Scheduling: Open problems old and new

(an update on Schuurman-Woeginger'99)

Nikhil Bansal

(TU Eindhoven, Netherlands)

Very influential survey by Schuurman and Woeginger'99:

Ten open problems in approximation for scheduling

Dramatic progress in the field in last 20 years

Several of these problems completely solved

Several new rounding techniques: Iterated Rounding

New relaxations: Configuration LPs, Knapsack cover inequalities

Better understanding of SDPs

LP/SDP Hierarchies

Unique Games Conjecture, progress on hardness

…

Goal

Describe some of these developments/ideas

Mention new problems/remaining challenges

Mostly approximation, classic objectives

Disclaimer: Necessarily incomplete and biased

Will not cover: Online, game theory + scheduling, energy, …

Offline: Entire input known in advance. Find best solution.

(Often NP-Hard) Approximation ratio (Alg) = $\max_I$ Alg(I)/OPT(I)

Don't know OPT: compare with some kind of lower bound on OPT.

SW'99 structure

Problems 1,2,3,9 (precedence constraints)

Problems 4,8 (Unrelated machines)

Problem 10 (Flow Time)

Problems 5,6,7 (shop scheduling)

Precedence constraints

Open Prob. 1: P|prec| C_{max}

(Find $2 - \delta$ apx. or show $\frac{4}{3} + \delta$ hardness)

Setting: Jobs of size p_j (think p_j=1), m identical machines

Precedence (DAG) $i \prec j$ (j cannot start before i ends)

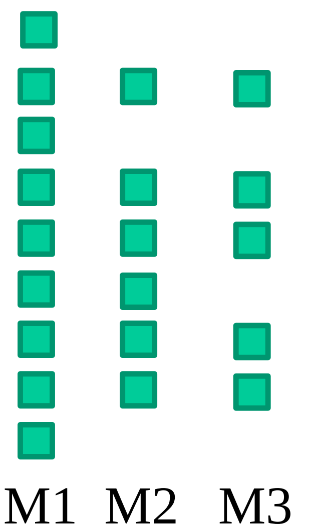
Minimize Makespan

Graham's list scheduling: 2-1/m apx

OPT $\geq \frac{\sum_j p_j}{m}$ (average load)

OPT $\geq$ length of longest chain

(Q: What about unrelated machines?)



Precedence constraints

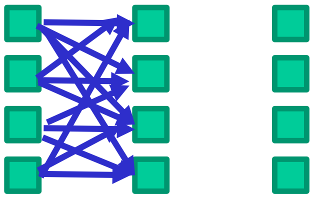
Open Prob. 1: P|prec| C_{max}

(Find $2-\delta$ apx. or show $\frac{4}{3}+\delta$ hardness)

Thm (Svensson'10): Cannot beat 2 assuming UGC variant.

Problem 1b. What if m = O(1), can you get $2-\delta$?

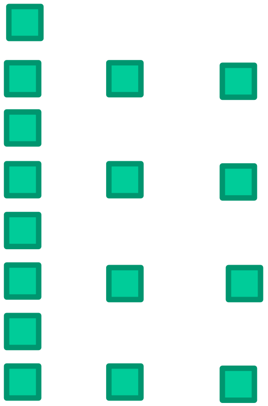
(running time can be any f(m))



Blocks of m+1 jobs

LP formulations, time indexed $x_{i,j,t}$ are weak

Claire Mathieu (MAPSP): Can hierarchies help?



Thm (Rothvoss'16): Can get $1+\epsilon$ apx.

Need $\exp\left((\log\log n)^2\right)$ rounds: k rounds, time = n^k.

quasi-poly << time << exponential

Weighted completion time

Open Prob 9. $1|\text{prec}|\sum_j w_j C_j$ Find a $2-\delta$ apx?

(weighted completion time, 1 machine)

Several different 2 approximations (since 80's)

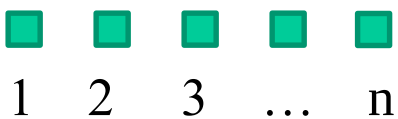
LPs, Gantt charts, vertex cover, graph decompositions, …

Thm [Bansal-Khot'09]: 2 best possible, assuming UGC variant

Key idea: Vertex Ordering Problem

Given n jobs of size 1. Groups $G_1, \ldots, G_k$ of jobs.

Order jobs. Group G_i finishes when all jobs scheduled



G1 = {1,3,4}
G2 = {2,4,5}
G3 = {1,3,5}
…

Question: What if groups are random (or all) with $\frac{1}{\epsilon}$ jobs.

