On the performance of Smith's rule in single-machine scheduling with nonlinear cost

Wiebke Höhn
Technische Universität Berlin

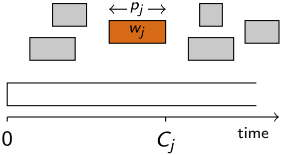
Tobias Jacobs
NEC Laboratories Europe

18th Combinatorial Optimization Workshop
Aussois 2014

Generalized min-sum scheduling

Given: jobs $j = 1, \ldots, n$ with

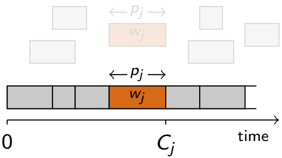
- weight $w_j > 0$
- processing time $p_j > 0$



Generalized min-sum scheduling

Given: jobs $j = 1, \ldots, n$ with

- weight $w_j > 0$
- processing time $p_j > 0$



Task: compute sequence with minimum cost $\sum_j w_j f(C_j)$

- C_j completion time of job j
- non-decreasing, non-negative cost function f

Motivation

- priorities and fairness
  $\rightsquigarrow$ L_k-norms/monomials compromise on worst and average case

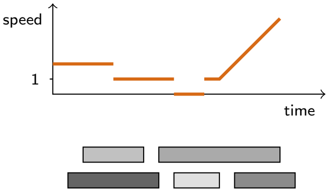
Motivation

- priorities and fairness
  $\rightsquigarrow$ L_k-norms/monomials compromise on worst and average case
- linear cost $\sum_{\mathrm{j}} w_j C_j^{(s)}$ but non-uniform speed s

Motivation

- priorities and fairness
  $\rightsquigarrow$ L_k-norms/monomials compromise on worst and average case

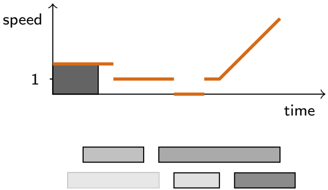
- linear cost $\sum_j w_j C_j^{(s)}$ but non-uniform speed s



Motivation

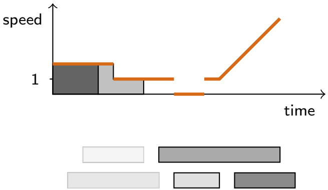
- priorities and fairness
  $\rightsquigarrow$ L_k-norms/monomials compromise on worst and average case

- linear cost $\sum_j w_j C_j^{(s)}$ but non-uniform speed s



Motivation

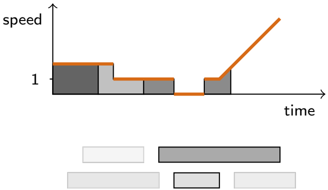
- priorities and fairness
  $\rightsquigarrow$ L_k-norms/monomials compromise on worst and average case
- linear cost $\sum_j w_j C_j^{(s)}$ but non-uniform speed s



Motivation

- priorities and fairness
  $\rightsquigarrow$ L_k-norms/monomials compromise on worst and average case

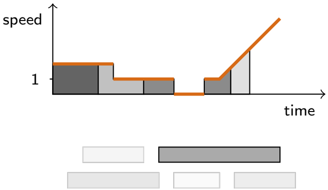
- linear cost $\sum_j w_j C_j^{(s)}$ but non-uniform speed s



Motivation

- priorities and fairness
  $\rightsquigarrow$ L_k-norms/monomials compromise on worst and average case

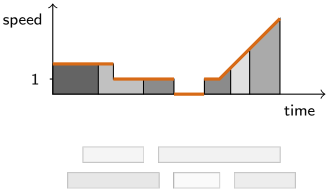
- linear cost $\sum_j w_j C_j^{(s)}$ but non-uniform speed s



Motivation

- priorities and fairness
  $\rightsquigarrow$ L_k-norms/monomials compromise on worst and average case

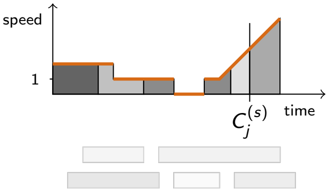
- linear cost $\sum_j w_j C_j^{(s)}$ but non-uniform speed s



Motivation

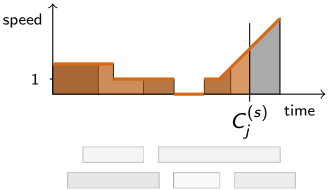
- priorities and fairness
  $\rightsquigarrow$ L_k-norms/monomials compromise on worst and average case

- linear cost $\sum_j w_j C_j^{(s)}$ but non-uniform speed s



Motivation

- priorities and fairness
  $\rightsquigarrow$ L_k-norms/monomials compromise on worst and average case
- linear cost $\sum_j w_j C_j^{(s)}$ but non-uniform speed s

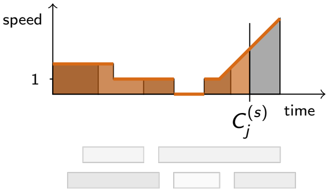


$$\int_0^{C_j^{(s)}} s(t)\,dt = \sum_{i \le j} p_i$$

Motivation

- priorities and fairness
  $\rightsquigarrow$ L_k-norms/monomials compromise on worst and average case

- linear cost $\sum_j w_j C_j^{(s)}$ but non-uniform speed s

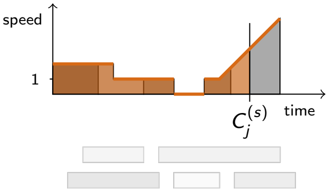


$$\int_0^{C_j^{(s)}} s(t)\,dt = \sum_{i \le j} p_j = C_j^{(1)}$$

Motivation

- priorities and fairness
  $\rightsquigarrow$ L_k-norms/monomials compromise on worst and average case

