The Geometry of Scheduling

Nikhil Bansal (TU Eindhoven)

Kirk Pruhs (U. Pittsburgh)

The problem

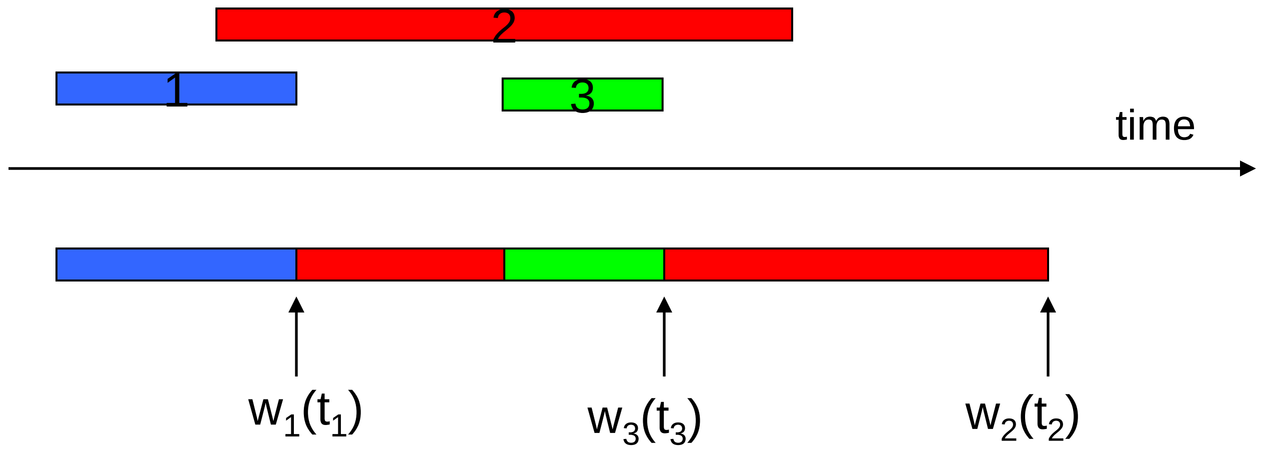
Input: Jobs 1,…n.

Job i has release time r_i, size p_i,

Non-decreasing weight function $w_i(t)$: cost if i finishes at t.

Goal: Find a schedule that minimizes the total cost.

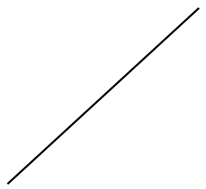
Allow preemption: A job can be interrupted.



Examples

1. Weighted Flow time

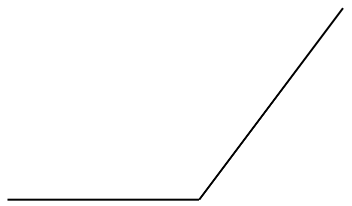
   $w_i(t) = w(i)\,(t - r_i)$ for $t \geq r_i$.



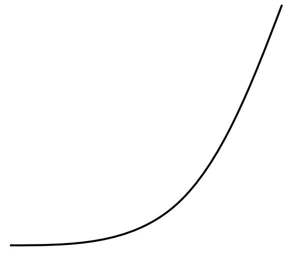
2. Weighted Tardiness

   $w_i(t) = 0$ if $t \leq d_i$

   $= w(i)\,(t - d_i)$ if $t > d_i$



3. Weighted flow time squared.



4. Hard deadlines

… (mixture of various things)

Previous Results

1. Weighted Flow Time: Extensive work

   $O(\log^2 nP)$ [Chekuri-Khanna-Zhu 02]

   $O(\log W)$, $O(\log nP)$ [Bansal-Dhamdhere 03]

2. Weighted Tardiness: n-1 apx. if all $r_i = 0$ [Cheng et al 05]

   (nothing for general release times)

   Various special cases studied.

3. Flow Time Squared: Only guarantees with resource augmentation [Bansal-Pruhs 03]

Main Result

Thm 1: An O(log log nP) approx. for general problem

n= number of jobs

P= maximum job size.