Answer: Most groups finish around $n(1-\epsilon)$

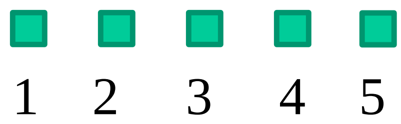
Question: What if groups are disjoint

Answer: Objective around n/2

Key idea: Vertex Ordering Problem

Given n jobs of size 1. Groups $G_1, \ldots, G_k$ of jobs.

Order jobs. Group G_i finishes when all jobs scheduled



G1 = {1,3,4}
G2 = {2,4,5}
G3 = {1,3,5}
…

Bansal-Khot'09: Assuming UGC, cannot distinguish in poly time

Good case: There exists an ordering, s.t. on average group finished around n/2 (structure)

Bad case: For every ordering, most groups finish close to end (random-like)

How to relate to $1|\text{prec}|\sum_j w_j C_j$?

Given n jobs of size 1.   Groups $G_1, \ldots, G_k$ of jobs.

Order jobs. Group G_i finishes when all jobs scheduled

1 2 3 4 5

G1 = {1,3,4}
G2 = {2,4,5}
G3 = {1,3,5}
…

Type 1:   Jobs 1,…,n :   Size = 1, Weight = 0

Type 2: A job for each group: Size = 0, Weight = 1

1 min UGC Hardness

Template: Create instance on hypercube x = $(x_1, \ldots, x_d)$

Solution = 0-1 labeling of hypercube

s.t. Any dictator f(x) = x_i (good solution)

"far" from dictator: formalized by low influence (bad soln.)

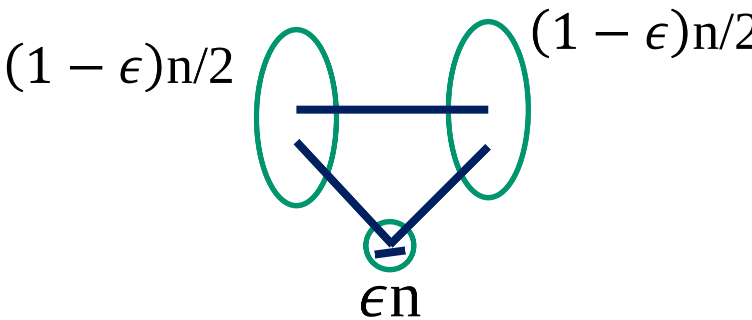
Groups = subcubes of hypercube on ϵd dimensions.

Dictator labeling = good value

Low influence, balanced labeling = bad (key technical part)

Long code test with 1 free bit:

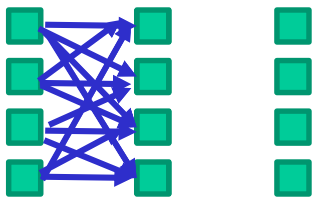
Structured independent set hardness



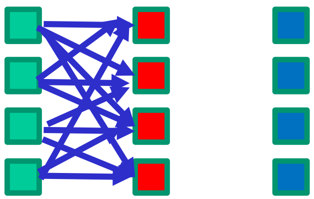
Back to P|prec| C_{max}

Svensson's insight to show hardness of 2:

Combine this + blocks of m+1 (m+ ϵm) instance

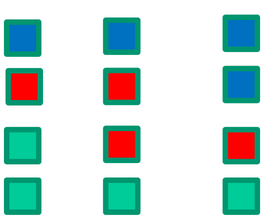


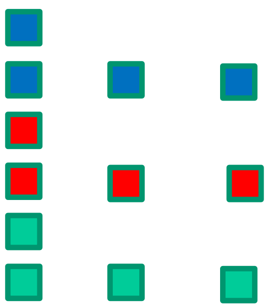
Blocks of m+1 jobs



Jobs Group-jobs

Good case





Bad case

Some technical work: Groups >> number of jobs

When m=O(1)

Rothvoss'16: Approximation scheme (super poly. time)

Key insight: List scheduling $\leq$ Load + max-chain

Done if chains are small ($\leq \epsilon$ OPT)

Use Sherali Adams Hierarchies to break chains.

1 min introduction to Hierachies

Traditional LP: $x_{i,j,t}$

Supposed to be 0-1, interpret fraction as probabilities.