- linear cost $\sum_j w_j C_j^{(s)}$ but non-uniform speed s

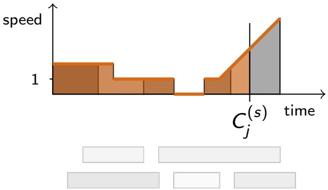


$$S\left(C_j^{(s)}\right) := \int_0^{C_j^{(s)}} s(t)\,dt = \sum_{i \le j} p_j = C_j^{(1)}$$

Motivation

- priorities and fairness
  $\rightsquigarrow$ L_k-norms/monomials compromise on worst and average case

- linear cost $\sum_j w_j C_j^{(s)}$ but non-uniform speed s

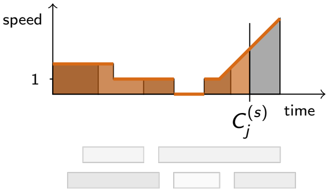


$$S\left(C_j^{(s)}\right) := \int_0^{C_j^{(s)}} s(t)\,dt = \sum_{i \leq j} p_j = C_j^{(1)}$$

$$\Leftrightarrow C_j^{(s)} = S^{-1}\left(C_j^{(1)}\right)$$

- priorities and fairness
  $\rightsquigarrow$ L_k-norms/monomials compromise on worst and average case

- linear cost $\sum_j w_j C_j^{(s)}$ but non-uniform speed s



$$S\left(C_j^{(s)}\right) := \int_0^{C_j^{(s)}} s(t)\,dt = \sum_{i \le j} p_j = C_j^{(1)}$$

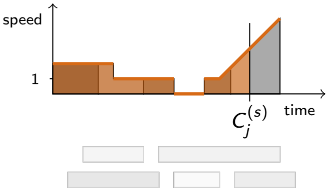
$$\Leftrightarrow C_j^{(s)} = S^{-1}\left(C_j^{(1)}\right)$$

increasing speed s $\leftrightarrow$ concave cost f
decreasing speed s $\leftrightarrow$ convex cost f

Motivation

- priorities and fairness
  $\rightsquigarrow$ L_k-norms/monomials compromise on worst and average case

- linear cost $\sum_j w_j C_j^{(s)}$ but non-uniform speed s



$$S\left(C_j^{(s)}\right) := \int_0^{C_j^{(s)}} s(t)\,dt = \sum_{i \leq j} p_j = C_j^{(1)}$$

$$\Leftrightarrow C_j^{(s)} = S^{-1}\left(C_j^{(1)}\right)$$

increasing speed s $\leftrightarrow$ concave cost f
decreasing speed s $\leftrightarrow$ convex cost f

Our main focus: convex / concave cost functions

Outline

1. Analysis of Smith's rule for convex (and concave) cost

2. Exact algorithms for monomials

Related work & complexity status

linear	in P [Smith 1956]

Related work & complexity status

linear	in P [Smith 1956]
exponential	in P [Rothkopf 1966]

Related work & complexity status

linear	in P [Smith 1956]	
exponential	in P [Rothkopf 1966]	
general	PTAS [Megow, Verschae 2012]	strongly NP-hard [H., Jacobs 2012]

Related work & complexity status

linear	in P [Smith 1956]	
exponential	in P [Rothkopf 1966]	
general / piece-wise linear	PTAS [Megow, Verschae 2012]	strongly NP-hard [H., Jacobs 2012]

Related work & complexity status

linear	in P [Smith 1956]	
exponential	in P [Rothkopf 1966]	
general	PTAS [Megow, Verschae 2012]	strongly NP-hard [H., Jacobs 2012]
piece-wise linear		
convex		weakly NP-hard [Yuan '92]

Related work & complexity status

linear	in P [Smith 1956]	
exponential	in P [Rothkopf 1966]	
general	PTAS [Megow, Verschae 2012]	strongly NP-hard [H., Jacobs 2012]
piece-wise linear	PTAS [Megow, Verschae 2012]	strongly NP-hard [H., Jacobs 2012]
convex	FPTAS ?	weakly NP-hard [Yuan '92] strongly NP-hard ?

Related work & complexity status

linear	in P [Smith 1956]	
exponential	in P [Rothkopf 1966]	
general	PTAS [Megow, Verschae 2012]	strongly NP-hard [H., Jacobs 2012]
piece-wise linear		
convex	FPTAS ?	weakly NP-hard [Yuan '92] strongly NP-hard ?
concave	in P / FPTAS ?	(strongly) NP-hard ?

Related work & complexity status

linear	in P [Smith 1956]		
exponential	in P [Rothkopf 1966]		
general	PTAS [Megow, Verschae 2012]		strongly NP-hard [H., Jacobs 2012]
piece-wise linear			
convex		FPTAS ?	weakly NP-hard [Yuan '92] strongly NP-hard ?
concave		in P / FPTAS ?	(strongly) NP-hard ?
monomials t^k		in P / FPTAS ?	(strongly) NP-hard ?

Related work & complexity status

linear	in P [Smith 1956]	
exponential	in P [Rothkopf 1966]	
general	PTAS [Megow, Verschae 2012]	strongly NP-hard [H., Jacobs 2012]
piece-wise linear		
convex	FPTAS ?	weakly NP-hard [Yuan '92] strongly NP-hard ?
concave	in P / FPTAS ?	(strongly) NP-hard ?
monomials t^k	in P / FPTAS ?	(strongly) NP-hard ?
piece-wise linear, const. # pieces	FPTAS [Megow, Verschae '12]	weakly NP-hard [Yuan '92]

Analysis of Smith's rule

Smith's rule

Schedule jobs in non-increasing order of their density $\frac{w_j}{p_j}$.

