Valid Inequalities for the Cumulative Constraint and the Cumulative Job Shop Scheduling Problem

Tallys Yunes [1] Dimitris Magos [2] Ioannis Mourtos [3]

[1] Dept. of Management Science, University of Miami
tallys@miami.edu

[2] Dept. of Informatics, Technological Educational Institute of Athens
dmagos@teiath.gr

[3] Dept. of Mgmt. Science and Technology, Athens Univ. of Econ. & Business
mourtos@aueb.gr

Tepper School of Business, March 25, 2011

Problem Description

Given n jobs (p_j = processing time, c_j=resource consumption) and a single machine with capacity C, the constraint

$$\text{cumulative}((s_1, \ldots, s_n), (p_1, \ldots, p_n), (c_1, \ldots, c_n), C)$$

states that the job start times $s_j \in [r_j, d_j - p_j]$ must be such that the machine capacity is never exceeded

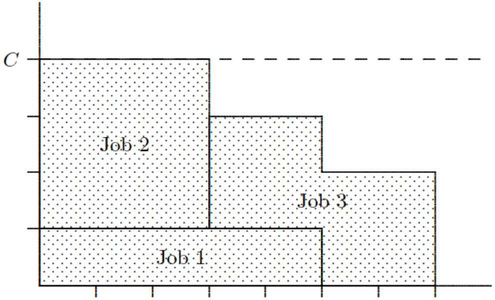
Problem Description

Given n jobs (p_j = processing time, c_j=resource consumption) and a single machine with capacity C, the constraint

$$\text{cumulative}((s_1, \ldots, s_n), (p_1, \ldots, p_n), (c_1, \ldots, c_n), C)$$

states that the job start times $s_j \in [r_j, d_j - p_j]$ must be such that the machine capacity is never exceeded

j	p_j	c_j	r_j	d_j
1	5	1	0	5
2	3	3	0	5
3	4	2	1	7



Problem Description (cont.)

Many applications:

- Production planning and scheduling
- Resource-constrained project scheduling
- Berth allocation at container ports

Problem Description (cont.)

Many applications:

- Production planning and scheduling
- Resource-constrained project scheduling
- Berth allocation at container ports
- Baking (pizza, cakes, cookies,...)

Problem Description (cont.)

Many applications:

- Production planning and scheduling
- Resource-constrained project scheduling
- Berth allocation at container ports
- Baking (pizza, cakes, cookies,...)

As the number of jobs and the time horizon (especially) get larger, optimizing with even a single cumulative can be challenging for both MIP and CP

A library of single-machine Cumulative Scheduling Problems:
CuSPLIB: http://moya.bus.miami.edu/∼tallys/cusplib

Outline

- Problem Description and Related Work
- Revisiting Identical Jobs
- Arbitrary Resource Consumption
- Cumulative Job-Shop Scheduling
- Preliminary Experiments
- Conclusion

Some Related Work

Mostly disjunctive, rather than cumulative, with a few exceptions:

- Queyranne and Schulz '95: parallel machines with non-stationary speeds; $p_j = 1$; generalization of our problem
- Hooker and Yan '01: facet-defining inequalities for identical jobs; valid inequalities for the general case
- Hooker '07: valid inequalities for the general case
- Hardin, Nemhauser, and Savelsbergh '08: $r_j = 0$, $d_j = \infty$, arbitrary c_j, p_j; x_{jt} variables; lifted cover-clique inequalities (some computation: 25 instances with 15 jobs each)

Basic Definitions

Jobs are indexed by $N = \{1, \ldots, n\}$

r_j^{lb} :	earliest release date
r_j^{ub} :	latest release date
p_j :	processing time
c_j :	resource consumption
C :	capacity of the resource

Basic Definitions

Jobs are indexed by $N = \{1, \ldots, n\}$

r_j^{lb} :	earliest release date
r_j^{ub} :	latest release date
p_j :	processing time
c_j :	resource consumption
C :	capacity of the resource

Let s_j denote the start time of job $j \in N$, such that

$$\sum_{j \in N_t} c_j \leq C, \ \ \forall t \tag{1}$$

$$r_j^{lb} \leq s_j \leq r_j^{ub} (= d_j - p_j), \ \ \forall j \in N \tag{2}$$

where $N_t = \{j \in N : s_j \leq t < s_j + p_j\}$

Identical Jobs

We initially assume that all $j \in N$ are identical, i.e.
$r_j^{lb} = r_0$, $r_j^{ub} = r_1$, $p_j = p_0$, $c_j = c_0$

Identical Jobs

We initially assume that all $j \in N$ are identical, i.e. $r_j^{lb} = r_0$, $r_j^{ub} = r_1$, $p_j = p_0$, $c_j = c_0$

Given $K \subseteq N$, let $s(K) = \sum_{j \in K} s_j$

Identical Jobs

We initially assume that all $j \in N$ are identical, i.e.
$r_j^{lb} = r_0$, $r_j^{ub} = r_1$, $p_j = p_0$, $c_j = c_0$

Given $K \subseteq N$, let $s(K) = \sum_{j \in K} s_j$

Let $\lambda = \left\lfloor \frac{C}{c_0} \right\rfloor$ = maximum number of jobs running at a time

Identical Jobs

We initially assume that all $j \in N$ are identical, i.e.
$r_j^{lb} = r_0$, $r_j^{ub} = r_1$, $p_j = p_0$, $c_j = c_0$

Given $K \subseteq N$, let $s(K) = \sum_{j \in K} s_j$

Let $\lambda = \left\lfloor \frac{C}{c_0} \right\rfloor$ = maximum number of jobs running at a time

Let $\rho(K) = \left\lceil \frac{|K|}{\lambda} \right\rceil - 1$ = periods required to run $|K|$ jobs, minus 1

Identical Jobs

We initially assume that all $j \in N$ are identical, i.e.
$r_j^{lb} = r_0$, $r_j^{ub} = r_1$, $p_j = p_0$, $c_j = c_0$

Given $K \subseteq N$, let $s(K) = \sum_{j \in K} s_j$

Let $\lambda = \left\lfloor \frac{C}{c_0} \right\rfloor$ = maximum number of jobs running at a time

Let $\rho(K) = \left\lceil \frac{|K|}{\lambda} \right\rceil - 1$ = periods required to run $|K|$ jobs, minus 1

$s(K) \geq f(K) = |K| r_0 + p_0 \rho(K) \left(|K| - \frac{\lambda}{2}(\rho(K) + 1) \right)$

Identical Jobs

We initially assume that all $j \in N$ are identical, i.e.
$r_j^{lb} = r_0$, $r_j^{ub} = r_1$, $p_j = p_0$, $c_j = c_0$

Given $K \subseteq N$, let $s(K) = \sum_{j \in K} s_j$

Let $\lambda = \left\lfloor \frac{C}{c_0} \right\rfloor$ = maximum number of jobs running at a time

Let $\rho(K) = \left\lceil \frac{|K|}{\lambda} \right\rceil - 1$ = periods required to run $|K|$ jobs, minus 1

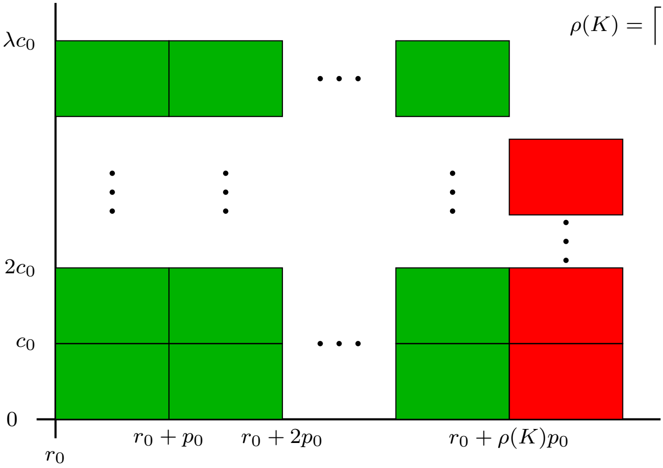
$s(K) \geq f(K) = |K| r_0 + p_0 \rho(K) \left(|K| - \frac{\lambda}{2}(\rho(K) + 1) \right)$

$s(K) \leq g(K) = |K| r_1 - p_0 \rho(K) \left(|K| - \frac{\lambda}{2}(\rho(K) + 1) \right)$