New variables on k tuples: $x_{i,j,t,i',j',t'}$

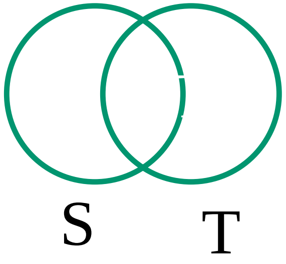
Interpretation: joint probability

Hierarchies min cx Ax =B (Base LP)

Lifted LP: Should satisfy consistency

Obvious relations $x_{i,j,t,i',j',t'} \leq x_{i,j,t}$

$$x_{i,j,t} = \sum_{i',t'} x_{i,j,t,i',j',t'}$$

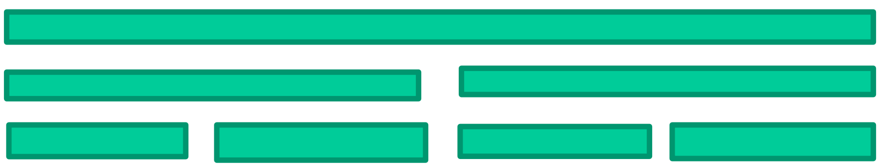


Can interpret Pr[A|B] = Pr[$A \cap B$]/Pr[B] $(x_{i,j,t,i',j',t'} / x_{i,j,t})$

Reducing chain length

Given a k-level solution, can condition on a variable (i.e. set x=0 or 1) and get valid k-1 level solution.

(lose a level at each conditioning: precious resource)



…

Time horizon $T \leq nm$

Level of job j = Span [support($x_{i,j,t}$)]

In each interval of size T' make chains << T'

If long chain, condition on middle job

(half the chain moves a level down)

+ Clever recursion

Open Prob 2: Q|prec| $C_{\max}$

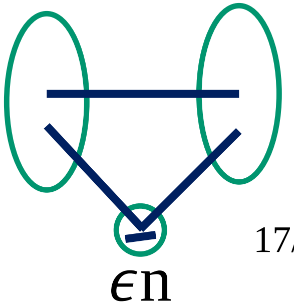
Q: Uniform machines model speed s_i, size = $\frac{p_j}{s_i}$

O(log n) Chudak Shmoys 90's (vs. 2 for identical machines)

Problem: Find better apx. or show non-constant hardness

(also in top 10 list of Shmoys-Williamson)

Probably hardness is the right answer



[Bazzi, Norouzi-Fard'15]: Non-constant hardness assuming a certain structured Hypergraph vertex cover is hard

Open Prob 3: P|prec, $c_{jk}|C_{\max}$

Open Prob. 3: P|prec|$C_{\max}$ with communication delays

Model: $j \prec k$ Lags c_{jk} if j and k run on different machines

(models transition time across machines/networks)

Not understood at all. Almost completely open.

Only very special cases ($p_j = 1$). Only apx hardness.

Finding any promising LP/ SDP relaxations would be a big step.

Shop Scheduling

Open problems 5,6,7

Huge area, many variants

Matching hardness in many cases

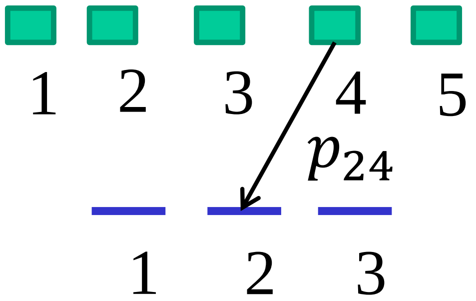
Mastrolilli, Svensson (job shops)

Bansal, Khot (open shops)

…

Unrelated machine setting

p_{ij}: Size of job j on machine i (unrelated: p_{ij} arbitrary)



n=5 Jobs

m=3 machines

Open prob. 8: R || $\sum_j w_j C_j$

Can you beat 3/2

[Schulz Skutella, Skutella, Chudak, Sethuraman Squillante: late 90's]

Open prob. 4: R || $C_{\max}$

Can you beat 2 [Lenstra, Shmoys, Tardos 87]

Total Weighted Completion Time

Min $\sum_j w_j C_j$ C_j: Completion time of j

Total completion time of these jobs
$w_1 p_1 + w_2(p_1 + p_2) + w_3(p_1 + p_2 + p_3)$

On any machine i, Smith rule: Decreasing order of $\frac{w_j}{p_{ij}}$

Only issue: Where to assign jobs

Objective: $\sum_i \sum_j w_j \left(\sum_{j' \preccurlyeq_i j} p_{ij'} \, x_{ij'} \, x_{ij} \right)$

Smith ordering $\prec_i$: $j' \prec_i j$ if $w_{j'}/p_{ij'} \geq w_j/p_{ij}$

Break ties arbitrarily to get total ordering

3/2: Convex programming
[Skutella, Sethuraman Squillante, Chudak late 90's]

Thm [Bansal, Srinivasan, Svensson'16]: 3/2 – 10^{-7} apx.