Previously, no sub-logarithmic guarantees known even for very special cases.
(Recent results improve to O(log log P)

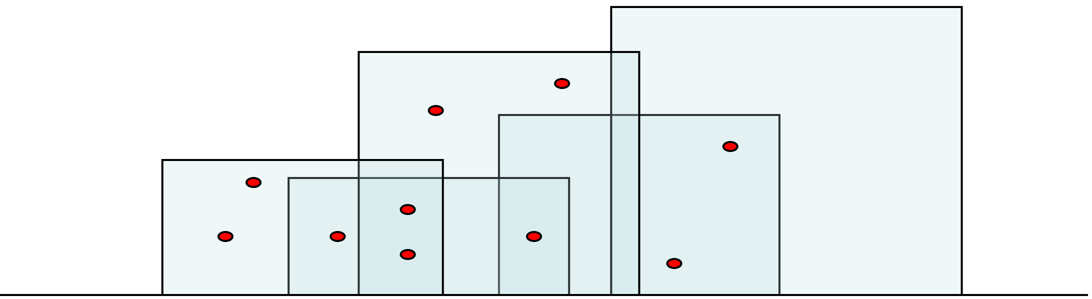
Thm 2: If all release times are 0, can get 16 approx.

Previous best for weighted tardiness was n-1
(Improved to $2+\epsilon$ [Cheung-Shmoys 11])

Main Idea

Reduce to a 2-d geometric covering problem (2RC)

(lose only factor 4 in reduction)



Scheduling is most naturally viewed as a packing problem.

Capacitated Version $\longrightarrow$

$$\text{Min} \sum_r w_r x_r$$
$$\sum_{r\,:\,p \in r} c_r x_r \geq d_p \quad \forall p$$
$$x_r \in \{0,1\}$$

Give O(log log nP) approx for 2RC.

2-approx if unit demands and capacities (regular set cover)

Outline

1. Reduction to 2RC

2. Reduce 2RC to other uncapacitated geometric problems (Knapsack Cover inequalities)

3. Use Geometric structure to beat O(log n)

Reduction to 2RC

An Alternate View

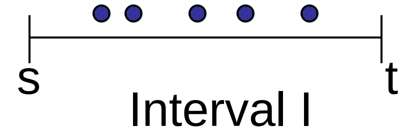
Find deadlines $d_1,\ldots,d_n$ s.t. schedule is feasible.

(feasible: each job can be completed by its deadline)

Total Cost = $\sum_j w_j(d_j)$

Consider interval I = [s,t]. Let J(I) = jobs with $r_j \in [s,t]$.

Call Size(J(I)) = total size of jobs in J(I)



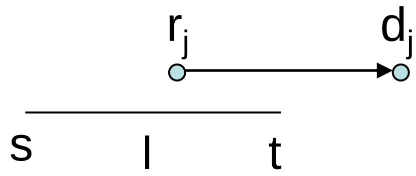
Necessary for feasibility: Total size of jobs in J(I) assigned deadline $d>t$ must be $\geq$ Size(J(I)) – |I|.

Also sufficient: Via Hall's thm for bipartite matching

Feasible Deadlines -> Schedule (use Earliest Deadline First)

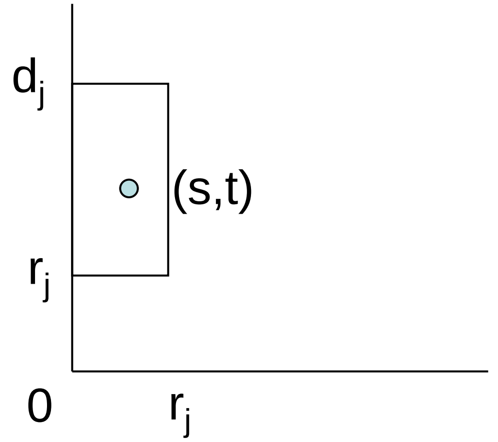
Alternate View

Every interval I = (s,t) "demands" that
Jobs with $r_j \in [s,t]$, $d_j > t$ have total size >= Size(J(I)) – |I|.



Map I=[s,t] to point (s,t) with demand d(I) = Size(J(I)) – |I|.

For job j with deadline d_j
Associate rectangle $[0,r_j] \times [r_j,d_j]$
Weight = $w_j(d_j)$ and capacity = p_j.

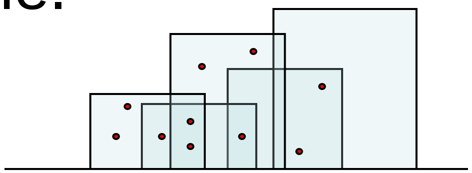


Schedule is feasible iff each point's demand satisfied.

Geometric Formulation

Input: 1) For each job j, several weighted rectangles $[0,r_j] \times [r_j,d_j]$, one for each possible deadline.