Graphical Intuition Behind $f(K)$

$$\lambda = \left\lfloor \frac{C}{c_0} \right\rfloor$$

$$\rho(K) = \left\lceil \frac{|K|}{\lambda} \right\rceil - 1$$

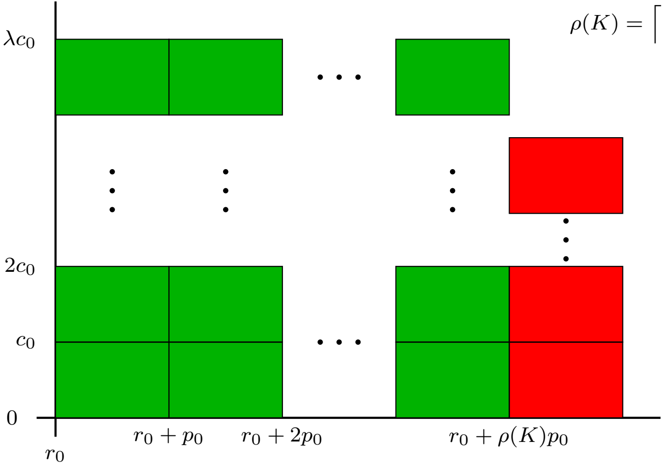


$$f(K) =$$

Graphical Intuition Behind $f(K)$

$$\lambda = \left\lfloor \frac{C}{c_0} \right\rfloor$$

$$\rho(K) = \left\lceil \frac{|K|}{\lambda} \right\rceil - 1$$

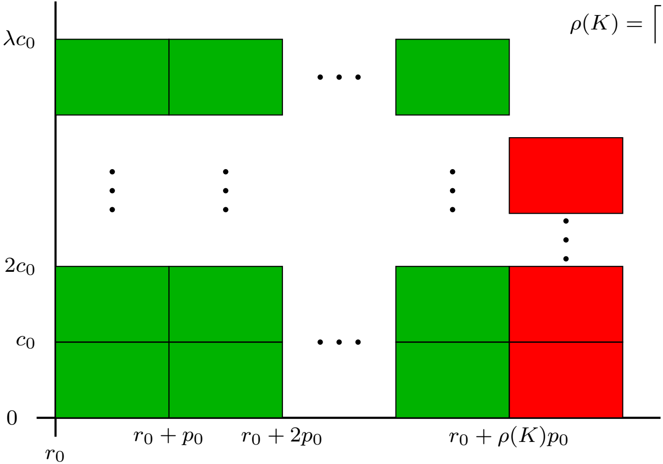


$$f(K) = \sum_{i=0}^{\rho(K)-1} (r_0 + ip_0)\lambda$$

Graphical Intuition Behind $f(K)$

$$\lambda = \left\lfloor \frac{C}{c_0} \right\rfloor$$

$$\rho(K) = \left\lceil \frac{|K|}{\lambda} \right\rceil - 1$$



$$f(K) = \sum_{i=0}^{\rho(K)-1} (r_0 + ip_0)\lambda + (r_0 + \rho(K)p_0)(|K| - \lambda\rho(K))$$

Cumulative Polyhedra with Identical Jobs

f is supermodular and g is submodular on 2^N

Cumulative Polyhedra with Identical Jobs

f is supermodular and g is submodular on 2^N

The lower problem polyhedron of start times (r_0 finite, $r_1 = \infty$) is the extended contrapolymatroid associated with f:

$$EP(f) = \{s \in \mathbb{R}^n : s(K) \geq f(K), \forall K \subseteq N\}$$
$$B(f) = \{s \in EP(f) : s(N) = f(N)\}$$

Cumulative Polyhedra with Identical Jobs

f is supermodular and g is submodular on 2^N

The lower problem polyhedron of start times (r_0 finite, $r_1 = \infty$) is the extended contrapolymatroid associated with f:

$$EP(f) = \{s \in \mathbb{R}^n : s(K) \geq f(K), \forall K \subseteq N\}$$
$$B(f) = \{s \in EP(f) : s(N) = f(N)\}$$

The upper problem polyhedron start times ($r_0 = -\infty$, r_1 finite) is the extended polymatroid associated with g:

$$EP(g) = \{s \in \mathbb{R}^n : s(K) \leq g(K), \forall K \subseteq N\}$$
$$B(g) = \{s \in EP(g) : s(N) = g(N)\}$$

Cumulative Polyhedra with Identical Jobs (cont.)

Theorem

For $j \in N$, $s_j \geq r_0$ define extreme rays of $EP(f)$ and, for $|K| > \lfloor C/c_0 \rfloor$, $s(K) \geq f(K)$ define facets of $EP(f)$

Cumulative Polyhedra with Identical Jobs (cont.)

Theorem

For $j \in N$, $s_j \geq r_0$ define extreme rays of $EP(f)$ and, for $|K| > \lfloor C/c_0 \rfloor$, $s(K) \geq f(K)$ define facets of $EP(f)$

Theorem

$B(f)$ is completely described by $s_j = r_0$ $(j \in N)$ and the inequalities $s(K) \geq f(K)$ when $|K| > \lfloor C/c_0 \rfloor$ and either $|K| = |N| - 1$ or

$$\frac{|N|}{\left\lfloor \frac{C}{c_0} \right\rfloor} > \left\lfloor \frac{|K|}{\left\lfloor \frac{C}{c_0} \right\rfloor} \right\rfloor + 1$$

Cumulative Polyhedra with Identical Jobs (cont.)

Theorem

For $j \in N$, $s_j \geq r_0$ define extreme rays of $EP(f)$ and, for $|K| > \lfloor C/c_0 \rfloor$, $s(K) \geq f(K)$ define facets of $EP(f)$

Theorem

$B(f)$ is completely described by $s_j = r_0$ $(j \in N)$ and the inequalities $s(K) \geq f(K)$ when $|K| > \lfloor C/c_0 \rfloor$ and either $|K| = |N| - 1$ or

$$\frac{|N|}{\left\lfloor \frac{C}{c_0} \right\rfloor} > \left\lfloor \frac{|K|}{\left\lfloor \frac{C}{c_0} \right\rfloor} \right\rfloor + 1$$

Theorem

When $r_1 \geq r_0 + \rho(N)p_0$, the convex hull of all feasible schedules $P = EP(f) \cap EP(g)$ is a generalized polymatroid

Cumulative Polyhedra with Identical Jobs (cont.)

- Can optimize a linear function of s over $EP(f)$ and $EP(g)$ in polynomial time

Cumulative Polyhedra with Identical Jobs (cont.)

- Can optimize a linear function of s over $EP(f)$ and $EP(g)$ in polynomial time
  - Greedy algorithm:
    $s_{\pi(j)} = f(\{\pi(1), \ldots, \pi(j)\}) - f(\{\pi(1), \ldots, \pi(j-1)\})$

Cumulative Polyhedra with Identical Jobs (cont.)

- Can optimize a linear function of s over $EP(f)$ and $EP(g)$ in polynomial time
  - Greedy algorithm:
    $s_{\pi(j)} = f(\{\pi(1), \ldots, \pi(j)\}) - f(\{\pi(1), \ldots, \pi(j-1)\})$
  - $O(n \log n)$ separation: sort s_j's in non-decreasing order

Cumulative Polyhedra with Identical Jobs (cont.)

- Can optimize a linear function of s over $EP(f)$ and $EP(g)$ in polynomial time
  - Greedy algorithm: $s_{\pi(j)} = f(\{\pi(1), \ldots, \pi(j)\}) - f(\{\pi(1), \ldots, \pi(j-1)\})$
  - $O(n \log n)$ separation: sort s_j's in non-decreasing order
- However, inequalities $s(K) \geq f(K)$ and $s(K) \leq g(K)$ can still be useful for problems with multiple cumulative constraints!

Cumulative Polyhedra with Identical Jobs (cont.)

- Can optimize a linear function of s over $EP(f)$ and $EP(g)$ in polynomial time
  - Greedy algorithm: $s_{\pi(j)} = f(\{\pi(1), \ldots, \pi(j)\}) - f(\{\pi(1), \ldots, \pi(j-1)\})$
  - $O(n \log n)$ separation: sort s_j's in non-decreasing order
- However, inequalities $s(K) \geq f(K)$ and $s(K) \leq g(K)$ can still be useful for problems with multiple cumulative constraints!
- Hold that thought...

Cumulative Scheduling with Arbitrary c_j

Cumulative Scheduling with Arbitrary c_j

For all $j \in N$, $r_j^{lb} = r_0$, $r_j^{ub} = \infty$, $p_j = p_0$, but c_j arbitrary

Cumulative Scheduling with Arbitrary c_j

For all $j \in N$, $r_j^{lb} = r_0$, $r_j^{ub} = \infty$, $p_j = p_0$, but c_j arbitrary

Definition

Given $K \subseteq N$, the ordered tuple $Q = (J_1, ..., J_q)$, with $J_i \subset K$, is a feasible partition of K if

(i) $\bigcup_{i=1}^{q} J_i = K$;

(ii) $J_{i_1} \cap J_{i_2} = \emptyset$, for any $i_1 \neq i_2 \in \{1, \ldots, q\}$;

(iii) $\sum_{j \in J_i} c_j \leq C$, for all $i \in \{1, \ldots, q\}$.

Cumulative Scheduling with Arbitrary c_j

For all $j \in N$, $r_j^{lb} = r_0$, $r_j^{ub} = \infty$, $p_j = p_0$, but c_j arbitrary

Definition

Given $K \subseteq N$, the ordered tuple $Q = (J_1, ..., J_q)$, with $J_i \subset K$, is a feasible partition of K if

(i) $\bigcup_{i=1}^{q} J_i = K$;

(ii) $J_{i_1} \cap J_{i_2} = \emptyset$, for any $i_1 \neq i_2 \in \{1, \ldots, q\}$;

(iii) $\sum_{j \in J_i} c_j \leq C$, for all $i \in \{1, \ldots, q\}$.

If Q satisfies $|J_1| \geq |J_2| \geq \cdots \geq |J_q|$, it is called a decreasing feasible partition

Cumulative Scheduling with Arbitrary c_j (cont.)

It's feasible to assign $s_j = r_0 + (i-1)p_0$ for all $J_i \in Q$, which yields

$$\sum_{j \in K} s_j = \sum_{i=1}^{q} (r_0 + (i-1)p_0)|J_i| = h(K, Q)$$

Cumulative Scheduling with Arbitrary c_j (cont.)