Analysis of Smith's rule

Smith's rule

Schedule jobs in non-increasing order of their density $\frac{w_j}{p_j}$.

How good is this simple algorithm for a fixed convex/concave cost function?

Analysis of Smith's rule

Smith's rule

Schedule jobs in non-increasing order of their density $\frac{w_j}{p_j}$.

How good is this simple algorithm for a fixed convex/concave cost function?

- Smith's rule is a $\frac{\sqrt{3}+1}{2}$-approximation for any concave cost function f [Stiller & Wiese '10]

Analysis of Smith's rule

Smith's rule

Schedule jobs in non-increasing order of their density $\frac{w_j}{p_j}$.

How good is this simple algorithm for a fixed convex/concave cost function?

- Smith's rule is a $\frac{\sqrt{3}+1}{2}$-approximation for any concave cost function f [Stiller & Wiese '10]

Theorem

The tight approximation ratio of Smith's rule for fixed convex f is

$$\sup_{0<q,p} \frac{\int_0^q f(t)dt + p \cdot f(q+p)}{p \cdot f(p) + \int_p^{p+q} f(t)dt} .$$

Analysis of Smith's rule

Smith's rule

Schedule jobs in non-increasing order of their density $\frac{w_j}{p_j}$.

How good is this simple algorithm for a fixed convex/concave cost function?

- Smith's rule is a $\frac{\sqrt{3}+1}{2}$-approximation for any concave cost function f [Stiller & Wiese '10]

Theorem

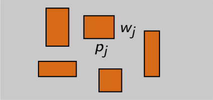
The tight approximation ratio of Smith's rule for fixed convex f is

$$\sup_{0<q,p} \frac{\int_0^q f(t)dt + p \cdot f(q+p)}{p \cdot f(p) + \int_p^{p+q} f(t)dt} .$$

$\rightsquigarrow$ holds with inverse ratio for concave cost function

Narrow space of worst-case instances for convex cost:

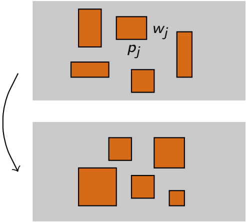
Narrow space of worst-case instances for convex cost:



We can assume w.l.o.g. that:

Analysis of Smith's rule

Narrow space of worst-case instances for convex cost:



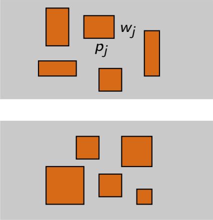
We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j

Analysis of Smith's rule

Narrow space of worst-case instances for convex cost:

merge
density classes
inductively

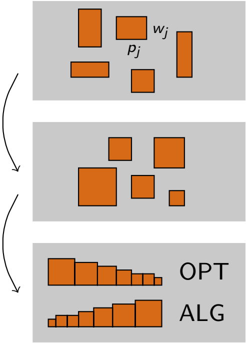


We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j

Narrow space of worst-case instances for convex cost:

merge
density classes
inductively

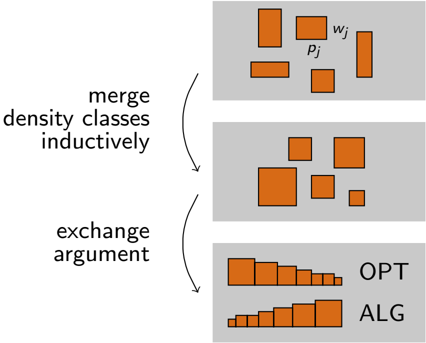


We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite

Analysis of Smith's rule

Narrow space of worst-case instances for convex cost:

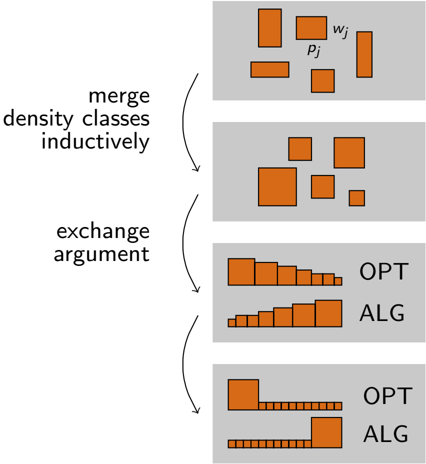


We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite

Analysis of Smith's rule

Narrow space of worst-case instances for convex cost:

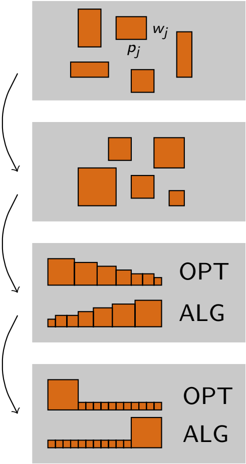


We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite
3. one big job & several very small jobs

Analysis of Smith's rule

Narrow space of worst-case instances for convex cost:

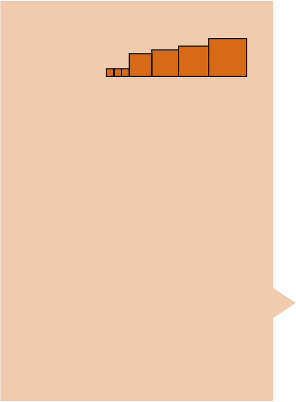


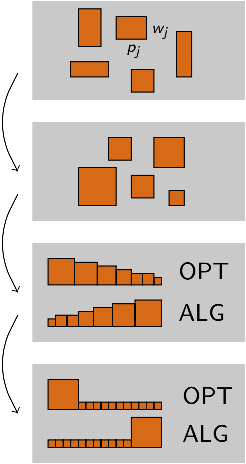
We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite
3. one big job & several very small jobs

