log\log L}\right)^{1/3-\epsilon}, O(1)\right)$ for any $\epsilon > 0$, unless NP=ZPP.

Induced Matching
Bipartite Graphs



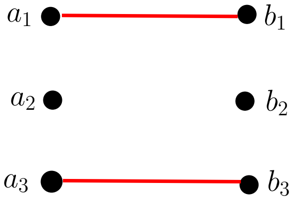
- $G = (A, B, E)$ bipartite graph.
- $M \subseteq E$ is an induced matching if
- $G(M)$ is a matching.

Induced Matching
Bipartite Graphs



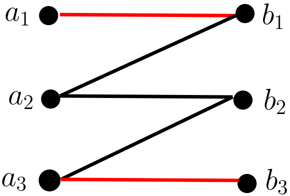
- $G = (A, B, E)$ bipartite graph.
- $M \subseteq E$ is an induced matching if
- $G(M)$ is a matching.

Induced Matching
Bipartite Graphs



- $G = (A, B, E)$ bipartite graph.
- $M \subseteq E$ is an induced matching if
- $G(M)$ is a matching.

Induced Matching
Bipartite Graphs

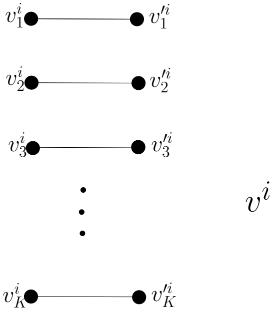


- $G = (A, B, E)$ bipartite graph.
- $M \subseteq E$ is an induced matching $\Rightarrow G(M)$ is a matching.
- $\max |M|$.

- $A_{ij} = 1 \Leftrightarrow \{i, j\} \in E$
- Largest identity matrix.

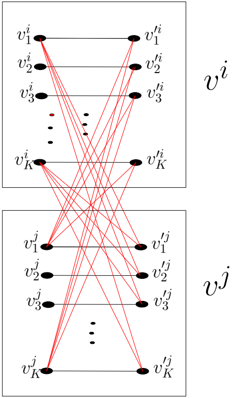
$$\begin{array}{c} \\ a_1 \\ a_2 \\ a_3 \end{array} \begin{array}{c} \begin{array}{ccc} b_1 & b_2 & b_3 \end{array} \\ \left[\begin{array}{ccc} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{array}\right] \end{array}$$

Reduction



- For a vertex v^i in $V(G)$.
- K copies: $\{v_1^i, \ldots, v_K^i\}$.
- K copies: $\{v_1'^i, \ldots, v_K'^i\}$
- Edge $v_j^i, v_j'^i$

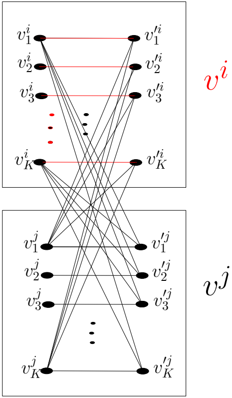
Reduction



- For $\{v^i, v^j\} \in E(G)$.
- All edges between gadgets.

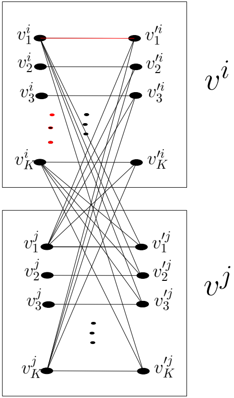
This completes the reduction.

Hardness of Approximation



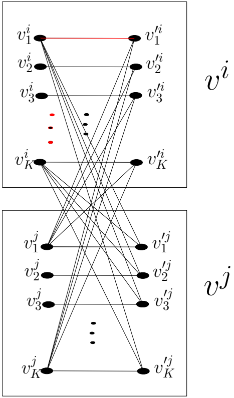
- Independent set IS of G, then
- $|MIDM| \geq K \cdot |IS|$

Hardness of Approximation



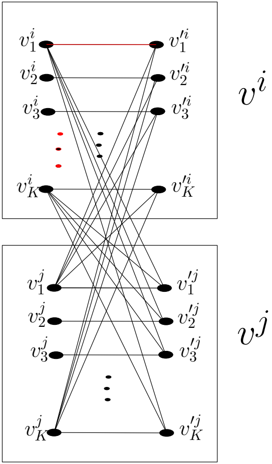
- (from prev) $|MIDM|/K \geq |IS|$
- $|IS| \geq |MIDM|/K - 1$.