It's feasible to assign $s_j = r_0 + (i-1)p_0$ for all $J_i \in Q$, which yields

$$\sum_{j \in K} s_j = \sum_{i=1}^{q} (r_0 + (i-1)p_0)|J_i| = h(K,Q)$$

Lemma

Given $K \subseteq N$, the following inequality is valid for the cumulative scheduling polyhedron

$$\sum_{j \in K} s_j \geq \min_{Q \in \mathfrak{P}(K)} h(K,Q)$$

where $\mathfrak{P}(K)$ is the set of all decreasing feasible partitions of K

Facets for Cumulative Scheduling with Arbitrary c_j

$$\sum_{j \in K} s_j \geq \min_{Q \in \mathfrak{P}(K)} h(K,Q) = \min_{Q \in \mathfrak{P}(K)} \sum_{i=1}^{|Q|} (r_0 + (i-1)p_0)|J_i|$$

Theorem

Given $K \subseteq N$, let $Q^* = (J_1, \ldots, J_q) = \text{argmin}_{Q \in \mathfrak{P}(K)} h(K,Q)$. The resulting inequality defines a facet of the cumulative scheduling polyhedron if

(i) $q \geq 2$

(ii) $\sum_{j \in J_i} c_j - \min_{j \in J_i} c_j + \min_{j \in J_{i+1}} c_j \leq C, \ \ \forall i = 1, \ldots, q-1$

(iii) *Either $|J_q| = 1$ or $\sum_{j \in J_q} c_j - \min_{j \in J_q} c_j + c_{j^*} \leq C$,*

for some $j^* \in \bigcup_{i=1}^{q-1} J_i$

What Kind of Partition Minimizes $h(K,Q)$?

$$\sum_{j \in K} s_j \geq \min_{Q \in \mathfrak{P}(K)} h(K,Q) = \min_{Q \in \mathfrak{P}(K)} \sum_{i=1}^{|Q|} (r_0 + (i-1)p_0)|J_i|$$

What Kind of Partition Minimizes $h(K, Q)$?

$$\sum_{j \in K} s_j \geq \min_{Q \in \mathfrak{P}(K)} h(K, Q) = \min_{Q \in \mathfrak{P}(K)} \sum_{i=1}^{|Q|} (r_0 + (i-1)p_0)|J_i|$$

Definition

Let $Q = (J_1, \ldots, J_q)$ and $Q' = (J'_1, \ldots, J'_q)$ be two elements of $\mathfrak{P}(q, K)$. Q' is said to majorize Q (denoted $Q' \succ Q$) if $\sum_{i=1}^{v} |J'_i| \geq \sum_{i=1}^{v} |J_i|$, for all $v \in \{1, \ldots, q-1\}$, and $\sum_{i=1}^{q} |J'_i| = \sum_{i=1}^{q} |J_i|$. Moreover, Q' is called a majorizer of $\mathfrak{P}(q, K)$ if $Q' \succ Q$ for all $Q \in \mathfrak{P}(q, K)$. The set of all majorizers of $\mathfrak{P}(q, K)$ is denoted by $\mathfrak{P}^{\succ}(q, K)$

What Kind of Partition Minimizes $h(K, Q)$?

$$\sum_{j \in K} s_j \geq \min_{Q \in \mathfrak{P}(K)} h(K, Q) = \min_{Q \in \mathfrak{P}(K)} \sum_{i=1}^{|Q|} (r_0 + (i-1)p_0)|J_i|$$

Definition

Let $Q = (J_1, \ldots, J_q)$ and $Q' = (J'_1, \ldots, J'_q)$ be two elements of $\mathfrak{P}(q, K)$. Q' is said to majorize Q (denoted $Q' \succ Q$) if $\sum_{i=1}^{v} |J'_i| \geq \sum_{i=1}^{v} |J_i|$, for all $v \in \{1, \ldots, q-1\}$, and $\sum_{i=1}^{q} |J'_i| = \sum_{i=1}^{q} |J_i|$. Moreover, Q' is called a majorizer of $\mathfrak{P}(q, K)$ if $Q' \succ Q$ for all $Q \in \mathfrak{P}(q, K)$. The set of all majorizers of $\mathfrak{P}(q, K)$ is denoted by $\mathfrak{P}^{\succ}(q, K)$

Proposition

Given $K \subseteq N$ and a fixed partition size q, the value of $h(K, Q)$ is minimized by a majorizer of $\mathfrak{P}(q, K)$

What Kind of Partition Minimizes $h(K, Q)$? (cont.)

Intuitive guess: A majorizer with minimum size q^*?

What Kind of Partition Minimizes $h(K, Q)$? (cont.)

Intuitive guess: A majorizer with minimum size q^*?

Smaller $q \not\Rightarrow$ smaller $h(K, Q)$:

What Kind of Partition Minimizes $h(K, Q)$? (cont.)

Intuitive guess: A majorizer with minimum size q^*?

Smaller $q \not\Rightarrow$ smaller $h(K, Q)$:
$K = \{1, \ldots, 10\}$, $r_0 = 1$, $p_0 = 1$, $C = 10$, $c = (1, 1, 1, 1, 1, 1, 1, 1, 5, 6)$

What Kind of Partition Minimizes $h(K, Q)$? (cont.)

Intuitive guess: A majorizer with minimum size q^*?

Smaller $q \not\Rightarrow$ smaller $h(K, Q)$:
$K = \{1, \ldots, 10\}$, $r_0 = 1$, $p_0 = 1$, $C = 10$, $c = (1, 1, 1, 1, 1, 1, 1, 1, 5, 6)$
$Q_1 = (\{1, 2, 3, 4, 9\}, \{5, 6, 7, 8, 10\})$, $Q_2 = (\{1, 2, 3, 4, 5, 9\}, \{6, 7, 8, 10\})$

What Kind of Partition Minimizes $h(K, Q)$? (cont.)

Intuitive guess: A majorizer with minimum size q^*?

Smaller $q \not\Rightarrow$ smaller $h(K, Q)$:
$K = \{1, \ldots, 10\}$, $r_0 = 1$, $p_0 = 1$, $C = 10$, $c = (1, 1, 1, 1, 1, 1, 1, 1, 5, 6)$
$Q_1 = (\{1, 2, 3, 4, 9\}, \{5, 6, 7, 8, 10\})$, $Q_2 = (\{1, 2, 3, 4, 5, 9\}, \{6, 7, 8, 10\})$
$Q_2 \succ Q_1$, $h(K, Q_2) = 14$

What Kind of Partition Minimizes $h(K, Q)$? (cont.)

Intuitive guess: A majorizer with minimum size q^*?

Smaller $q \not\Rightarrow$ smaller $h(K, Q)$:
$K = \{1, \ldots, 10\}$, $r_0 = 1$, $p_0 = 1$, $C = 10$, $c = (1, 1, 1, 1, 1, 1, 1, 1, 5, 6)$
$Q_1 = (\{1, 2, 3, 4, 9\}, \{5, 6, 7, 8, 10\})$, $Q_2 = (\{1, 2, 3, 4, 5, 9\}, \{6, 7, 8, 10\})$
$Q_2 \succ Q_1$, $h(K, Q_2) = 14$
But $Q_3 = (\{1, ..., 8\}, \{9\}, \{10\})$ yields $h(K, Q_3) = 13$

What Kind of Partition Minimizes $h(K, Q)$? (cont.)

Intuitive guess: A majorizer with minimum size q^*?

Smaller $q \not\Rightarrow$ smaller $h(K, Q)$:
$K = \{1, \ldots, 10\}$, $r_0 = 1$, $p_0 = 1$, $C = 10$, $c = (1, 1, 1, 1, 1, 1, 1, 1, 5, 6)$
$Q_1 = (\{1, 2, 3, 4, 9\}, \{5, 6, 7, 8, 10\})$, $Q_2 = (\{1, 2, 3, 4, 5, 9\}, \{6, 7, 8, 10\})$
$Q_2 \succ Q_1$, $h(K, Q_2) = 14$
But $Q_3 = (\{1, ..., 8\}, \{9\}, \{10\})$ yields $h(K, Q_3) = 13$

A majorizer may not exist:

What Kind of Partition Minimizes $h(K, Q)$? (cont.)

Intuitive guess: A majorizer with minimum size q^*?

Smaller $q \not\Rightarrow$ smaller $h(K, Q)$:
$K = \{1, \ldots, 10\}$, $r_0 = 1$, $p_0 = 1$, $C = 10$, $c = (1, 1, 1, 1, 1, 1, 1, 1, 5, 6)$
$Q_1 = (\{1, 2, 3, 4, 9\}, \{5, 6, 7, 8, 10\})$, $Q_2 = (\{1, 2, 3, 4, 5, 9\}, \{6, 7, 8, 10\})$
$Q_2 \succ Q_1$, $h(K, Q_2) = 14$
But $Q_3 = (\{1, ..., 8\}, \{9\}, \{10\})$ yields $h(K, Q_3) = 13$

A majorizer may not exist:
$K = \{1, \ldots, 9\}$, $C = 8$, $c = (1, 1, 1, 1, 3, 3, 3, 3, 3)$, $q^* = 3$

What Kind of Partition Minimizes $h(K, Q)$? (cont.)

Intuitive guess: A majorizer with minimum size q^*?

Smaller $q \not\Rightarrow$ smaller $h(K, Q)$:
$K = \{1, \ldots, 10\}$, $r_0 = 1$, $p_0 = 1$, $C = 10$, $c = (1, 1, 1, 1, 1, 1, 1, 1, 5, 6)$
$Q_1 = (\{1, 2, 3, 4, 9\}, \{5, 6, 7, 8, 10\})$, $Q_2 = (\{1, 2, 3, 4, 5, 9\}, \{6, 7, 8, 10\})$
$Q_2 \succ Q_1$, $h(K, Q_2) = 14$
But $Q_3 = (\{1, ..., 8\}, \{9\}, \{10\})$ yields $h(K, Q_3) = 13$

A majorizer may not exist:
$K = \{1, \ldots, 9\}$, $C = 8$, $c = (1, 1, 1, 1, 3, 3, 3, 3, 3)$, $q^* = 3$
$Q_1 = (\{1, 2, 3, 4, 5\}, \{6, 7\}, \{8, 9\})$

What Kind of Partition Minimizes $h(K, Q)$? (cont.)

Intuitive guess: A majorizer with minimum size q^*?