Analysis of Smith's rule

Narrow space of worst-case instances for convex cost:

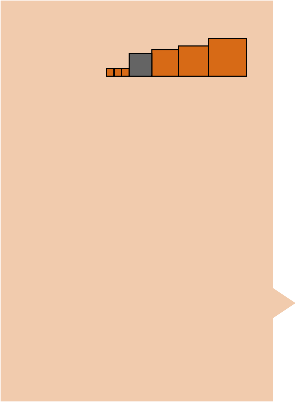


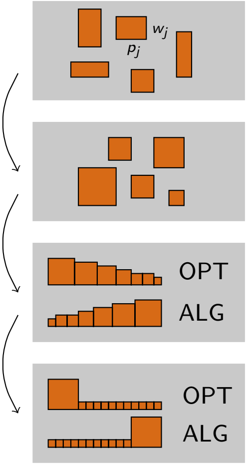


We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite
3. one big job & several very small jobs

Narrow space of worst-case instances for convex cost:



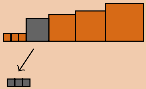


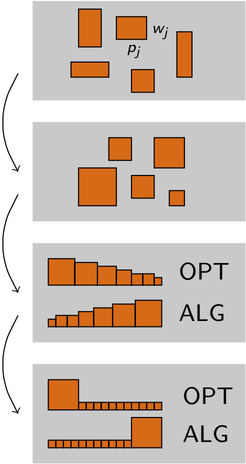
We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite
3. one big job & several very small jobs

Analysis of Smith's rule

Narrow space of worst-case instances for convex cost:



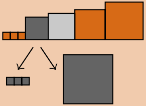


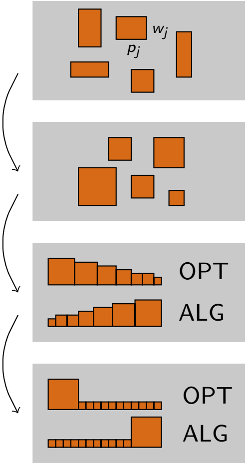
We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite
3. one big job & several very small jobs

Analysis of Smith's rule

Narrow space of worst-case instances for convex cost:



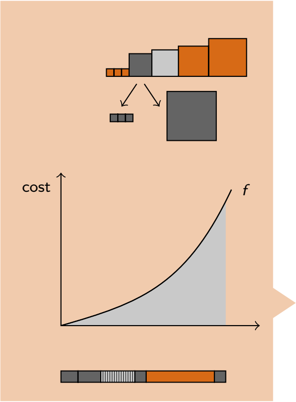


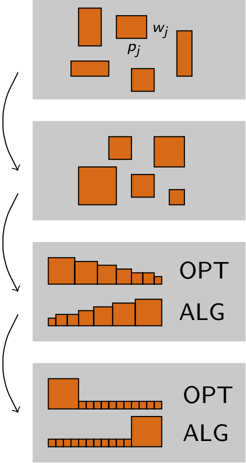
We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite
3. one big job & several very small jobs

Analysis of Smith's rule

Narrow space of worst-case instances for convex cost:



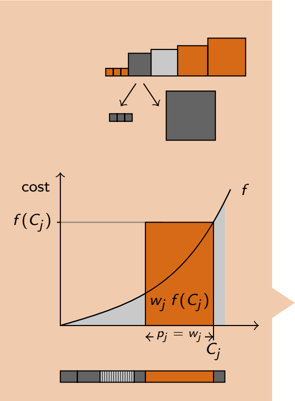


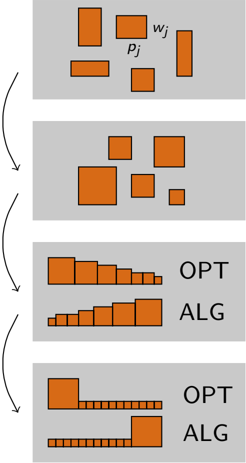
We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite
3. one big job & several very small jobs

Analysis of Smith's rule

Narrow space of worst-case instances for convex cost:



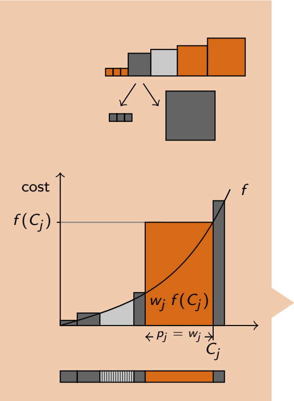


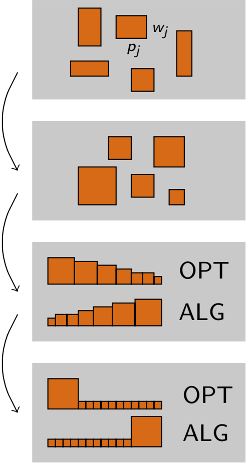
We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite
3. one big job & several very small jobs

Analysis of Smith's rule

Narrow space of worst-case instances for convex cost:



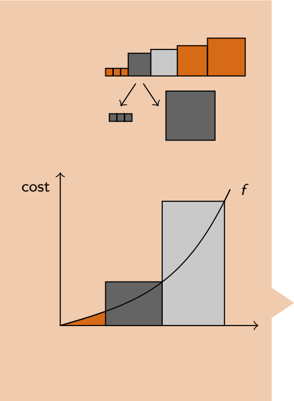


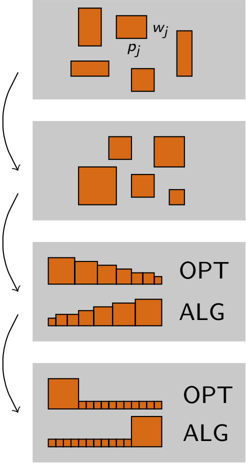
We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite
3. one big job & several very small jobs