Hardness of Approximation



- MIS hard to approximate beyond $\Omega(n^{1-\epsilon})$ for any $\epsilon > 0$ unless NP=ZPP.
- MIMP hard to approximate beyond $\Omega(n^{1/3-\epsilon})$. for any $\epsilon > 0$ unless $NP = ZPP$.

Hardness of Approximation



Theorem

MIM is hard to approximate beyond $n^{1/3-\epsilon}$ for any $\epsilon > 0$ unless NP=ZPP.

Hardness of Approximation

Using a similar reduction as in the Maximum Induced Matching on Bipartite Graphs, we can show

Hardness of Approximation

Using a similar reduction as in the Maximum Induced Matching on Bipartite Graphs, we can show

Theorem

MFS with 0/1 matrices is hard to approximate beyond $\left(\left(\frac{\log L}{\log\log L}\right)^{1/3-\epsilon}, O(1)\right)$ unless NP=ZPP.

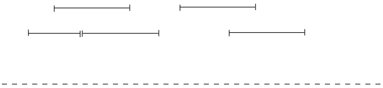
General 0/1 Matrices

Approximation (α, β)	Running Time	Hardness (α, β)
$(O(\log \frac{nL}{\epsilon}), 1+\epsilon)$	$poly(m, n, \log L, \frac{1}{\epsilon})$	$(O(\log^{\mu} n), O(1))$
		$(O((\frac{\log L}{\log \log L})^{\frac{1}{3}-\epsilon}), O(1))$

Here $L = \max\{l_1, \ldots, l_m\}$, $\min\{l_i, l_i \neq 0\} = 1$

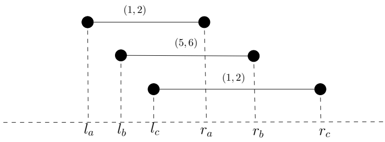
Interval Matrix

A matrix A is an interval matrix, if it is the *clique-vertex* incidence matrix of an *interval graph*. In other words, the matrix has the *consecutive one's property* in the rows.



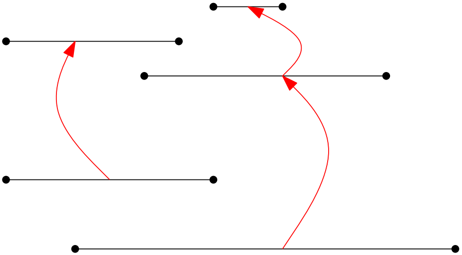
$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Interval Graph



- Intervals with total order on end-points.
- Each interval has lower/upper bounds on lengths.
- Draw maximum number of intervals satisfying order of end-points, and length constraints.

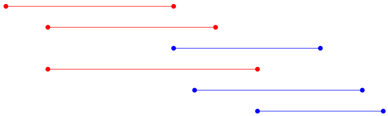
Interval Matrices



$(\sqrt{m}, 1)$-approx

- Define a partial order $\mathcal{P}$ by containment.
- Claim: We can partition $\mathcal{P}$ into partial orders $P_1, \ldots, P_k$ s.t. each P_i is either a chain or an anti-chain, and $k \leq 2 \cdot \sqrt{|\mathcal{P}|}$.

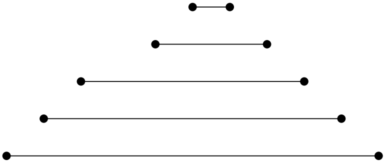
Interval Matrices



$(\sqrt{m}, 1)$-approx

- Each anti-chain can be partitioned into at most 2 sets V_1 and V_2 such that each V_i is a disjoint union of *staircases*.

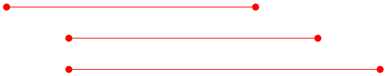
Interval Matrices



$(\sqrt{m}, 1)$-approximation

1. Find the best *tower* with each possible interval as base by dynamic programming.
2. Find the best staircase between two intervals for each pair.
3. Find the best set of disjoint staircases by maximum independent set of interval graphs.
4. Return the best of the two.