Smaller $q \not\Rightarrow$ smaller $h(K, Q)$:
$K = \{1, \ldots, 10\}$, $r_0 = 1$, $p_0 = 1$, $C = 10$, $c = (1, 1, 1, 1, 1, 1, 1, 1, 5, 6)$
$Q_1 = (\{1, 2, 3, 4, 9\}, \{5, 6, 7, 8, 10\})$, $Q_2 = (\{1, 2, 3, 4, 5, 9\}, \{6, 7, 8, 10\})$
$Q_2 \succ Q_1$, $h(K, Q_2) = 14$
But $Q_3 = (\{1, ..., 8\}, \{9\}, \{10\})$ yields $h(K, Q_3) = 13$

A majorizer may not exist:
$K = \{1, \ldots, 9\}$, $C = 8$, $c = (1, 1, 1, 1, 3, 3, 3, 3, 3)$, $q^* = 3$
$Q_1 = (\{1, 2, 3, 4, 5\}, \{6, 7\}, \{8, 9\})$
$Q_2 = (\{1, 2, 5, 6\}, \{3, 4, 7, 8\}, \{9\})$

What Kind of Partition Minimizes $h(K, Q)$? (cont.)

Intuitive guess: A majorizer with minimum size q^*?

Smaller $q \not\Rightarrow$ smaller $h(K, Q)$:
$K = \{1, \dots, 10\}$, $r_0 = 1$, $p_0 = 1$, $C = 10$, $c = (1, 1, 1, 1, 1, 1, 1, 1, 5, 6)$
$Q_1 = (\{1, 2, 3, 4, 9\}, \{5, 6, 7, 8, 10\})$, $Q_2 = (\{1, 2, 3, 4, 5, 9\}, \{6, 7, 8, 10\})$
$Q_2 \succ Q_1$, $h(K, Q_2) = 14$
But $Q_3 = (\{1, ..., 8\}, \{9\}, \{10\})$ yields $h(K, Q_3) = 13$

A majorizer may not exist:
$K = \{1, \dots, 9\}$, $C = 8$, $c = (1, 1, 1, 1, 3, 3, 3, 3, 3)$, $q^* = 3$
$Q_1 = (\{1, 2, 3, 4, 5\}, \{6, 7\}, \{8, 9\})$
$Q_2 = (\{1, 2, 5, 6\}, \{3, 4, 7, 8\}, \{9\})$
Neither $Q_1 \succ Q_2$, nor $Q_2 \succ Q_1$

What Kind of Partition Minimizes $h(K, Q)$? (cont.)

Intuitive guess: A majorizer with minimum size q^*?

Smaller $q \not\Rightarrow$ smaller $h(K, Q)$:
$K = \{1, \ldots, 10\}$, $r_0 = 1$, $p_0 = 1$, $C = 10$, $c = (1, 1, 1, 1, 1, 1, 1, 1, 5, 6)$
$Q_1 = (\{1, 2, 3, 4, 9\}, \{5, 6, 7, 8, 10\})$, $Q_2 = (\{1, 2, 3, 4, 5, 9\}, \{6, 7, 8, 10\})$
$Q_2 \succ Q_1$, $h(K, Q_2) = 14$
But $Q_3 = (\{1, ..., 8\}, \{9\}, \{10\})$ yields $h(K, Q_3) = 13$

A majorizer may not exist:
$K = \{1, \ldots, 9\}$, $C = 8$, $c = (1, 1, 1, 1, 3, 3, 3, 3, 3)$, $q^* = 3$
$Q_1 = (\{1, 2, 3, 4, 5\}, \{6, 7\}, \{8, 9\})$
$Q_2 = (\{1, 2, 5, 6\}, \{3, 4, 7, 8\}, \{9\})$
Neither $Q_1 \succ Q_2$, nor $Q_2 \succ Q_1$
No majorizer of size 3 exists.

What Kind of Partition Minimizes $h(K, Q)$? (cont.)

Intuitive guess: A majorizer with minimum size q^*?

Smaller $q \not\Rightarrow$ smaller $h(K, Q)$:
$K = \{1, \ldots, 10\}$, $r_0 = 1$, $p_0 = 1$, $C = 10$, $c = (1, 1, 1, 1, 1, 1, 1, 1, 5, 6)$
$Q_1 = (\{1, 2, 3, 4, 9\}, \{5, 6, 7, 8, 10\})$, $Q_2 = (\{1, 2, 3, 4, 5, 9\}, \{6, 7, 8, 10\})$
$Q_2 \succ Q_1$, $h(K, Q_2) = 14$
But $Q_3 = (\{1, ..., 8\}, \{9\}, \{10\})$ yields $h(K, Q_3) = 13$

A majorizer may not exist:
$K = \{1, \ldots, 9\}$, $C = 8$, $c = (1, 1, 1, 1, 3, 3, 3, 3, 3)$, $q^* = 3$
$Q_1 = (\{1, 2, 3, 4, 5\}, \{6, 7\}, \{8, 9\})$
$Q_2 = (\{1, 2, 5, 6\}, \{3, 4, 7, 8\}, \{9\})$
Neither $Q_1 \succ Q_2$, nor $Q_2 \succ Q_1$
No majorizer of size 3 exists.

We suspect it is NP-Hard (GAP), but no proof yet

A Lower Bound on $h(K, Q)$

Let $K = \{1, \ldots, |K|\} \subseteq N$

$$\min \sum_{j=1}^{|K|} \sum_{t=1}^{|K|} \frac{(r_0 + (t-1)p_0)}{c_j} y_{jt}$$

$$\sum_{t=1}^{|K|} y_{jt} = c_j, \ \ \forall\, j \in K$$

$$\sum_{j=1}^{|K|} y_{jt} \leq C, \ \ \forall\, t \in K$$

$$y_{jt} \in \{0, c_j\}, \ \ \forall\, j, t \in K$$

where y_{jt} = resource consumption of j at time t

A Lower Bound on $h(K,Q)$

Let $K = \{1, \ldots, |K|\} \subseteq N$

$$\min \sum_{j=1}^{|K|} \sum_{t=1}^{|K|} \frac{(r_0 + (t-1)p_0)}{c_j} y_{jt}$$

$$\sum_{t=1}^{|K|} y_{jt} = c_j, \ \ \forall\, j \in K$$

$$\sum_{j=1}^{|K|} y_{jt} \leq C, \ \ \forall\, t \in K$$

$$y_{jt} \in \{0, c_j\}, \ \ \forall\, j, t \in K$$

where $y_{jt} =$ resource consumption of j at time t

The LP relaxation of this model is a transportation problem

A Lower Bound on $h(K, Q)$ (cont.)

Recall previous example:

$K = \{1, \ldots, 10\}$, $r_0 = 1$, $p_0 = 1$, $C = 10$, $c = (1, 1, 1, 1, 1, 1, 1, 1, 5, 6)$

$Q_3 = (\{1, ..., 8\}, \{9\}, \{10\})$ yields minimum $h(K, Q_3) = 13$

A Lower Bound on $h(K, Q)$ (cont.)

Recall previous example:

$K = \{1, \ldots, 10\}$, $r_0 = 1$, $p_0 = 1$, $C = 10$, $c = (1, 1, 1, 1, 1, 1, 1, 1, 5, 6)$

$Q_3 = (\{1, ..., 8\}, \{9\}, \{10\})$ yields minimum $h(K, Q_3) = 13$

The transportation problem relaxation gives:

$y_{11} = y_{21} = \cdots = y_{81} = 1$

$y_{91} = 2$, $y_{92} = 3$

$y_{10,2} = 6$

For a value of 11.6, which can be rounded up to 12

Generalization of Job-Shop Scheduling

Generalization of Job-Shop Scheduling

Traditional job-shop:

- m disjunctive machines, n jobs
- s_{ij} = start time of job j on machine i
- Each job visits (all) machines in a specific order

Generalization of Job-Shop Scheduling

Traditional job-shop:

- m disjunctive machines, n jobs
- s_{ij} = start time of job j on machine i
- Each job visits (all) machines in a specific order

Known Fact: $J3|p_{ij} = 1|C_{\max}$ or $\sum C_j$ is NP-Hard

Generalization of Job-Shop Scheduling

Traditional job-shop:

- m disjunctive machines, n jobs
- s_{ij} = start time of job j on machine i
- Each job visits (all) machines in a specific order

Known Fact: $J3|p_{ij} = 1|C_{\max}$ or $\sum C_j$ is NP-Hard

Cumulative job-shop:

- m cumulative machines, n jobs, same s_{ij} variables
- Each job visits (all) machines in a specific order
- Jobs have identical c_{ij} on each machine, but may differ across machines

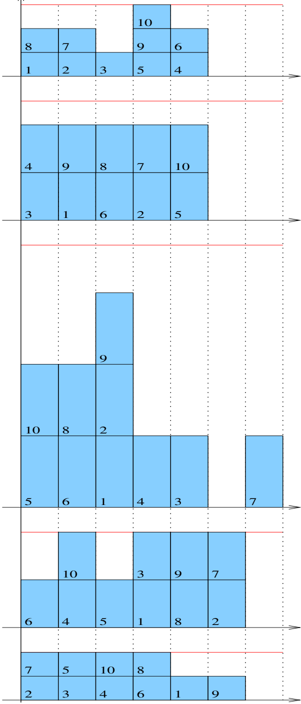
Cumulative Job-Shop: $p_{ij} = 1, \min \sum s_{ij}$

Sample instance:

$m = 5$: $C_i = (3, 5, 11, 4, 2)$
$n = 10$: $c_i = (1, 2, 3, 2, 1)$

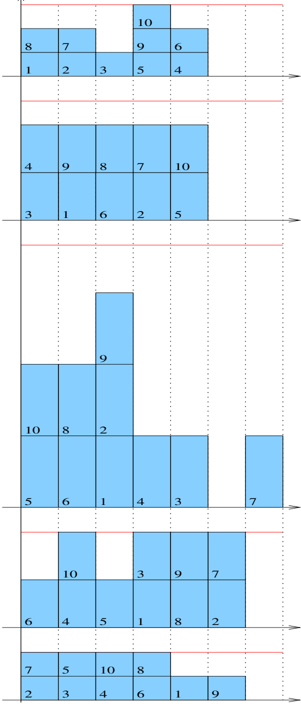
Machine sequences:

Job 1: 1, 2, 3, 4, 5
Job 2: 5, 1, 3, 2, 4
Job 3: 2, 5, 1, 4, 3
Job 4: 2, 4, 5, 3, 1
Job 5: 3, 5, 4, 1, 2
Job 6: 4, 3, 2, 5, 1
Job 7: 5, 1, 2, 4, 3
Job 8: 1, 3, 2, 5, 4
Job 9: 2, 3, 1, 4, 5
Job 10: 3, 4, 5, 1, 2



Cumulative Job-Shop: $p_{ij} = 1, \min \sum s_{ij}$

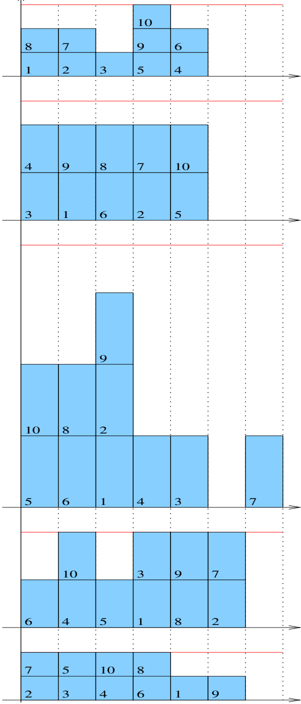
How to use $s(K) \geq f(K)$ and $s(K) \leq g(K)$ cuts:



Cumulative Job-Shop: $p_{ij} = 1, \min \sum s_{ij}$

How to use $s(K) \geq f(K)$ and $s(K) \leq g(K)$ cuts:

- Machine sequencing constraints impose initial release dates on jobs: r_{ij}

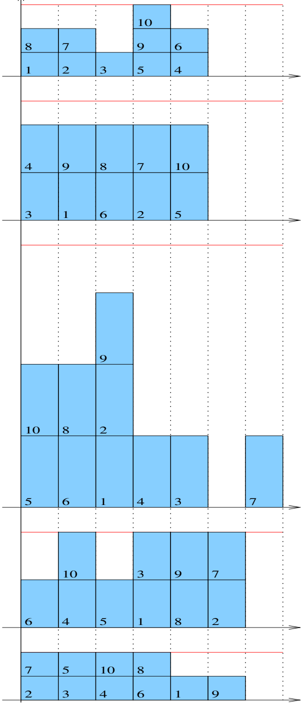


Cumulative Job-Shop: $p_{ij} = 1, \min \sum s_{ij}$

How to use $s(K) \geq f(K)$ and $s(K) \leq g(K)$ cuts:

- Machine sequencing constraints impose initial release dates on jobs: r_{ij}

Job 1: 1, 2, 3, 4, 5
Job 2: 5, 1, 3, 2, 4
Job 3: 2, 5, 1, 4, 3
Job 4: 2, 4, 5, 3, 1
Job 5: 3, 5, 4, 1, 2
Job 6: 4, 3, 2, 5, 1
Job 7: 5, 1, 2, 4, 3
Job 8: 1, 3, 2, 5, 4
Job 9: 2, 3, 1, 4, 5
Job 10: 3, 4, 5, 1, 2

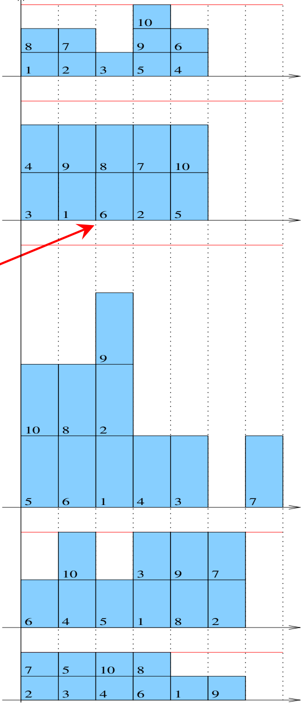


Cumulative Job-Shop: $p_{ij} = 1, \min \sum s_{ij}$

How to use $s(K) \geq f(K)$ and $s(K) \leq g(K)$ cuts:

- Machine sequencing constraints impose initial release dates on jobs: r_{ij}

Job 1: 1, 2, 3, 4, 5
Job 2: 5, 1, 3, 2, 4
Job 3: 2, 5, 1, 4, 3
Job 4: 2, 4, 5, 3, 1
Job 5: 3, 5, 4, 1, 2
Job 6: 4, 3, 2, 5, 1
Job 7: 5, 1, 2, 4, 3
Job 8: 1, 3, 2, 5, 4
Job 9: 2, 3, 1, 4, 5
Job 10: 3, 4, 5, 1, 2



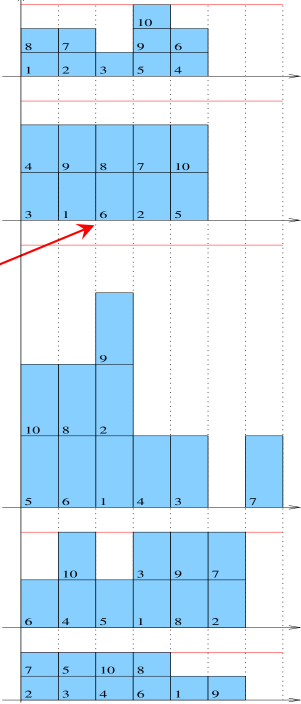
Cumulative Job-Shop: $p_{ij} = 1, \min \sum s_{ij}$

How to use $s(K) \geq f(K)$ and $s(K) \leq g(K)$ cuts:

- Machine sequencing constraints impose initial release dates on jobs: r_{ij}

  Job 1: 1, 2, 3, 4, 5
  Job 2: 5, 1, 3, 2, 4
  Job 3: 2, 5, 1, 4, 3
  Job 4: 2, 4, 5, 3, 1
  Job 5: 3, 5, 4, 1, 2
  Job 6: 4, 3, 2, 5, 1
  Job 7: 5, 1, 2, 4, 3
  Job 8: 1, 3, 2, 5, 4
  Job 9: 2, 3, 1, 4, 5
  Job 10: 3, 4, 5, 1, 2

- At machine i, jobs with $r_{ij} = r$ form **class** r



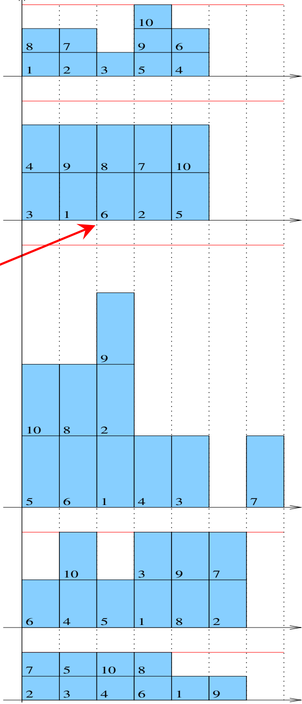
Cumulative Job-Shop: $p_{ij} = 1, \min \sum s_{ij}$

How to use $s(K) \geq f(K)$ and $s(K) \leq g(K)$ cuts:

- Machine sequencing constraints impose initial release dates on jobs: r_{ij}

Job 1: 1, 2, 3, 4, 5
Job 2: 5, 1, 3, 2, 4
Job 3: 2, 5, 1, 4, 3
Job 4: 2, 4, 5, 3, 1
Job 5: 3, 5, 4, 1, 2
Job 6: 4, 3, 2, 5, 1
Job 7: 5, 1, 2, 4, 3
Job 8: 1, 3, 2, 5, 4
Job 9: 2, 3, 1, 4, 5
Job 10: 3, 4, 5, 1, 2

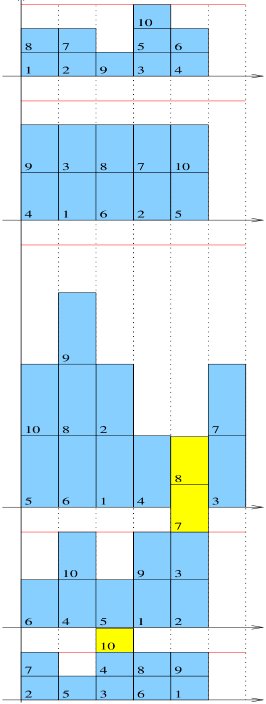
- At machine i, jobs with $r_{ij} = r$ form class r
- Separate $s(K) \geq f(K)$ for each class



Cumulative Job-Shop: $p_{ij} = 1, \min \sum s_{ij}$

How to use $s(K) \geq f(K)$ and $s(K) \leq g(K)$ cuts:

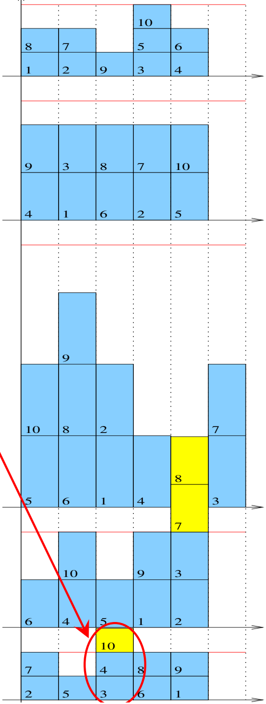
- If infeasible:



Cumulative Job-Shop: $p_{ij} = 1, \min \sum s_{ij}$

How to use $s(K) \geq f(K)$ and $s(K) \leq g(K)$ cuts:

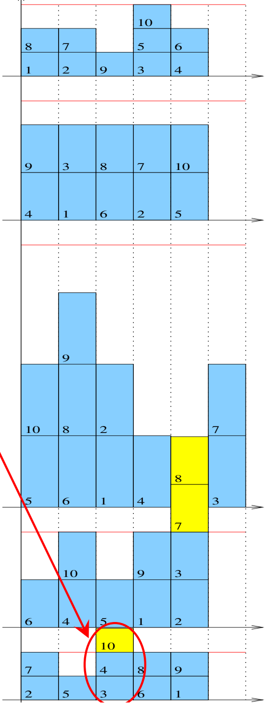
- If infeasible:
  Either 3, 4, or 10 must have $r_{5j} < 2$



Cumulative Job-Shop: $p_{ij} = 1, \min \sum s_{ij}$

How to use $s(K) \geq f(K)$ and $s(K) \leq g(K)$ cuts:

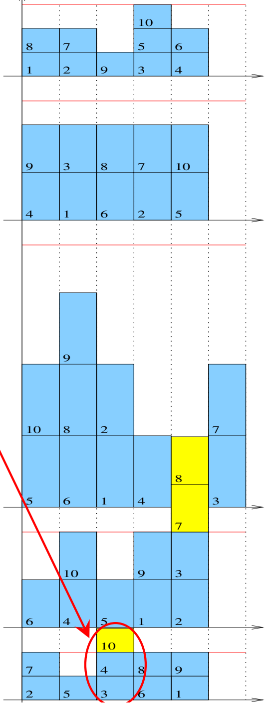
- If infeasible:
  Either 3, 4, or 10 must have $r_{5j} < 2$
  Say it's job 3



Cumulative Job-Shop: $p_{ij} = 1, \min \sum s_{ij}$

How to use $s(K) \geq f(K)$ and $s(K) \leq g(K)$ cuts:

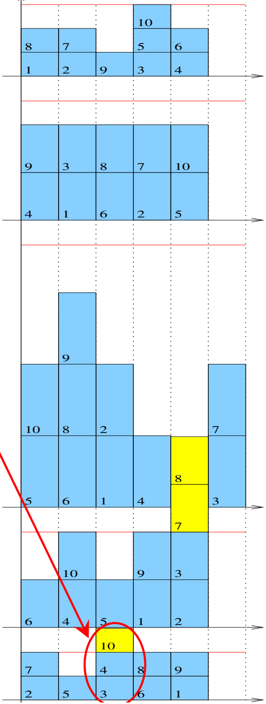
- If infeasible:
  Either 3, 4, or 10 must have $r_{5j} < 2$
  Say it's job 3
  Branch: $s_{53} \leq 1$ and $s_{53} \geq 2$



Cumulative Job-Shop: $p_{ij} = 1, \min \sum s_{ij}$

How to use $s(K) \geq f(K)$ and $s(K) \leq g(K)$ cuts:

- If infeasible:
  Either 3, 4, or 10 must have $r_{5j} < 2$
  Say it's job 3
  Branch: $s_{53} \leq 1$ and $s_{53} \geq 2$
  Update: $r^{ub}_{53} = 1$ and $r^{lb}_{53} = 2$

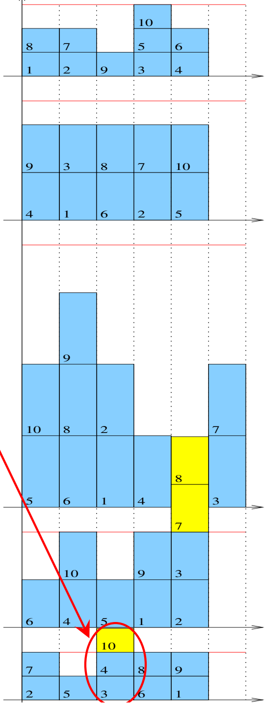


Cumulative Job-Shop: $p_{ij} = 1, \min \sum s_{ij}$

How to use $s(K) \geq f(K)$ and $s(K) \leq g(K)$ cuts:

- If infeasible:
  Either 3, 4, or 10 must have $r_{5j} < 2$
  Say it's job 3
  Branch: $s_{53} \leq 1$ and $s_{53} \geq 2$
  Update: $r_{53}^{ub} = 1$ and $r_{53}^{lb} = 2$

- Branching alters job classes: allows cuts to remove violations



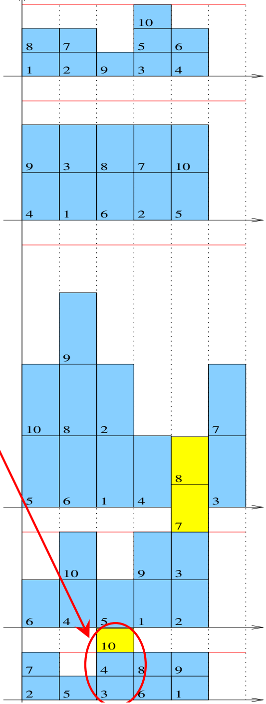
Cumulative Job-Shop: $p_{ij} = 1, \min \sum s_{ij}$

How to use $s(K) \geq f(K)$ and $s(K) \leq g(K)$ cuts:

- If infeasible:
  Either 3, 4, or 10 must have $r_{5j} < 2$
  Say it's job 3
  Branch: $s_{53} \leq 1$ and $s_{53} \geq 2$
  Update: $r_{53}^{ub} = 1$ and $r_{53}^{lb} = 2$

- Branching alters job classes: allows cuts to remove violations

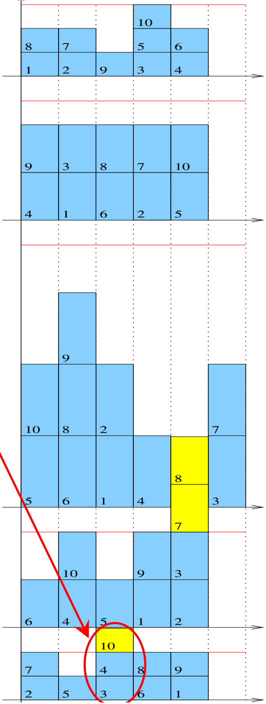
- Branching imposes upper bounds on s_{ij}: separate $s(K) \leq g(K)$ as well



Cumulative Job-Shop: $p_{ij} = 1, \min \sum s_{ij}$

How to use $s(K) \geq f(K)$ and $s(K) \leq g(K)$ cuts:

- If infeasible:
  Either 3, 4, or 10 must have $r_{5j} < 2$
  Say it's job 3
  Branch: $s_{53} \leq 1$ and $s_{53} \geq 2$
  Update: $r_{53}^{ub} = 1$ and $r_{53}^{lb} = 2$
- Branching alters job classes: allows cuts to remove violations
- Branching imposes upper bounds on s_{ij}: separate $s(K) \leq g(K)$ as well
- How to pick violation and job to branch on?
  Still experimenting with this...



Before we continue...

Before we continue...

Before we continue...

Potential Branching Heuristic:

Before we continue...

$$\left(\begin{array}{l}\underline{\text{Potential Branching Heuristic}}: \\ \\ \quad \text{Have generic problem } \mathcal{P} \\ \\ \\ \\ \\ \\ \\ \end{array}\right.$$

Before we continue...

Potential Branching Heuristic:

Have generic problem $\mathcal{P}$

Have family $\mathcal{F}$ of cuts
that work when $\mathcal{P}$ is restricted
to satisfy certain conditions $\mathcal{R}$

Before we continue...

Potential Branching Heuristic:

Have generic problem $\mathcal{P}$

Have family $\mathcal{F}$ of cuts
that work when $\mathcal{P}$ is restricted
to satisfy certain conditions $\mathcal{R}$

Bias branching on $\mathcal{P}$ to bring it
closer to satisfying $\mathcal{R}$ during search

Before we continue...

Potential Branching Heuristic:

Have generic problem $\mathcal{P}$

Have family $\mathcal{F}$ of cuts
that work when $\mathcal{P}$ is restricted
to satisfy certain conditions $\mathcal{R}$

Bias branching on $\mathcal{P}$ to bring it
closer to satisfying $\mathcal{R}$ during search

Separate $\mathcal{F}$ cuts when violated

Before we continue...

Potential Branching Heuristic:

Have generic problem $\mathcal{P}$

Have family $\mathcal{F}$ of cuts
that work when $\mathcal{P}$ is restricted
to satisfy certain conditions $\mathcal{R}$

Bias branching on $\mathcal{P}$ to bring it
closer to satisfying $\mathcal{R}$ during search

Separate $\mathcal{F}$ cuts when violated

Preliminary Computational Experiments

Initial set of instances generated by hand:

- Number of machines $m = 5$ and $C_i = (3, 5, 11, 4, 2)$
- Number of jobs $n \in \{10, 15, 20, 25, 30, 35, 40, 50, 60\}$
- Resource consumption $c_i = (1, 2, 3, 2, 1)$
- $p_{ij} = 1$ and time horizon $T = 30$

Preliminary Computational Experiments

Initial set of instances generated by hand:

- Number of machines $m = 5$ and $C_i = (3, 5, 11, 4, 2)$
- Number of jobs $n \in \{10, 15, 20, 25, 30, 35, 40, 50, 60\}$
- Resource consumption $c_i = (1, 2, 3, 2, 1)$
- $p_{ij} = 1$ and time horizon $T = 30$

Solution methods:

- Default CPLEX 12.2 with typical x_{ijt} formulation

Preliminary Computational Experiments

Initial set of instances generated by hand:

- Number of machines $m = 5$ and $C_i = (3, 5, 11, 4, 2)$
- Number of jobs $n \in \{10, 15, 20, 25, 30, 35, 40, 50, 60\}$
- Resource consumption $c_i = (1, 2, 3, 2, 1)$
- $p_{ij} = 1$ and time horizon $T = 30$

Solution methods:

- Default CPLEX 12.2 with typical x_{ijt} formulation
- B&C with s_{ij} variables, classed f and g cuts, and 2 branching strategies:

Preliminary Computational Experiments

Initial set of instances generated by hand:

- Number of machines $m = 5$ and $C_i = (3, 5, 11, 4, 2)$
- Number of jobs $n \in \{10, 15, 20, 25, 30, 35, 40, 50, 60\}$
- Resource consumption $c_i = (1, 2, 3, 2, 1)$
- $p_{ij} = 1$ and time horizon $T = 30$

Solution methods:

- Default CPLEX 12.2 with typical x_{ijt} formulation
- B&C with s_{ij} variables, classed f and g cuts, and 2 branching strategies:
  - EFC = Earliest violation, first candidate job
  - ESDFF = Earliest viol., smallest s_{ij} domain, fractional s_{ij} first

Preliminary Computational Experiments

Initial set of instances generated by hand:

- Number of machines $m = 5$ and $C_i = (3, 5, 11, 4, 2)$
- Number of jobs $n \in \{10, 15, 20, 25, 30, 35, 40, 50, 60\}$
- Resource consumption $c_i = (1, 2, 3, 2, 1)$
- $p_{ij} = 1$ and time horizon $T = 30$

Solution methods:

- Default CPLEX 12.2 with typical x_{ijt} formulation
- B&C with s_{ij} variables, classed f and g cuts, and 2 branching strategies:
  - EFC = Earliest violation, first candidate job
  - ESDFF = Earliest viol., smallest s_{ij} domain, fractional s_{ij} first
- Latest violation seems to perform worse than earliest

n	Optimal Solution	CPLEX Default			CPLEX 1-Thread		
		Best	Nodes	Time	Best	Nodes	Time
10	54		1	0.1		1	0.1
15	98		4K	57		733	7
20	154		12K	122		3K	26
25	225		51K	4K		17K	817
30	306		83K	11K		233K	79K
35	402	414	131K	2K^m	405	161K	43K^m
40	508	525	91K	4K^m	516	108K	8K^m
50	760*	791	114K	3K^m	784	89K	15K^m
60	1,062*	*no sol.*	90K	13K^m	*no sol.*	67K	44K^m

n	Optimal Solution	CPLEX Default Best	CPLEX Default Nodes	CPLEX Default Time	CPLEX 1-Thread Best	CPLEX 1-Thread Nodes	CPLEX 1-Thread Time
10	54		1	0.1		1	0.1
15	98		4K	57		733	7
20	154		12K	122		3K	26
25	225		51K	4K		17K	817
30	306		83K	11K		233K	79K
35	402	414	131K	2K^{m}	405	161K	43K^{m}
40	508	525	91K	4K^{m}	516	108K	8K^{m}
50	760*	791	114K	3K^{m}	784	89K	15K^{m}
60	1,062*	*no sol.*	90K	13K^{m}	*no sol.*	67K	44K^{m}

n	EFC Best	EFC Nodes	EFC Time	ESDFF Best	ESDFF Nodes	ESDFF Time
10		15	0.1		15	0.1
15		393K	2K		10K	31
20		221	1		8K	30
25		119K	25K		1K	14
30		2K	32		11K	392
35		38K	26K		21K	6K
40		14K	724		7K	395
50	762	159K	24h^{t}	762	77K	24h^{t}
60	1,097	66K	24h^{t}	1108	68K	24h^{t}

Preliminary Computational Experiments (cont.)

Modified job-shop instances from OR-Library

- 81 instances taken from 6 articles from 1988 to 1992
- Number of jobs $n \in \{10, 15, 20, 30, 50\}$
- Number of machines $m \in \{5, 10, 15, 20\}$
- Modifications:
  - $p_{ij} = 1$
  - $C = 6$ for all machines
  - c_i can be 2 or 3 (randomly chosen)

Preliminary Computational Experiments (cont.)

Modified job-shop instances from OR-Library

- 81 instances taken from 6 articles from 1988 to 1992
- Number of jobs $n \in \{10, 15, 20, 30, 50\}$
- Number of machines $m \in \{5, 10, 15, 20\}$
- Modifications:
  - $p_{ij} = 1$
  - $C = 6$ for all machines
  - c_i can be 2 or 3 (randomly chosen)

Solution methods:

- Default CPLEX 12.2 with typical x_{ijt} formulation

Preliminary Computational Experiments (cont.)

Modified job-shop instances from OR-Library

- 81 instances taken from 6 articles from 1988 to 1992
- Number of jobs $n \in \{10, 15, 20, 30, 50\}$
- Number of machines $m \in \{5, 10, 15, 20\}$
- Modifications:
  - $p_{ij} = 1$
  - $C = 6$ for all machines
  - c_i can be 2 or 3 (randomly chosen)

Solution methods:

- Default CPLEX 12.2 with typical x_{ijt} formulation
- B&C with s_{ij} variables, classed f and g cuts, and 4 branching strategies:

Preliminary Computational Experiments (cont.)

Modified job-shop instances from OR-Library

- 81 instances taken from 6 articles from 1988 to 1992
- Number of jobs $n \in \{10, 15, 20, 30, 50\}$
- Number of machines $m \in \{5, 10, 15, 20\}$
- Modifications:
  - $p_{ij} = 1$
  - $C = 6$ for all machines
  - c_i can be 2 or 3 (randomly chosen)

Solution methods:

- Default CPLEX 12.2 with typical x_{ijt} formulation
- B&C with s_{ij} variables, classed f and g cuts, and 4 branching strategies:
  - EFC = Earliest violation, first candidate job
  - ESDFF = Earliest viol., smallest s_{ij} domain, fractional s_{ij} first
  - ESDFL = Earliest viol., smallest s_{ij} domain, fractional s_{ij} last
  - SODE = Smallest overall s_{ij} domain, earliest viol. first if tied

Instances with $n = 10$: 5 with $m = 5$, 18 with $m = 10$

Instances with $n = 10$: 5 with $m = 5$, 18 with $m = 10$

All methods found all 23 optimal solutions

Instances with $n = 10$: 5 with $m = 5$, 18 with $m = 10$

All methods found all 23 optimal solutions

	CPLEX		B&C best		B&C avg.	
# inst.	Nodes	Time	Nodes	Time	Nodes	Time

Instances with $n = 10$: 5 with $m = 5$, 18 with $m = 10$

All methods found all 23 optimal solutions

	# inst.	CPLEX Nodes	CPLEX Time	B&C best Nodes	B&C best Time	B&C avg. Nodes	B&C avg. Time
$m = 5$	4	1	0.10	37	0.08	same	same
	1	1	0.09	1,079	1.7	same	same

Instances with $n = 10$: 5 with $m = 5$, 18 with $m = 10$

All methods found all 23 optimal solutions

	# inst.	CPLEX Nodes	CPLEX Time	B&C best Nodes	B&C best Time	B&C avg. Nodes	B&C avg. Time
$m = 5$	4	1	0.10	37	0.08	same	same
	1	1	0.09	1,079	1.7	same	same
$m = 10$	16	32	0.52	82	0.25	97	0.28
	2	423	6.70	57,163	140	81,951	212

Instances with $n = 15$: 5 for each $m = 5$, 10, and 15

Instances with $n = 15$: 5 for each $m = 5$, 10, and 15

CPLEX found all 15 optimal solutions
B&C methods found all 15, but didn't prove one of them

Instances with $n = 15$: 5 for each $m = 5$, 10, and 15

CPLEX found all 15 optimal solutions
B&C methods found all 15, but didn't prove one of them

	CPLEX		B&C best		B&C avg.	
# inst.	Nodes	Time	Nodes	Time	Nodes	Time

Instances with $n = 15$: 5 for each $m = 5$, 10, and 15

CPLEX found all 15 optimal solutions

B&C methods found all 15, but didn't prove one of them

		CPLEX		B&C best		B&C avg.	
	# inst.	Nodes	Time	Nodes	Time	Nodes	Time
$m = 5$	4	243	1.05	189	0.56	270	0.76
	1	41	0.77	146,162	388	166,955	446

Instances with $n = 15$: 5 for each $m = 5$, 10, and 15

CPLEX found all 15 optimal solutions
B&C methods found all 15, but didn't prove one of them

		CPLEX		B&C best		B&C avg.	
	# inst.	Nodes	Time	Nodes	Time	Nodes	Time
$m = 5$	4	243	1.05	189	0.56	270	0.76
	1	41	0.77	146,162	388	166,955	446
$m = 10$	4	310	2.89	639	2.52	676	2.64
	1	600	2.95	508,659^m		509,876^m	

Instances with $n = 15$: 5 for each $m = 5$, 10, and 15

CPLEX found all 15 optimal solutions
B&C methods found all 15, but didn't prove one of them

	# inst.	CPLEX Nodes	CPLEX Time	B&C best Nodes	B&C best Time	B&C avg. Nodes	B&C avg. Time
$m = 5$	4	243	1.05	189	0.56	270	0.76
	1	41	0.77	146,162	388	166,955	446
$m = 10$	4	310	2.89	639	2.52	676	2.64
	1	600	2.95	508,659^m		509,876^m	
$m = 15$	4	194	2.51	208	1.21	211	1.22
	1	150	1.94	73,346	366	75,161	375

