Approximation algorithms for machine scheduling problems with non-renewable resources

Márton Drótos [a], Péter Györgyi[b], Tamás Kis[ab]

[a]Institute for Computer Science and Control, Budapest

[b]Operations Research Department, Eötvös Loránd University, Budapest

Aussois Comb. Opt. workshop, 2014.01.06-10.

Content

Basic notions

1. Approximation algorihms for optimization problems
   - ρ-approximation algorithm A for a minimization problem Π

     A has polynomial time complexity, and $A(I) \leq \rho OPT(I)$, $\forall I \in \Pi$

   - Polynomial Time Approximation Scheme for Π:
     i) Family of approximation algorithms $\{A_\varepsilon\}_{\varepsilon>0}$
     ii) A_ε is an $(1+\varepsilon)$-approximation algorithm for Π
     iii) A_ε has polynomial time complexity in the size of the input
   - Fully Polynomial Time Approximation Scheme for Π:
     i) Family of approximation algorithms $\{A_\varepsilon\}_{\varepsilon>0}$
     ii) A_ε is an $(1+\varepsilon)$-approximation algorithm for Π
     iii) A_ε has polynomial time complexity in the size of the input and in $1/\varepsilon$
2. class $APX = \{\Pi \mid \ \Pi$ admits a ρ-approx. alg. for some $\rho > 1\}$
3. Unless $P = NP$, *class* $APX \neq$ *class* $PTAS$

Introduction to machine scheduling with non-renewable resources

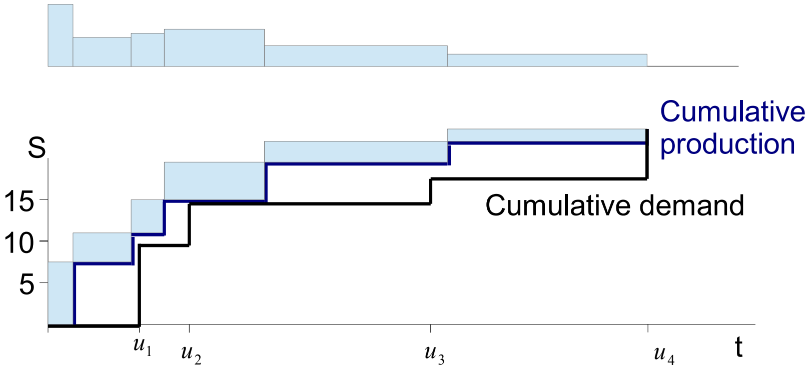
Extend basic machine scheduling problems with non-renewanle resources.

3 cases:

- Only producer jobs
- Only consumer jobs
- Both producer and consumer jobs (not in this talk)

Producer Jobs

Producer jobs



Earlier work on producer jobs

Problem data

- Single machine
- Set of products types S.
- Jobs: J_j, processing time p_j, type $s_j \in S$, and a quantity a_j^s produced from product s_j.
- External demand requests with deadlines $u_1 \leq \cdots \leq u_q$, product types $\rho_\ell \in S$, and quantities requested b_ℓ^ρ.

Results of Boysen et al (2012)

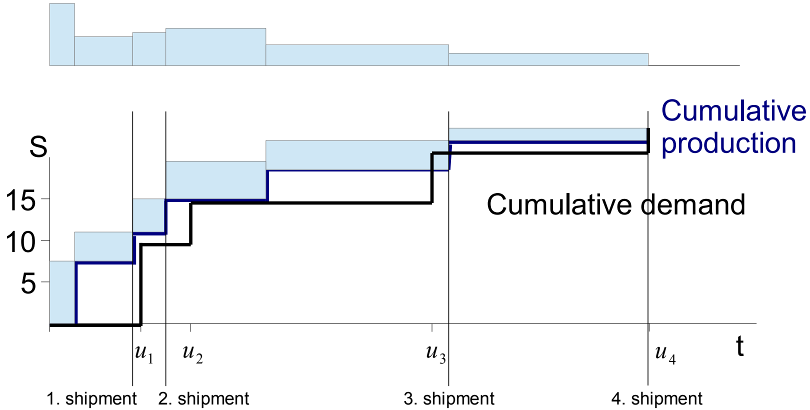
- Deciding whether a feasible schedule exists is NP complete
- Objective: minimize maximum stock size (assuming that we know that a feasible schedule exists)
- Several polynomial time special cases

New results on producer jobs

- Due dates instead of deadlines, shipments may be late
- A single job may produce several types of products
- Objective: minimize a regular function of shipment dates, e.g., tardiness
- Main results:

Problem	Complexity
$1\| \ p_j = p, \|S\| = 1 \mid \gamma$	$O(n \log n)$
$1\| \ a_j^s = a^s \mid \gamma$	$O(n \log n)$
$1\| \ p_j = 1, \|S\| = 2, q = 2 \mid \gamma^T$	ordinary NPh
$1\| \ p_j = 1, \|S\| = 2 \ \|\gamma^T$	strongly NPh
$1\| \ \|S\| = 1\|\gamma^T$	strongly NPh
$1\| \ p_j = 1, q = 2 \mid \gamma^T$	strongly NPh
$1\| \ \|S\| = const, q = const \mid \gamma$	pseudo-polytime
$1\| \ \|S\| = 1, q = 2 \mid p_{sum} + T_{\max}$	FPTAS