Convex Program: Integrality gap of 3/2
New SDP formulation

Independent Randomized Rounding cannot beat 3/2
New dependent rounding w/ strict negative correlation

Convex Programming

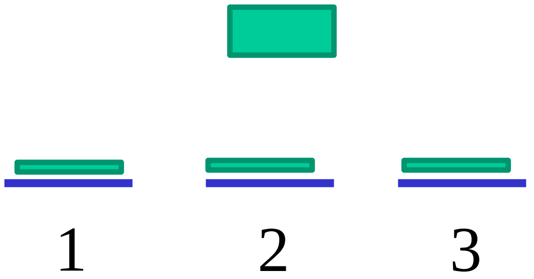
Objective (machine i): $\sum_j w_j \ (\sum_{j' \preccurlyeq_i j} p_{ij'} \ x_{ij'} x_{ij} \ )$

Convex Program: Min $\sum_i \ (Expression\ i)$

s.t. $\sum_i x_{ij} = 1$ for all j.

Bad example: 1 job of size 1, m machines

Convex Program: Sets x_{i1} = 1/m $\left(m \cdot \frac{1}{m^2} = \frac{1}{m}\right)$



Expression’ i: $= \frac{1}{2}\, p_{ij}^2 x_{ij} + \frac{1}{2} \left(\sum_j p_{ij} x_{ij}\right)^2$ [$x_{ij}^2 = x_{ij}$ valid]

(assume $w_j = p_{ij}$)

Fix: Reducing gap to 3/2

Expression' i: $= \frac{1}{2}\, p_{ij}^2 x_{ij} \; + \frac{1}{2}\, \left(\sum_j p_{ij} x_{ij}\right)^2$ $\quad [x_{ij}^2 = x_{ij}$ valid]

Still a gap of 2

Add constraint: $OPT \;\geq\; \sum_i \sum_j \, p_{ij}^2 x_{ij}$ (i.e. $\sum_i \sum_j w_{ij}\, p_{ij}\, x_{ij}$)

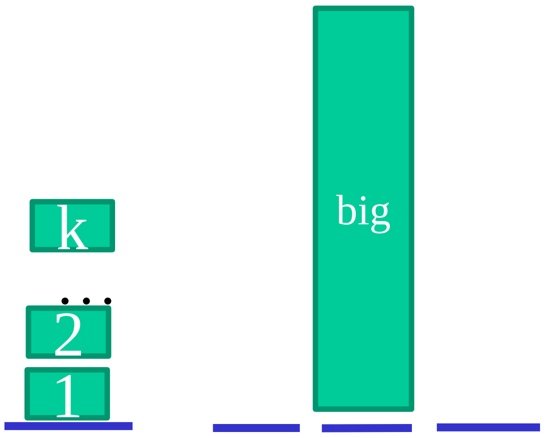
Cost $\geq$ L/2 + Q/2 Cost $\geq$ L

Somewhat adhoc mix of L and Q
Surprisingly, integrality gap improves to 3/2 (but not better)

Tight integrality gap example

k jobs: Size 1 each, only on machine 1

1 job: Size k^2 on any machine 2,…,k+1



Optimum: $\frac{k(k+1)}{2} + k^2 \approx \frac{3}{2}k^2$

Convex Program: $\left(OPT > \frac{L}{2} + \frac{Q}{2},\ \ OPT \geq L \right)$

Quadratic term (Q): $\approx \frac{k^2}{2} + \frac{1}{2}k^2$

Linear term (L) : $k + k^2$

New SDP

Write natural SDP (vectors v_{ij}, $x_{ij} = |v_{ij}|^2$)

Captures correlations and integrality more effectively

Add $v_{ij} \cdot v_{ij} = v_0 \cdot v_{ij}$ (like $x_{ij} = x_{ij}^2$)

($v_{ij} \cdot v_{ij}$' gives joint probability of j and j' on i)

Important consequence: Linear and quadratic terms combined more systematically

E.g. For any subset of jobs $S \subset J$

OPT $\geq$ L(S) + ½ $L(\overline{S})$ + ½ $Q(\overline{S})$

Previous OPT $\geq$ L(J) and OPT $\geq$ ½ L(J) + ½ Q(J)

The Rounding Issue

Given the x_{ij} (allocation probabilities). How to round?

Randomized rounding stuck at 3/2.

m identical jobs on m machines. x_{ij} = 1/m (split equally)

OPT = m

Pr[c jobs on a machine] := p_c ≈ 1/e (1/c!)