Analysis of Smith's rule

Narrow space of worst-case instances for convex cost:



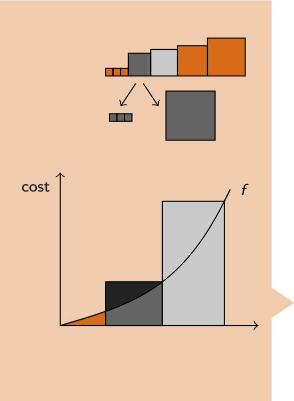


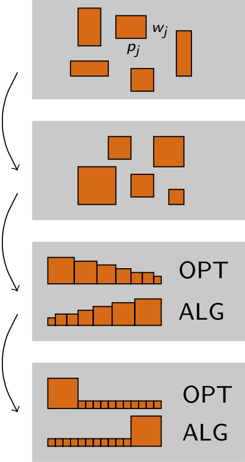
We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite
3. one big job & several very small jobs

Analysis of Smith's rule

Narrow space of worst-case instances for convex cost:

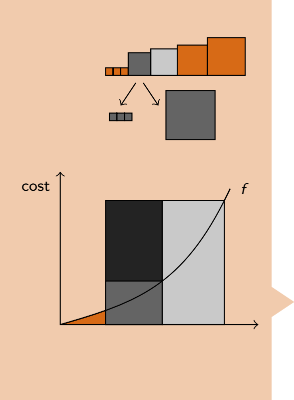


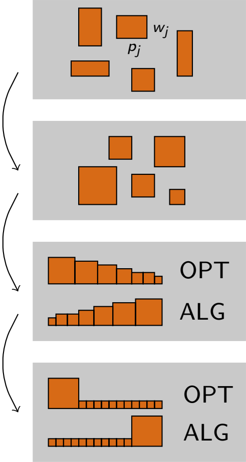


We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite
3. one big job & several very small jobs

Narrow space of worst-case instances for convex cost:

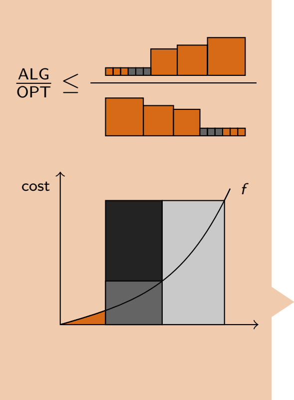


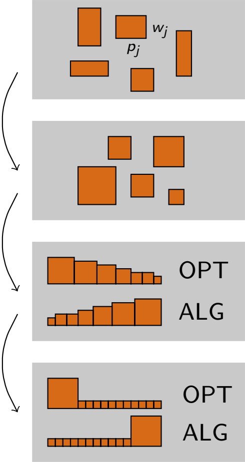


We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite
3. one big job & several very small jobs

Narrow space of worst-case instances for convex cost:



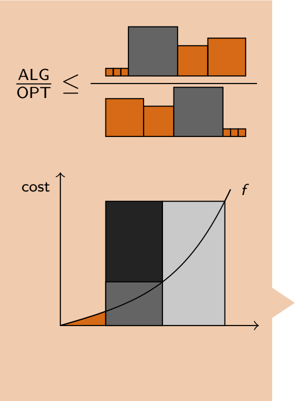


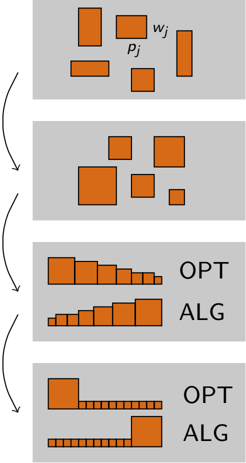
We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite
3. one big job & several very small jobs

Analysis of Smith's rule

Narrow space of worst-case instances for convex cost:



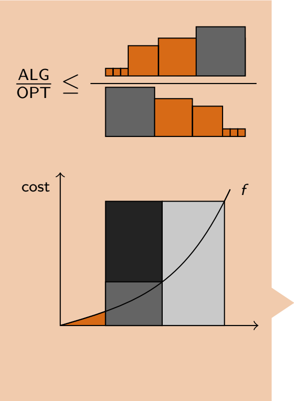


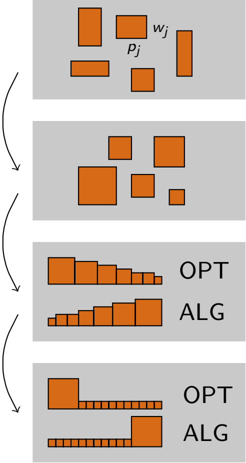
We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite
3. one big job & several very small jobs

Analysis of Smith's rule

Narrow space of worst-case instances for convex cost:





We can assume w.l.o.g. that:

1. $w_j = p_j$ for all jobs j
2. Smith's Rule chooses non-decreasing order, OPT the opposite
3. one big job & several very small jobs

Analysis of Smith's rule

Theorem

The tight approximation ratio of Smith's rule for fixed convex f is

$$\sup_{0<q,p} \frac{\int_0^q f(t)dt + p \cdot f(q+p)}{p \cdot f(p) + \int_p^{p+q} f(t)dt} \; .$$