2) Points with demands.



Problem: Find min wt collection of rectangles

1) Exactly one rectangle per job

Painful constraint

2) demand for all points is satisfied.

Cheating: If two rectangles of same job used to satisfy demand of a point

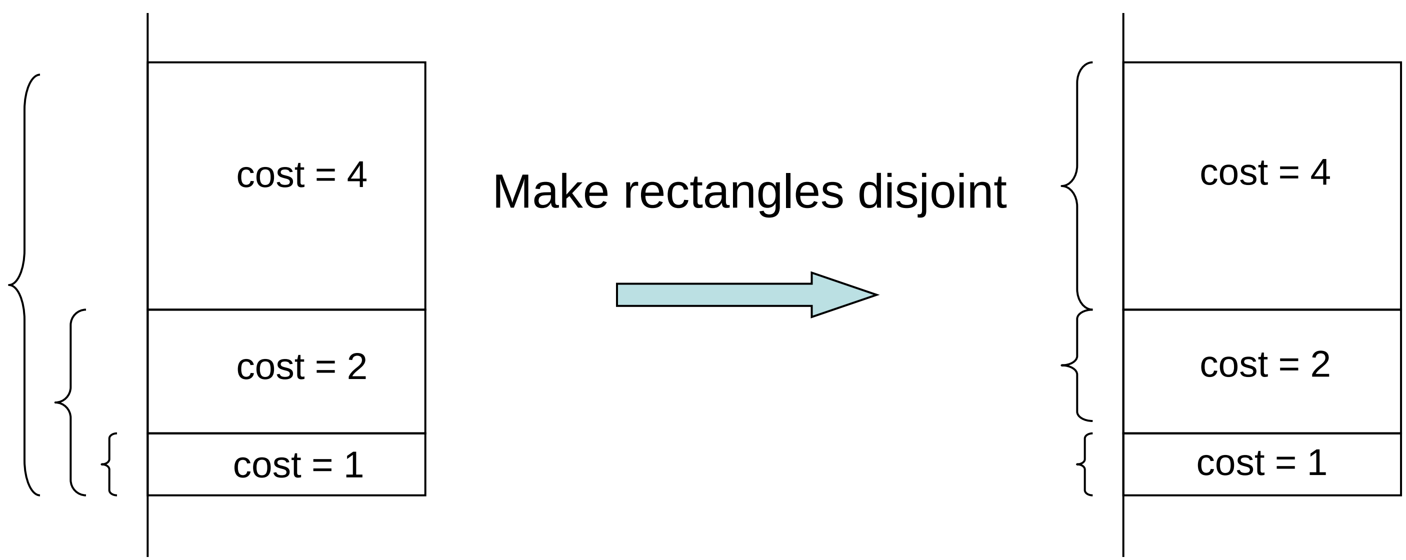
Can be removed by another trick.

2RC problem

For job j, let $d_{j,1}, d_{j,2}, \ldots$ be times when cost doubles.

Suffices to have those rectangles.

A pure covering problem now!



$R_j(i) = [0,r_j] \times [r_j,d_{j,i}]$

$R_j(i) = [0,r_j] \times (d_{j,i-1},d_{j,i}]$

Capacitated Covering Problems

Knapsack Cover Problem

$$\text{Min } \sum_i w_i x_i$$
$$\left.\begin{array}{l}\sum_i c_i x_i \geq B \qquad\qquad 0 <= x_i <= 1\end{array}\right\}$$

Just cover one point

Has arbitrarily large gap (2 items of size B-1, one with wt. 0, other wt. large)

Solution: Knapsack cover inequalities.

$$\sum_{i \in S^c} \min(c_i, B-C(S))\, x_i \geq B-C(S) \quad \forall S$$

Thm [Carr, Fleischer, Leung, Philips 00]: Gives 2 approximation.

Our problem: For each point, we add the knapsack cover inequalities.