Interval Matrices



$(\sqrt{m}, 1)$-approximation

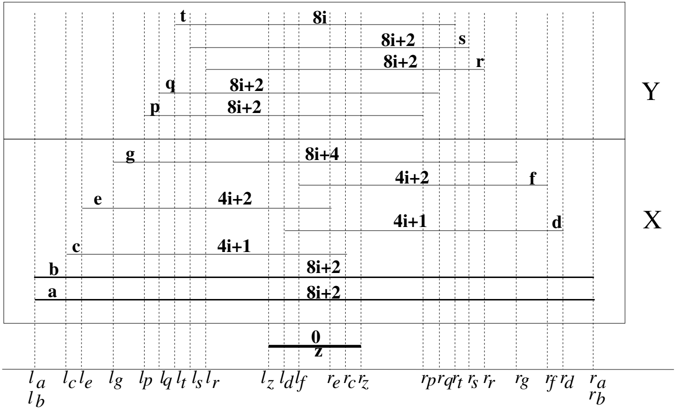
1. Find the best *tower* with each possible interval as base by dynamic programming.
2. Find the best staircase between two intervals for each pair.
3. Find the best set of disjoint staircases by maximum independent set of interval graphs.
4. Return the best of the two.

Interval Matrices

Approximation (α, β)	Running Time	Hardness
$(1+\epsilon, 1+\epsilon)$ [Grandoni,Rothvoss '10]	$poly(m, n, \frac{1}{\epsilon})$	NP-hard
$(\sqrt{m}, 1)$	$poly(m, n)$	APX-hard

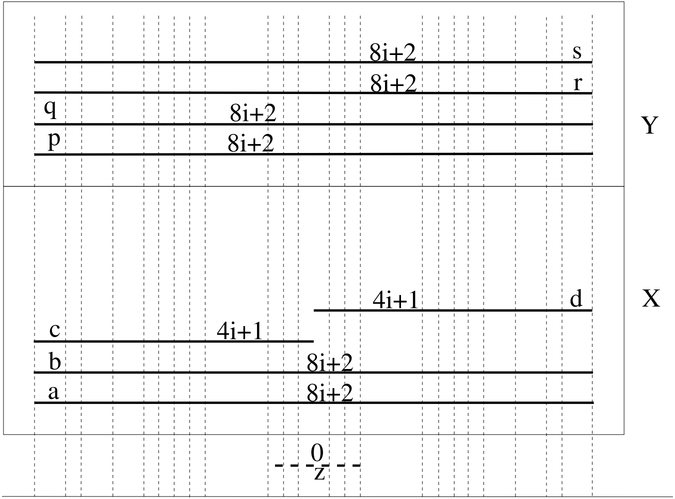
APX-hardness

Reduction from MAX-2-SAT.



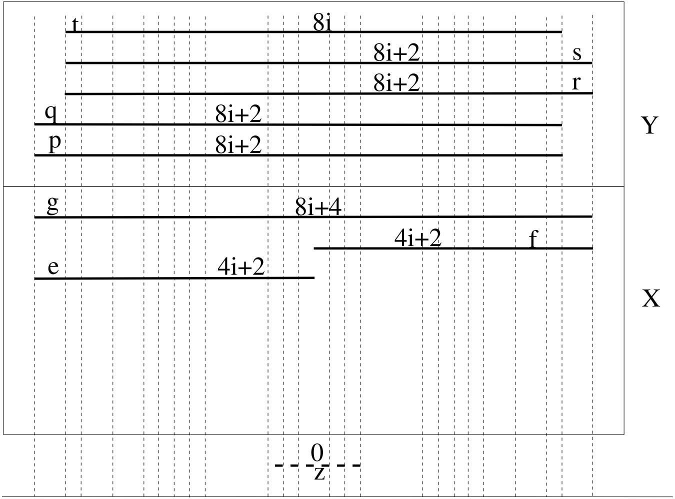
- Gadget for variable x_i.