E[c(c+1)/2 p_c] = 3/2

What would be the right rounding here?

Want to reduce variance

Main Idea

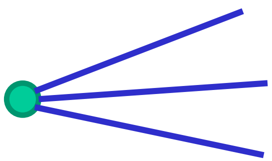
1) If few jobs, do matching type rounding
2) If many "similar" jobs, randomized rounding ok.

Basically, this works

Gandhi et al. (Randomized pipage) for assignment

Can round, so that get nice negative correlation

(e.g. $\Pr[x_{ij} x_{ij'}] \leq \Pr[x_{ij}] \Pr[x_{ij'}]$)



Our theorem: Strict negative correlation within "groups", and negative correlation across groups.

Question: Get a more respectable guarantee

With release times, 2 improved to 1.81 (Im, Li’16) (see talk)

Restricted assignment + same smith ratio? $\left(p_j \in \{w_j, \infty\}\right)$

Improve to 1.21

(Previous convex program still has gap 3/2)

Open 4: R|| $C_{\max}$ get $2-\delta$ apx or $1.5+\delta$ hardness

Landmark result of Lenstra, Shmoys, Tardos: OPT + $p_{\max}$

x_{ij} variables (Simple modern proof via iterated rounding)

Hard to beat even for parallel machines (m+1 unit jobs, m machines)

Configuration LP:

Variable $x_{i,S}$ for each “valid” job S with load ≤ OPT on machine i.

Constraints: $\sum_S x_{i,S} = 1 \quad \forall i$

$$\sum_i \sum_{S: j \in S} x_{i,S} = 1 \quad \forall j$$

Consider dual, give separation oracle (apply ellipsoid)

Lots of progress

Restricted assignment

Svensson'12: 1.94 estimation (configuration LP)

Jansen, Rohwedder'17: $11/6 \approx. 1.83$

More recently 1.83 in Quasi-polynomial time

Graph balancing (Ebenlendr, Krcal, Sgall'08)

Alas, Configuration LP has gap 2 for unrelated case.

Any hope?

Few levels of Sherali Adams capture configuration LP

Possible direction

Related: Santa Claus problem [Bansal, Sviridenko'06]

Maximize the minimum load (aka max-min allocation)

Lenstra, Shmoys, Tardos gives OPT – $p_{\max}$ (could be infinite apx.)

Naïve LP: Gap 2-1/m (makepsan) gives gap m (Santa Claus).

Restricted assignment: Config. LP O(log log n) [BS'06]

O(1) non-constructive via recursive LLL [Feige'08],

Hypergraph matching [Asadpour, Feige, Saberi' 09]

Now polytime [Haeupler, Saha, Srinivasan'10], …,[Annamalai'16]

Belief: $2 - c/\alpha$ approx. for makespan iff α for Santa Claus

Santa Claus on unrelated machines

Configuration LP $\Omega\left(m^{1/2}\right)$integrality gap

[Asadpour, Saberi'07]: $\tilde{O}(m^{1/2})$ (max entropy matchings)

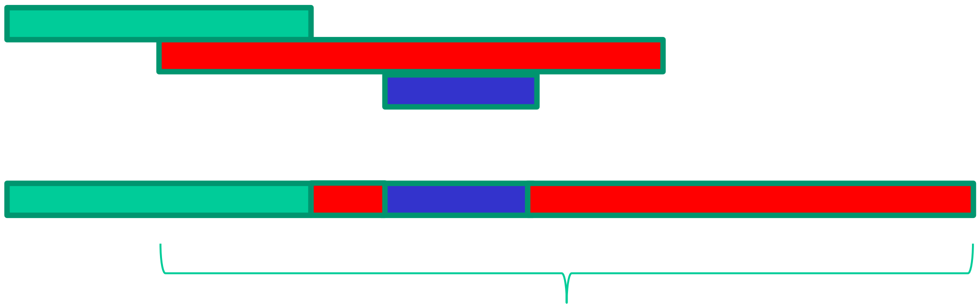
[Chakrabarty, Chuzhoy, Khanna'08]: $\tilde{O}(N^{\epsilon})$ apx. in $N^{1/\epsilon}$ time

(tour-de-force, amazing ideas, still use config LP)

polylog apx in quasi-poly time.

Perhaps easier (hard) goal: Get O(1) for Santa Claus.