Analysis of Smith's rule

Theorem

The tight approximation ratio of Smith's rule for fixed convex f is

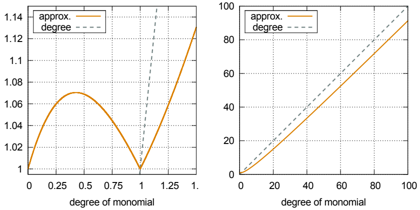
$$\sup_{0<q,p} \frac{\int_0^q f(t)dt + p\cdot f(q+p)}{p\cdot f(p) + \int_p^{p+q} f(t)dt} \ .$$

Corollary

If f is a polynomial of degree k with non-negative coefficients then the tight approximation ratio is

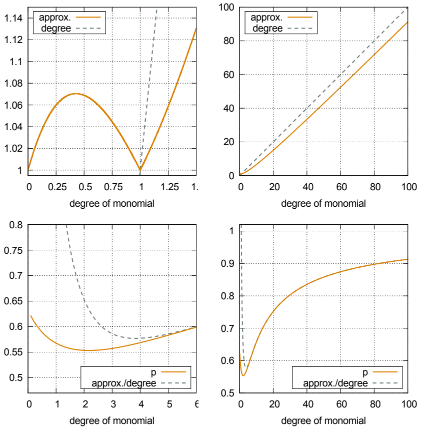
$$\alpha_k \ := \max_{0.5\leq p<1} \frac{(1-p)^{k+1} \ + \ (k+1)p}{kp^{k+1} \ + \ 1} \ .$$

Tight approximation ratios for polynomials



cost function	ratio
square root	1.07
degree 2 polynomials	1.31
degree 3 polynomials	1.76
degree 4 polynomials	2.31
degree 5 polynomials	2.93
degree 6 polynomials	3.60
degree 10 polynomials	6.58
degree 20 polynomials	15.04
exponential	∞

Tight approximation ratios for polynomials



Observation: $\frac{\alpha_k}{k} \approx p_k \quad \left(\begin{array}{c}\text{length of big job}\\ \text{corresponding to } \alpha_k\end{array}\right)$

Bounding the approximation ratio

Theorem

For cost function $f(t) = t^k$, the tight approximation factor α_k of Smith's ruler observes the following for $k \geq 4$:

- $\lim_{k\to\infty} \left(p_k - \sqrt[k+1]{\frac{1}{k^2}} \right) = 0,$

Bounding the approximation ratio

Theorem

For cost function $f(t) = t^k$, the tight approximation factor α_k of Smith's ruler observes the following for $k \geq 4$:

- $\lim_{k \to \infty} \left(p_k - \sqrt[k+1]{\frac{1}{k^2}} \right) = 0,$
- $\lim_{k \to \infty} \left(\alpha_k - k^{\frac{k-1}{k+1}} \right) = 0,$

Bounding the approximation ratio

Theorem

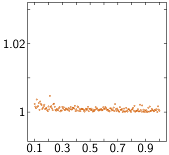
For cost function $f(t) = t^k$, the tight approximation factor α_k of Smith's ruler observes the following for $k \geq 4$:

- $\lim\limits_{k\to\infty} \left(p_k - \sqrt[k+1]{\frac{1}{k^2}} \right) = 0,$
- $\lim\limits_{k\to\infty} \left(\alpha_k - k^{\frac{k-1}{k+1}} \right) = 0,$
- $k - \alpha_k \;\geq\; \ln k - \frac{1}{2k}.$

Related computational results

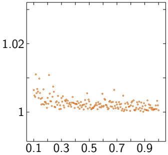
Approximation ratio in experiments:

Approximation ratio in experiments:



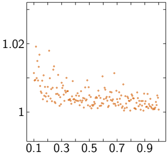
t^2

tight ratio 1.31



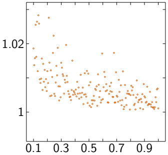
t^3

tight ratio 1.76



t^4

tight ratio 2.31



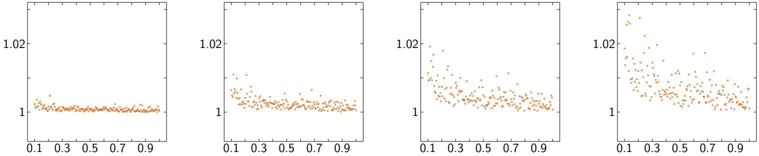
t^5

tight ratio 2.93

x-value $\rightsquigarrow$ correlation of w_j and p_j

Related computational results

Approximation ratio in experiments:



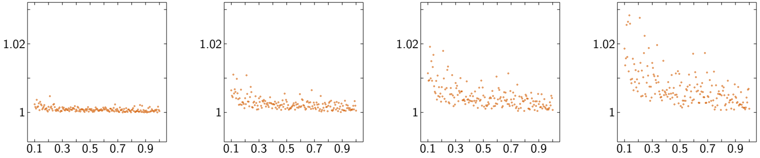
t^2	t^3	t^4	t^5
tight ratio 1.31	tight ratio 1.76	tight ratio 2.31	tight ratio 2.93

x-value $\rightsquigarrow$ correlation of w_j and p_j

$\rightsquigarrow$ experimental performance much better than worst-case

Related computational results

Approximation ratio in experiments:



t^2	t^3	t^4	t^5
tight ratio 1.31	tight ratio 1.76	tight ratio 2.31	tight ratio 2.93

x-value $\rightsquigarrow$ correlation of w_j and p_j

- $\rightsquigarrow$ experimental performance much better than worst-case
- $\rightsquigarrow$ more realistic analysis for processing times $1, 2, \ldots, p_{\max}$ and given $\sum p_j$