APX-hardness



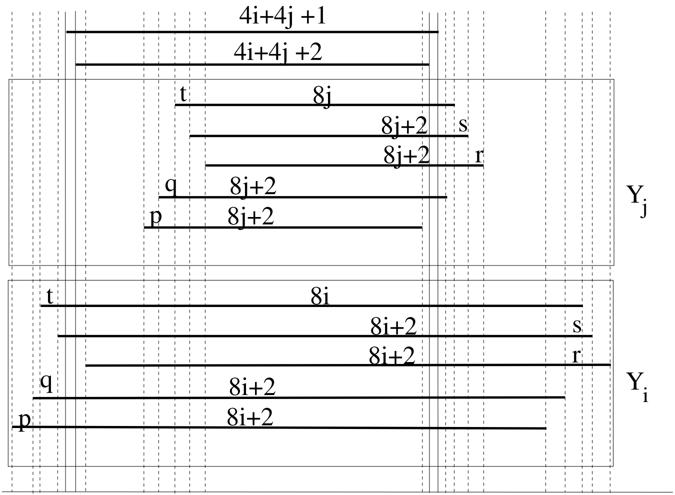
- TRUE configuration.

APX-hardness



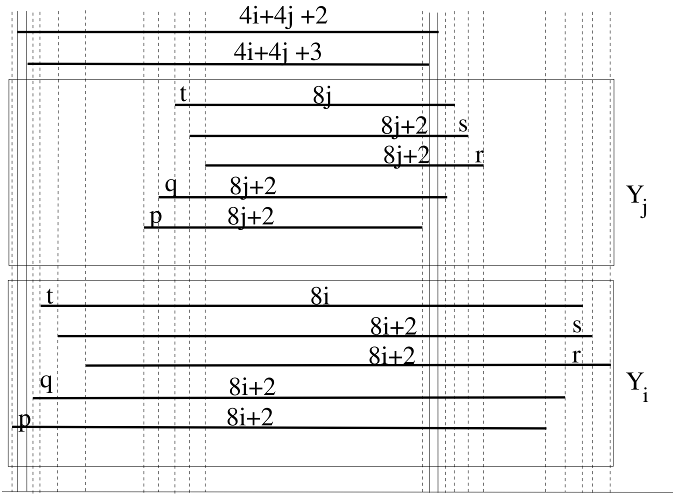
- FALSE configuration of gadget for x_i

APX-hardness



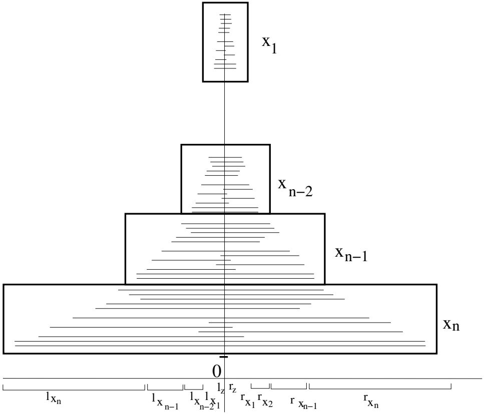
- Gadget for clause $(x_i \vee x_j)$

APX-hardness



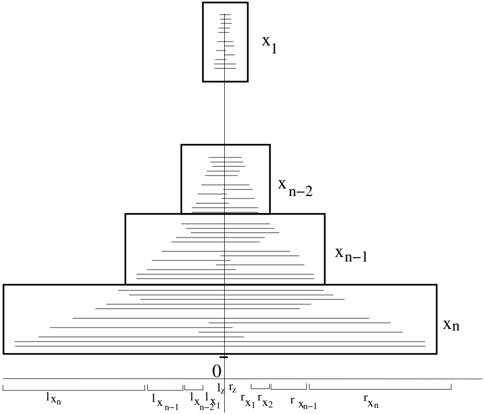
- Gadget for clause $(x_i \vee \overline{x_j})$.

APX-hardness



- The MFS instance.
- Each variable gadget is copied m_i times, where m_i is the number of clauses variable x_i appears in.
- A total of $O(m)$ intervals.

APX-hardness



- In any optimal solution, all variable gadgets in either TRUE or FALSE configuration.
- For each satisfied clause, exactly one interval is feasible iff it is satisfiable.

Open Questions

- MFS
  - Better approximation algorithms, or hardness for interval matrices.
  - Extend results on interval matrices to Totally Unimodular Matrices.
  - Non-Bi-criteria results.

References

Based on joint work with

- Stefan Canzar (CWI)
- Khaled Elbassioni, Amr Elmasry (Max-Planck-Institut, Saarbrucken)
- Saurabh Ray (EPFL)
- René Sitters (VU Amsterdam)

Thank you.