Flow time related metrics



Flow time: $f_j = C_j - r_j$
(comp. time – release time)

Open Pr. 10: $P|p_j, r_j| \sum_j F_j$
$1|\, p_j, r_j \,|\, \sum_j w_j F_j$

Leonardi Raz'97 O(log n) + various follow ups
[SW'99] Problem: Can you get O(1) approx.?

$$P\left|p_j, r_j\right| \sum_j F_j$$

$\Omega(\log n)$ hardness [Garg-Kumar 06]

O(log n) apx for more general settings

Uniform machines, Restricted assignment, …

Time indexed LP, fractional flow time objective LP

(surprising, because flow time is quite sensitive)

Very nice use of single source unsplittable flow

Technique breaks down for unrelated machines

[Bansal, Kulkarni'15]: $O(\log^2 n)$ apx. via iterated rounding

Q: Improve it to O(log n) ?

Single source unsplittable flow

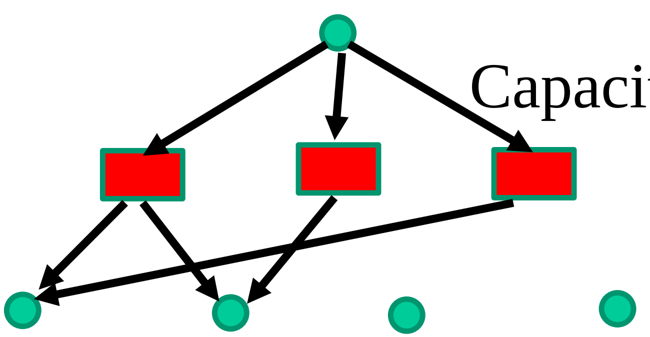
Given: Graph G=(V,E)   capacities c_e

Source s. Various demands $d_1, \ldots, d_k$. Feasible Flow $f_1, f_2, \ldots, f_k$

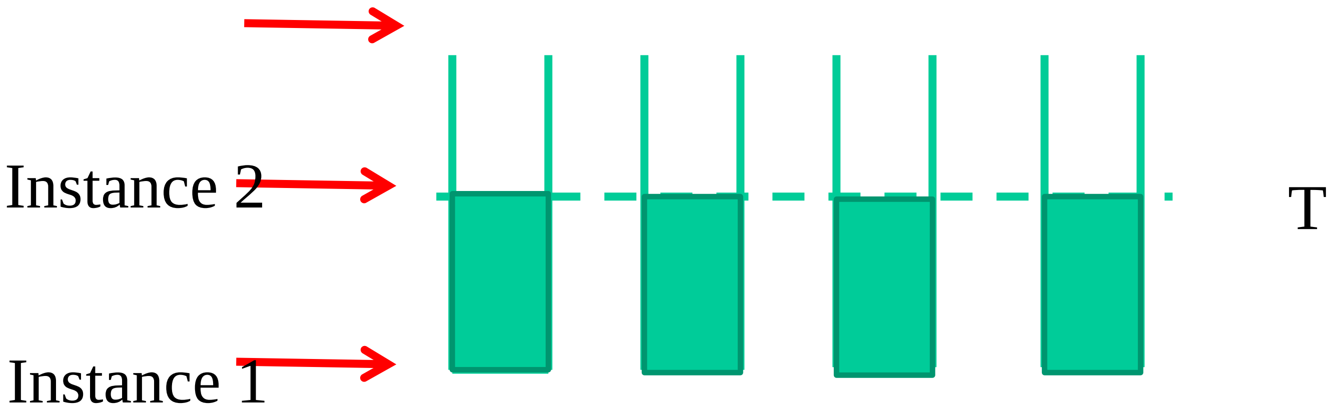
Goal: Make it unsplittable

Thm [Dinitz, Garg, Goemans'96] : Violate capacity by at most $d_{\max}$

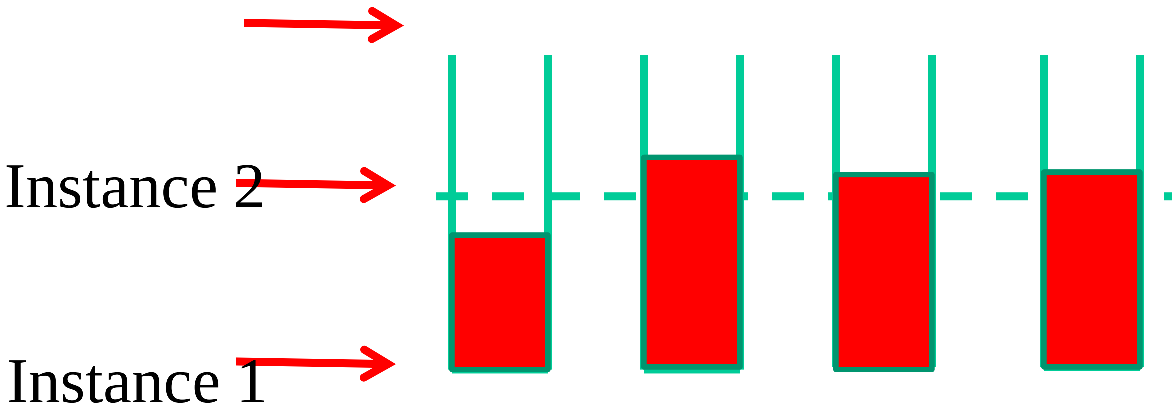
Easy Corollary: Makespan minimization for restricted assignment