Parametrized analysis of Smith's rule

Theorem

The tight approximation ratio of Smith's rule for convex f and fixed parameters $p_{\max}$ and $\sum_j p_j$ is

$$\sup\left\{ \frac{\mathrm{INC}(p, p_{\max}, \sum_j p_j)}{\mathrm{DEC}(p, p_{\max}, \sum_j p_j)} \,\middle|\, p = 0, 1, 2, \ldots, \sum_j p_j \right\}.$$

Parametrized analysis of Smith's rule

Theorem

The tight approximation ratio of Smith's rule for convex f and fixed parameters $p_{\max}$ and $\sum_j p_j$ is

$$\sup\left\{ \frac{\mathrm{INC}(p, p_{\max}, \sum_j p_j)}{\mathrm{DEC}(p, p_{\max}, \sum_j p_j)} \,\middle|\, p = 0, 1, 2, \ldots, \sum_j p_j \right\} .$$

$\rightsquigarrow$ proof follows same idea as unparametrized analysis

Parametrized analysis of Smith's rule

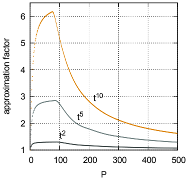
Theorem

The tight approximation ratio of Smith's rule for convex f and fixed parameters $p_{\max}$ and $\sum_j p_j$ is

$$\sup \left\{ \frac{\mathrm{INC}(p, p_{\max}, \sum_j p_j)}{\mathrm{DEC}(p, p_{\max}, \sum_j p_j)} \,\middle|\, p = 0, 1, 2, \ldots, \sum_j p_j \right\} .$$

$\rightsquigarrow$ proof follows same idea as unparametrized analysis

valuable lower bound for exact computations



$p_{\max} = 50$

Exact algorithms for quadratic cost

Approach proposed for quadratic cost:

- best first graph search based on A^* [Sen et al. '96, Kaindl et al. '01]

Exact algorithms for quadratic cost

Approach proposed for quadratic cost:

- best first graph search based on A^* [Sen et al. '96, Kaindl et al. '01]
- *local and global comparability* [Schild, Fredman '62, Sen et al. '90]

Exact algorithms for quadratic cost

Approach proposed for quadratic cost:

- best first graph search based on A^* [Sen et al. '96, Kaindl et al. '01]
- *local and global comparability* [Schild, Fredman '62, Sen et al. '90]
  - **global comparability:**
    not matter where scheduled

Exact algorithms for quadratic cost

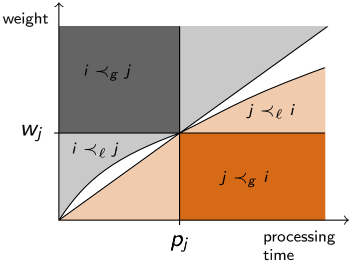
Approach proposed for quadratic cost:

- best first graph search based on A^* [Sen et al. '96, Kaindl et al. '01]
- *local and global comparability* [Schild, Fredman '62, Sen et al. '90]
  - **global comparability:**
    not matter where scheduled
  - **local comparability:**
    only if scheduled consecutively

Exact algorithms for quadratic cost

Approach proposed for quadratic cost:

- best first graph search based on A^* [Sen et al. '96, Kaindl et al. '01]
- *local and global comparability* [Schild, Fredman '62, Sen et al. '90]
  - **global comparability:** not matter where scheduled
  - **local comparability:** only if scheduled consecutively

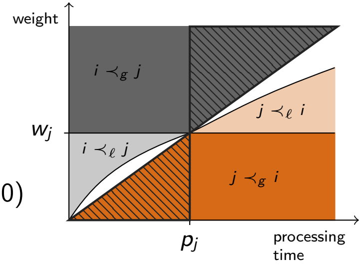


Exact algorithms for quadratic cost

Approach proposed for quadratic cost:

- best first graph search based on A^* [Sen et al. '96, Kaindl et al. '01]
- *local and global comparability* [Schild, Fredman '62, Sen et al. '90]
  - **global comparability:** not matter where scheduled
  - **local comparability:** only if scheduled consecutively

Conjecture Mondal, Sen (2000)

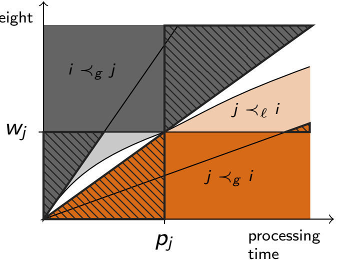


Exact algorithms for quadratic cost

Approach proposed for quadratic cost:

- best first graph search based on A^* [Sen et al. '96, Kaindl et al. '01]
- *local and global comparability* [Schild, Fredman '62, Sen et al. '90]
  - **global comparability:** not matter where scheduled
  - **local comparability:** only if scheduled consecutively

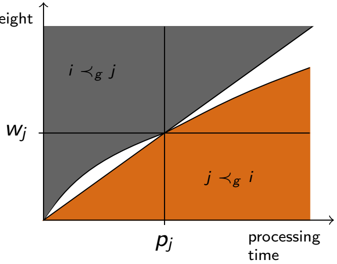
H., Jacobs



Exact algorithms for quadratic cost

Approach proposed for quadratic cost:

- best first graph search based on A^* [Sen et al. '96, Kaindl et al. '01]
- *local and global comparability* [Schild, Fredman '62, Sen et al. '90]
  - **global comparability:** not matter where scheduled
  - **local comparability:** only if scheduled consecutively



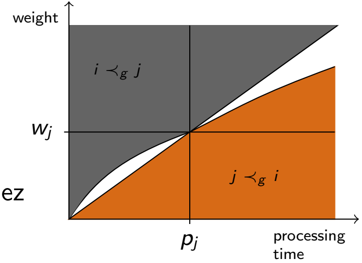
Dürr, Vasquez

Exact algorithms for quadratic cost

Approach proposed for quadratic cost:

- best first graph search based on A^* [Sen et al. '96, Kaindl et al. '01]
- *local and global comparability* [Schild, Fredman '62, Sen et al. '90]
  - **global comparability:** not matter where scheduled
  - **local comparability:** only if scheduled consecutively

Dürr, Vasquez



Approaches tested by us:

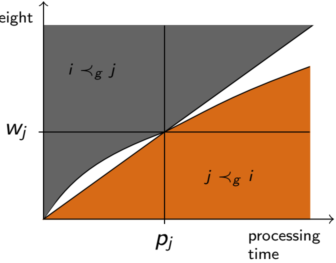
- different graph searches with integrated comparabilities

Exact algorithms for quadratic cost

Approach proposed for quadratic cost:

- best first graph search based on A^* [Sen et al. '96, Kaindl et al. '01]
- *local and global comparability* [Schild, Fredman '62, Sen et al. '90]
  - **global comparability:** not matter where scheduled
  - **local comparability:** only if scheduled consecutively

Dürr, Vasquez



Approaches tested by us:

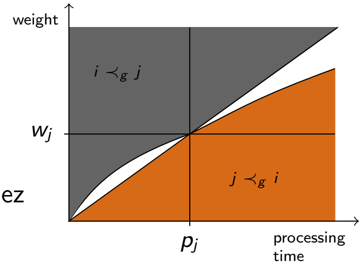
- different graph searches with integrated comparabilities
- quadratic IP with integrated comparabilities (Cplex 12.4)

Exact algorithms for quadratic cost

Approach proposed for quadratic cost:

- best first graph search based on A^* [Sen et al. '96, Kaindl et al. '01]
- *local and global comparability* [Schild, Fredman '62, Sen et al. '90]
  - **global comparability:** not matter where scheduled
  - **local comparability:** only if scheduled consecutively

Dürr, Vasquez



Approaches tested by us:

- different graph searches with integrated comparabilities
- quadratic IP with integrated comparabilities (Cplex 12.4)
  $\rightsquigarrow$ major numerical problems

Exact algorithms for monomial cost

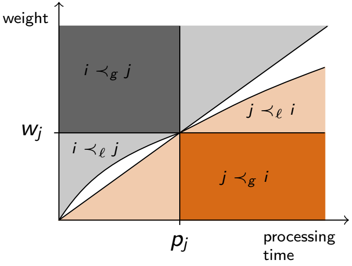
Feasible for monomial cost t^k:

- best first graph search based on A^*

Exact algorithms for monomial cost

Feasible for monomial cost t^k:

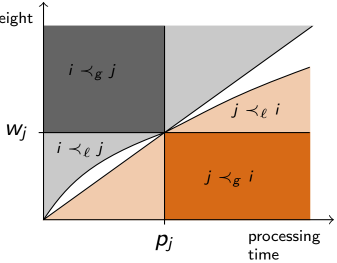
- best first graph search based on A^*
- *local and global comparability*



Exact algorithms for monomial cost

Feasible for monomial cost t^k:

- best first graph search based on A^*
- *local and global comparability*



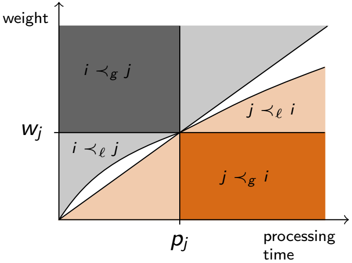
Constraint programming approach:
(joint work with Jens Schulz & Daniela Luft)

- start time based formulations with disjunctive constraint and domain propagation (SCIP 2.1.1)

Exact algorithms for monomial cost

Feasible for monomial cost t^k:

- best first graph search based on A^*
- *local and global comparability*



Constraint programming approach:
(joint work with Jens Schulz & Daniela Luft)

- start time based formulations with disjunctive constraint and domain propagation (SCIP 2.1.1)
  $\rightsquigarrow$ again major numerical problems (for t^2, t^3, t^4)

Conclusions

Single machine scheduling with weighted convex/concave cost:

Conclusions

Single machine scheduling with weighted convex/concave cost:

Approximation algorithms:

- tight (parametrized) analysis of Smith's rule

Conclusions

Single machine scheduling with weighted convex/concave cost:

Approximation algorithms:

- tight (parametrized) analysis of Smith's rule
- asymptotic approximation factor $k^{\frac{k-1}{k+1}}$ for cost t^k

Conclusions

Single machine scheduling with weighted convex/concave cost:

Approximation algorithms:

- tight (parametrized) analysis of Smith's rule
- asymptotic approximation factor $k^{\frac{k-1}{k+1}}$ for cost t^k

Exact algorithms for monomial cost:

- generic solvers have major numerical problems while problem-specific enumeration schemes don't

Conclusions

Single machine scheduling with weighted convex/concave cost:

Approximation algorithms:

- tight (parametrized) analysis of Smith's rule
- asymptotic approximation factor $k^{\frac{k-1}{k+1}}$ for cost t^k

Exact algorithms for monomial cost:

- generic solvers have major numerical problems while problem-specific enumeration schemes don't

$\rightsquigarrow$ complexity (almost) completely open

Conclusions

Single machine scheduling with weighted convex/concave cost:

Approximation algorithms:

- tight (parametrized) analysis of Smith's rule
- asymptotic approximation factor $k^{\frac{k-1}{k+1}}$ for cost t^k

Exact algorithms for monomial cost:

- generic solvers have major numerical problems while problem-specific enumeration schemes don't

$\rightsquigarrow$ complexity (almost) completely open

Thank you!