Instances with $n = 20$: $m \in \{5, 10, 15, 20\}$, 28 total

Instances with $n = 20$: $m \in \{5, 10, 15, 20\}$, 28 total

No method ran out of time; always found opt. or feasible solutions

Instances with $n = 20$: $m \in \{5, 10, 15, 20\}$, 28 total

No method ran out of time; always found opt. or feasible solutions

# Instances	$m = 5$	$m = 10$	$m = 15$	$m = 20$	Total
Tied					
CPLEX better					
ESDFF better					
Total					**28**

Instances with $n = 20$: $m \in \{5, 10, 15, 20\}$, 28 total

No method ran out of time; always found opt. or feasible solutions

# Instances	$m = 5$	$m = 10$	$m = 15$	$m = 20$	**Total**
Tied			1[a]		1
CPLEX better					
ESDFF better					
Total					**28**

a: neither method found optimum

Instances with $n = 20$: $m \in \{5, 10, 15, 20\}$, 28 total

No method ran out of time; always found opt. or feasible solutions

# Instances	$m = 5$	$m = 10$	$m = 15$	$m = 20$	**Total**
Tied			1[a]		1
CPLEX better	3[b]				
ESDFF better					
Total					**28**

a: neither method found optimum

b: one 20x faster; one optimum ESDFF didn't find

Instances with $n = 20$: $m \in \{5, 10, 15, 20\}$, 28 total

No method ran out of time; always found opt. or feasible solutions

# Instances	$m = 5$	$m = 10$	$m = 15$	$m = 20$	**Total**
Tied			1[a]		1
CPLEX better	3[b]	10[c]			
ESDFF better					
Total		10			**28**

a: neither method found optimum

b: one 20x faster; one optimum ESDFF didn't find

c: one non-optimal: loses to EFC: value of 269 vs. 266

Instances with $n = 20$: $m \in \{5, 10, 15, 20\}$, 28 total

No method ran out of time; always found opt. or feasible solutions

# Instances	$m = 5$	$m = 10$	$m = 15$	$m = 20$	**Total**
Tied			1[a]		1
CPLEX better	3[b]	10[c]	4[d]		
ESDFF better					
Total		10			**28**

a: neither method found optimum

b: one 20x faster; one optimum ESDFF didn't find

c: one non-optimal: loses to EFC: value of 269 vs. 266

d: one 3x faster; one optimum found by neither

Instances with $n = 20$: $m \in \{5, 10, 15, 20\}$, 28 total

No method ran out of time; always found opt. or feasible solutions

# Instances	$m = 5$	$m = 10$	$m = 15$	$m = 20$	**Total**
Tied			1^a		1
CPLEX better	3^b	10^c	4^d	1	18
ESDFF better					
Total		10			**28**

a: neither method found optimum

b: one 20x faster; one optimum ESDFF didn't find

c: one non-optimal: loses to EFC: value of 269 vs. 266

d: one 3x faster; one optimum found by neither

Instances with $n = 20$: $m \in \{5, 10, 15, 20\}$, 28 total

No method ran out of time; always found opt. or feasible solutions

# Instances	$m = 5$	$m = 10$	$m = 15$	$m = 20$	Total
Tied			1[a]		1
CPLEX better	3[b]	10[c]	4[d]	1	18
ESDFF better	3[e]				
Total	6	10			**28**

a: neither method found optimum

b: one 20x faster; one optimum ESDFF didn't find

c: one non-optimal: loses to EFC: value of 269 vs. 266

d: one 3x faster; one optimum found by neither

e: two 10x faster

Instances with $n = 20$: $m \in \{5, 10, 15, 20\}$, 28 total

No method ran out of time; always found opt. or feasible solutions

# Instances	$m = 5$	$m = 10$	$m = 15$	$m = 20$	**Total**
Tied			1^{a}		1
CPLEX better	3^{b}	10^{c}	4^{d}	1	18
ESDFF better	3^{e}		3^{f}		
Total	6	10	8		**28**

a: neither method found optimum

b: one 20x faster; one optimum ESDFF didn't find

c: one non-optimal: loses to EFC: value of 269 vs. 266

d: one 3x faster; one optimum found by neither

e: two 10x faster

f: two optima found by neither

Instances with $n = 20$: $m \in \{5, 10, 15, 20\}$, 28 total

No method ran out of time; always found opt. or feasible solutions

# Instances	$m = 5$	$m = 10$	$m = 15$	$m = 20$	**Total**
Tied			1^{a}		1
CPLEX better	3^{b}	10^{c}	4^{d}	1	18
ESDFF better	3^{e}		3^{f}	3^{g}	9
Total	6	10	8	4	**28**

a: neither method found optimum

b: one 20x faster; one optimum ESDFF didn't find

c: one non-optimal: loses to EFC: value of 269 vs. 266

d: one 3x faster; one optimum found by neither

e: two 10x faster

f: two optima found by neither

g: all 2x faster

Instances with $n = 30$, $m = 10$

Originally: S. Lawrence, *GSIA Tech. Report*, 1984

Instances with $n = 30$, $m = 10$

Originally: S. Lawrence, *GSIA Tech. Report*, 1984

	CPLEX			EFC		ESDFF		ESDFL		SODE	
Inst.	Best	Nodes	Time	Best	Nodes	Best	Nodes	Best	Nodes	Best	Nodes
la31	411	132K	24h	426	192K	428	203K	428	189K	-	191K
la32	402*	130K	38K	418	196K	415	194K	425	179K	409	190K
la33	410	132K^m	45K	411	192K	407	186K	407	351K	412	189K
la34	409	138K^m	76K	401	193K	399	534K	404	190K	400	195K
la35	392*	124K	52K	419	191K	419	188K	420	192K	420	192K

All four B&C algorithms ran out of memory (4GB)

Instances with $n = 50$, $m = 10$

Originally: Storer, Wu, and Vaccari, *Management Science*, 1992

Instances with $n = 50$, $m = 10$

Originally: Storer, Wu, and Vaccari, *Management Science*, 1992

	CPLEX			ESDFF			
Inst.	Best	Nodes	Time	Best	Nodes	Time	TTFF
swv11	-	58K^m	7,015	1115 (10.3%)	93K	24h	43
swv12	-	59K^m	8,565	1206 (16.25%)	91K	24h	41
swv13	-	54K^m	16,530	1108 (9.84%)	98K	24h	15
swv14	-	55K^m	12,510	1119 (11.53%)	74K	24h	265
swv15	-	62K^m	2,904	1118 (10.29%)	91K	24h	45
swv16	-	61K^m	2,734	-	140K^m		-
swv17	-	54K^m	16,961	967 (8.79%)	116K^m		51
swv18	-	56K^m	12,082	960 (6.25%)	134K^m		16,759
swv19	-	51K^m	55,676	940 (5.74%)	98K^m		288
swv20	-	51K^m	38,033	934 (6.10%)	136K^m		1,402

TTFF $=$ time till first feasible solution (in seconds)

Conclusions and Future Work

- Complete characterization of simple cumulative polytopes

Conclusions and Future Work

- Complete characterization of simple cumulative polytopes
- Inequalities remain valid (though not strong) for general case

Conclusions and Future Work

- Complete characterization of simple cumulative polytopes
- Inequalities remain valid (though not strong) for general case
  - use $r_0 = \min r_j$, $p_0 = \min p_j$, $c_0 = \min c_j$ in $f(K)$

Conclusions and Future Work

- Complete characterization of simple cumulative polytopes
- Inequalities remain valid (though not strong) for general case
  - use $r_0 = \min r_j$, $p_0 = \min p_j$, $c_0 = \min c_j$ in $f(K)$
- Facet-defining inequalities for arbitrary c_j (untested)

Conclusions and Future Work

- Complete characterization of simple cumulative polytopes
- Inequalities remain valid (though not strong) for general case
  - use $r_0 = \min r_j$, $p_0 = \min p_j$, $c_0 = \min c_j$ in $f(K)$
- Facet-defining inequalities for arbitrary c_j (untested)
- Preliminary experiments show some promise

Conclusions and Future Work

- Complete characterization of simple cumulative polytopes
- Inequalities remain valid (though not strong) for general case
  - use $r_0 = \min r_j$, $p_0 = \min p_j$, $c_0 = \min c_j$ in $f(K)$
- Facet-defining inequalities for arbitrary c_j (untested)
- Preliminary experiments show some promise
  - especially as n increases

Conclusions and Future Work

- Complete characterization of simple cumulative polytopes
- Inequalities remain valid (though not strong) for general case
  - use $r_0 = \min r_j$, $p_0 = \min p_j$, $c_0 = \min c_j$ in $f(K)$
- Facet-defining inequalities for arbitrary c_j (untested)
- Preliminary experiments show some promise
  - especially as n increases

- To do: find heuristic separation for arbitrary-c_j facets

Conclusions and Future Work

- Complete characterization of simple cumulative polytopes
- Inequalities remain valid (though not strong) for general case
  - use $r_0 = \min r_j$, $p_0 = \min p_j$, $c_0 = \min c_j$ in $f(K)$
- Facet-defining inequalities for arbitrary c_j (untested)
- Preliminary experiments show some promise
  - especially as n increases

- To do: find heuristic separation for arbitrary-c_j facets
- To do: continue with cumulative job-shop experiments

Conclusions and Future Work

- Complete characterization of simple cumulative polytopes
- Inequalities remain valid (though not strong) for general case
  - use $r_0 = \min r_j$, $p_0 = \min p_j$, $c_0 = \min c_j$ in $f(K)$
- Facet-defining inequalities for arbitrary c_j (untested)
- Preliminary experiments show some promise
  - especially as n increases

- To do: find heuristic separation for arbitrary-c_j facets
- To do: continue with cumulative job-shop experiments
- To do: find other applications with multiple cumulative constraints

Conclusions and Future Work

- Complete characterization of simple cumulative polytopes
- Inequalities remain valid (though not strong) for general case
  - use $r_0 = \min r_j$, $p_0 = \min p_j$, $c_0 = \min c_j$ in $f(K)$
- Facet-defining inequalities for arbitrary c_j (untested)
- Preliminary experiments show some promise
  - especially as n increases

- To do: find heuristic separation for arbitrary-c_j facets
- To do: continue with cumulative job-shop experiments
- To do: find other applications with multiple cumulative constraints
- To do: allow other parameters to differ: r_j, p_j

The End

Thank You!

Any Questions?