Trouble: (Max) Flow time



If all release times = 0, Max flow time = Makespan



Error can build up

Previous Approaches

Time-indexed LP formulation

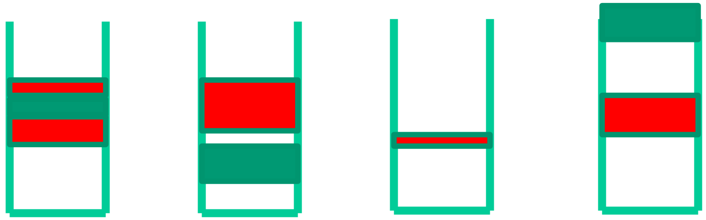
x_{ijt} : How much of job j scheduled on machine i at time t

All the constraints:

(No overload) $\sum_j x_{ijt} \leq 1$ for all (i,t)

(each job scheduled) $\sum_i ((\sum_t x_{ijt})/p_{ij}) \geq 1$ for all j

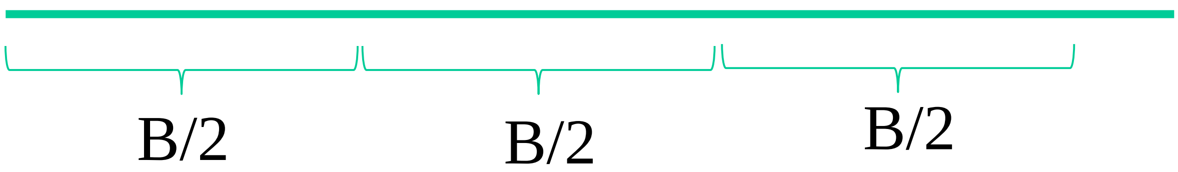
Some objective for "fractional" flow time.



(note: LP can put a job on many machines, and schedule the pieces in parallel)

Main new idea

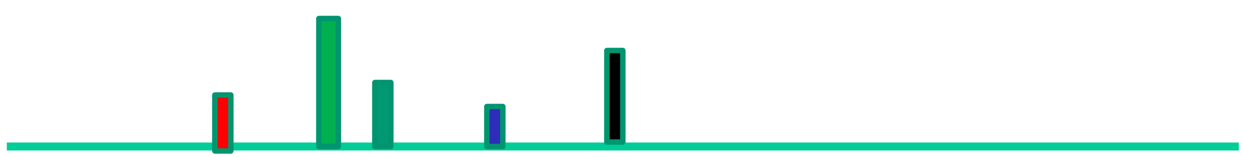
Reducing constraints



Replace $\sum_{jt} x_{ijt} \leq 1$ by "Load $\leq B/2$ for every interval"

Bounded overload: For any interval X of time (s,t)

Total work assigned to X $\leq$ (t-s) + B.



Iterated Rounding

LP $\quad \max cx$

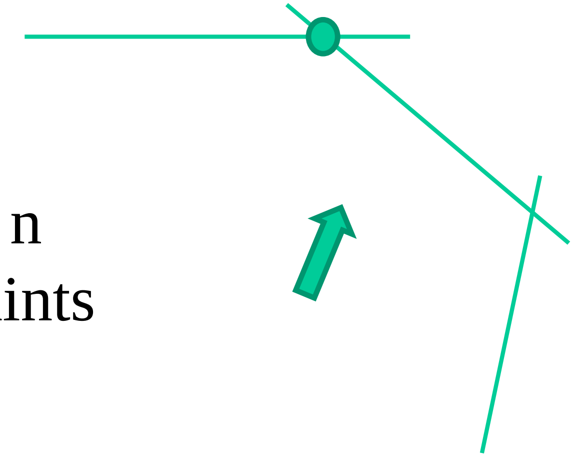
$Ax \leq b$

$0 \leq x \leq 1$

n variables

m non-trivial constraints

Obs: There is an optimum solution with $\geq$ n-m variables set to 0-1.