Priority Framework

Capacitated Covering (Set system S, capacities, demands)

Priority Cover: Sets, items get a priority
S covers i iff $i \in S$ and $Prio(S) \geq Prio(i)$

Note: capacity =1, demand = 1

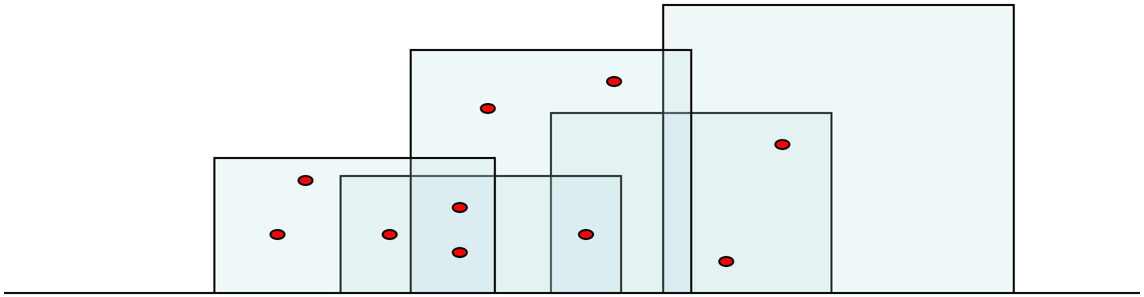
Multi-Cover:
Set system = S

capacity =1, demand = integer

Thm (Chakrabarty, Grant, Konemann 10): $O(\alpha + \beta)$ approx.
where α = approx for priority prob., β for multi-cover version.

Number of priorities = $O(\log C)$ where $C = c_{max}$

What this means for us?



Need good approximations for:

1. Multi-cover version of the above.
2. Priority Set Cover: R covers p if Prio(R) >= Prio(p)

Attach a new dimension to encode priority

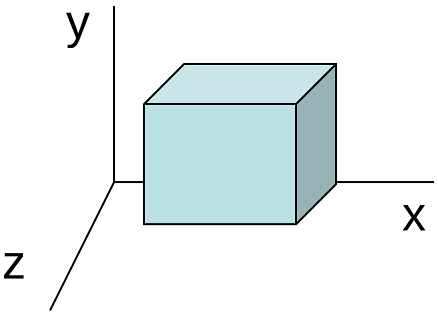
Point in 2d -> Point in 3d — Rectangle -> Cuboid

(s,t,prio) — [0,r] x [r,d] x [0,prio(R)]

Cover points in 3d by cuboids

Touch x-y plane and x-z plane.

O(log P) distinct heights



Geometric Set Cover

Extensive work.
Most of it for unweighted case.

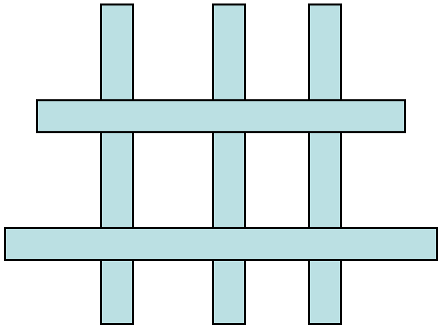
Bronniman-Goodrich: Small ε nets
Imply good approximation.
$1/\varepsilon\ f(1/\varepsilon)$ implies $f(OPT)$ approximation.

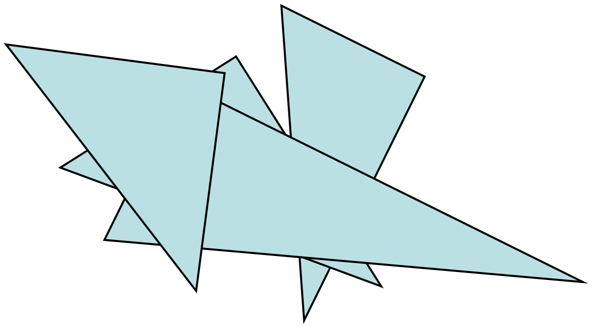
Geometric objects have nets of size $O(1/\varepsilon \log(1/\varepsilon))$
Implies $\log(OPT)$ for geometric problems.

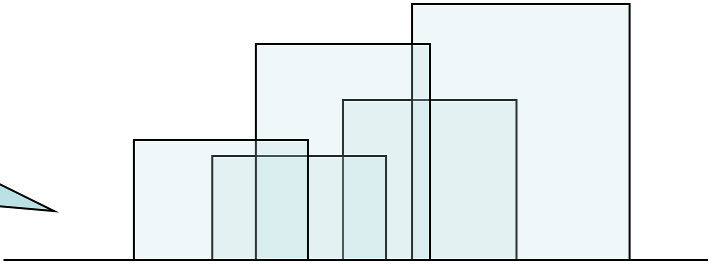
Union Complexity: Clarkson-Varadarajan 07 (unweighted)