Vertex: determined by solution of n (tight) linearly independent constraints

(basic feasible solutions)

Idea

At least half of the jobs assigned integrally

Repeat for $O(\log n)$ iterations.

Argue that not get much worse in each iteration

Weighted Flow Time $1|\, p_j, r_j \,|\, \sum_j w_j F_j$

Can you get an O(1) apx (my favorite scheduling problem)

Time indexed LP very bad (fractional vs integral flow time)

polylog(n,P,W) online [Chekuri,Khanna,Zhu'01, Bansal, Dhandhere'04]

QPTAS [Chekuri, Khanna'02]

O(log log n) [Bansal, Pruhs'10]

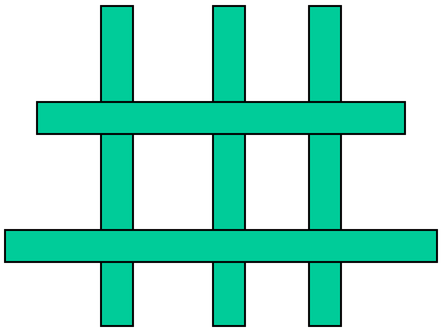
New relaxation, a knapsack cover problem for each time interval

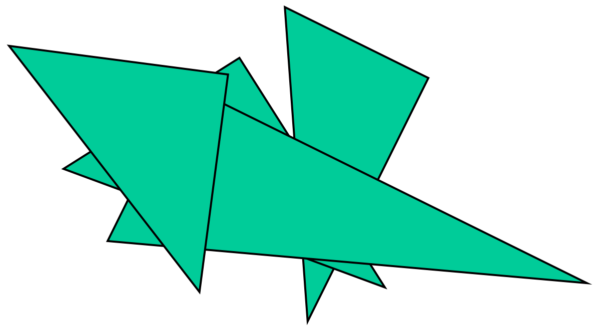
(one knapsack = exponentially many KC inequalities)

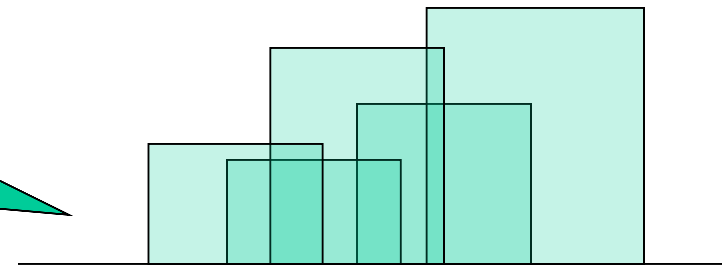
After various clean up steps: geometric set-cover with low union complexity objects

Weighted Geometric Set Cover

Union Complexity: Take k objects, look at their boundary (vertices,edges, holes). Scales as k h(k)







$\Omega(k^2)$

$O(k \log^* k)$
[Aronov, de Berg, Ezra, Sharir 11]

$O(k)$

Thm [Varadarajan'10, Chan-Grant-Konemann-Sharpe'12]: $O(\log (h(n)))$ apx.

Conjecture: The Knapsack Cover LP should be O(1) approx.

(Union complexity approach cannot work directly, need to use more structure)

[Im,Moseley]: Can get O(1) apx. For ℓ_p-norms for weighted flow

Some other Directions

Non-preemptive problems

$1||\sum_j F_j$ [Kellerer, Tautenham, Woeginger'97]: $n^{1/2}$ hardness

Need resource augmentation

Q: Get O(1) or polylog apx. with $1+\epsilon$ resource augmentation

[Bansal et al'07] O(1) apx, 12-speedup via a stronger LP.

But LP has a big gap if < 2 speedup.

Multi-dimensional problems

Job: $p_j = (p_{j1}, \ldots, p_{jd})$ resouces: CPU, memory, …

Unrelated p_{ij}

Vector Packing: Minimize number of bins

Thm [Bansal,Caprara, Sviridenko'06]: ln d + 1 (if d=O(1))

Round and approx. framework + Configuration LPs

Only 1.001 hardness (d=2)

Question: Get O(1) apx. when d is fixed.

$d^{1-\epsilon}$ hard (via coloring) when d part of input

Multidimensional Problems

m machines, makespan minimization

Thm: log d / log log d (even for unrelated case)

Proof: Iterative Lovasz Local Lemma

(Leighton, Srinivasan,…)

No non-constant hardness known

Questions!