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 log k) [Matousek et al 91]
$O(k \log^* k)$
[Aronov, de Berg, Ezra, Sharir 11]

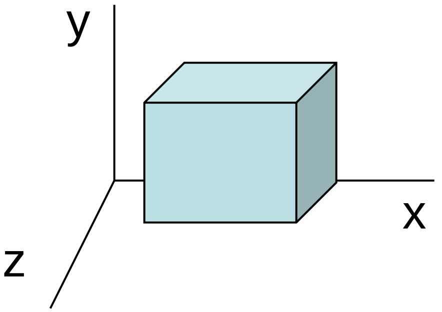
O(k)

Thm [Varadarajan'10]: $\exp(\log^*(n)) \log(h(n))$ approx. for weighted geometric set cover.

Thm [Chan-Grant-Konemann-Sharpe'12]: O(log (h(n))) approx.

Priority Problem

Weighted Set Cover with cuboids

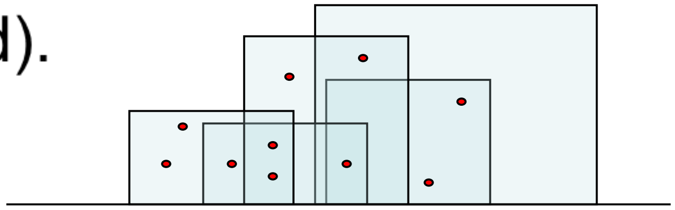


The union complexity of any k of our cuboids is $O(k \log P)$

So, Chan et al result implies $O(\log \log P)$ approx for priority problem.

The multi-cover version

Thm: If demands = 1 (i.e. regular set cover version of 2RC) There is a 2-approximation (LP-based).



Proof: Primal Dual

O(1) via union complexity.

Originally: O(log log nP)

[α approx. for set cover -> $\alpha \log(d_{\max})$ approx. for multi-cover]

New [Bansal-Pruhs]: Extend O(log (h(n)) result to multicover.

Concluding Remarks

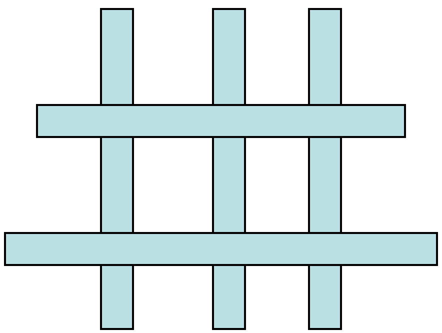
Main open problem:
O(1) approx for the scheduling problem?
Even for special cases eg. wt. flow time.

Could these ideas be useful in real time scheduling?

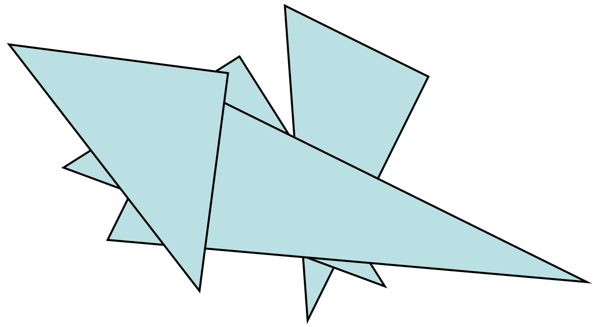
Thank You!

Weighted Geometric Set Cover

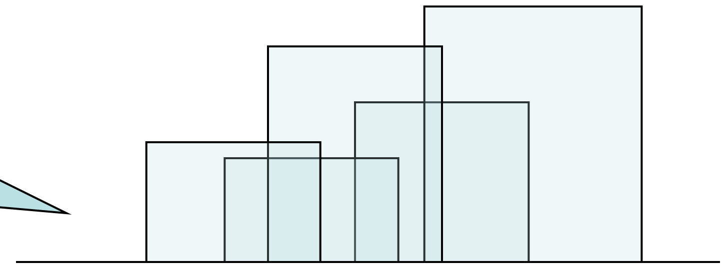
Union Complexity: Take k objects, look at their boundary (vertices,edges, holes). Scales as k h(k)



$\Omega(k^2)$



O(k log log k) [Matousek et al 91]
O(k $\log^*$ k exp(α(k))
[Ezra, Aronov, Sharir 11]



O(k)

Thm [Varadarajan'10]: exp($\log^*$(n)) log (h(n)) approx. for weighted geometric set cover.

Union complexity of k cuboids (for our type) is O(k log P)