Producer jobs



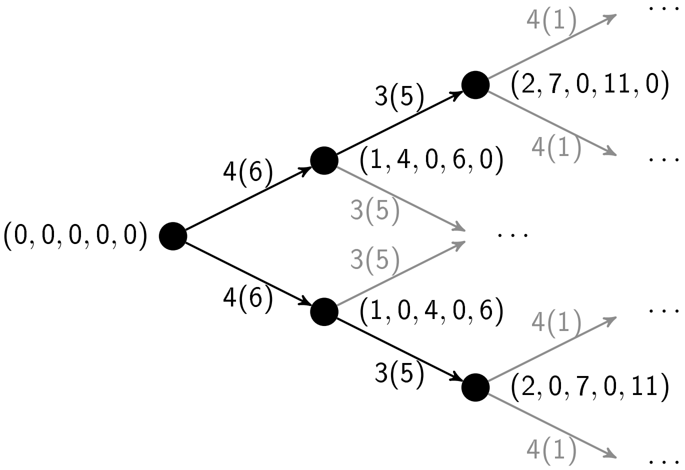
Polynomial time algorithms

1. Solving '$1|\ p_j = p, |S| = 1 \mid \gamma$' in $O(n \log n)$ time
   - Schedule the jobs in non-increasing a_j^s order.
2. Solving '$1|\ a_j^s = a^s \mid \gamma$' in $O(n \log n)$ time
   - Schedule the jobs in non-decreasing processing time order (SPT rule)

Pseudo-polynomial time algorithm for fixed number of resources and deliveries

- Fixed $|S|$ (number of resources), q (number of deliveries)
- time interval between two consecutive deliveries, and between 0 and the first delivery
- Define an acyclic directed graph with $n^J + 1$ levels, $0, 1, \ldots, n$
- Nodes: $N(j, P_1, \ldots, P_q, \Delta_1^1, \ldots, \Delta_q^{n^S})$, where
  - i) j is the job just scheduled
  - ii) P_i is the total processing time of those jobs scheduled in slot i
  - iii) Δ_i^s is the total amount of resource s scheduled in slot i
- Arcs: form $N(j, P_1, \ldots, P_q, \Delta_1^1, \ldots, \Delta_q^{n^S})$ to $N(j+1, P_1, \ldots, P_i + p_j, P_q, \Delta_1^1, \ldots, \Delta_i^s + a_j^s, \ldots, \Delta_q^{n^S})$
- Any directed path from the source node at level 0 to a terminal node at level n^J represents a feasible solution.
- Provides the optimal solution for any regual objective function of delivery completion times

Illustration



FPTAS algorithms for $1|q=2,|S|=1|T_{\max}$

1. $T_{\max}$ can be 0, therefore, we shift it by a problem instance specific value, e.g., u_q, the last delivery due-date.
2. FPTAS-1: one can turn the already seen pseudo-polynomial time algorithm into an FPTAS of time complexity (n^7/ε^4)
3. FPTAS-2: use an FPTAS for binary knapsack to solve it in e.g., $O(n^3/\varepsilon)$ time (state of the art is much better)

Consumer Jobs

Problem formulation

Problem data

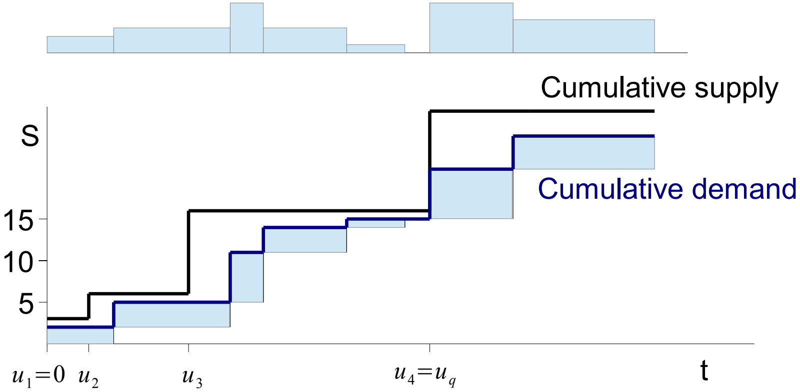
- Single machine, set of resource types R
- Jobs: J_j, processing time p_j, and quantities a_j^ρ required from each resource $\rho \in R$
- External supply at time points $0 = u_1 < \cdots < u_q$, and quantities supplied b_ℓ^ρ from each $\rho \in R$, $\ell = 1, \ldots, q$
- $u_{q+1} = \infty$, and $\sum_j a_j = \sum_\ell b_\ell$

Schedules

- Starting time S_j for each job J_j,
- Feasible if and only if jobs do not overlap in time, and $S_j < u_{\ell+1}$ implies

$$\sum_{i\ :\ S_i \le S_j} a_i^\rho \le \sum_{k=1}^{\ell} b_k^\rho, \qquad \forall \rho \in R$$

Consumer jobs



Earlier work on consumer jobs

Carlier (84), Grigoriev et al (2005):

- $1|rm=1|C_{max}$, $1|rm=1|L_{max}$ are NP-hard
- $1|rm=1, p_j=p|C_{max}$, $1|rm=1, p_j=1|L_{max}$ polynomially solvable
- $1|rm=1|C_{max}$ reduces to $F2||C_{max}$ in case of unit supply in each time unit.
- $1|rm=2, p_j=1|C_{max}$ NP-hard, there exist 2-approximation algorithms
- $1|rm=2, p_j=p|L_{max}$ NP-hard, there exist 2-approximation algorithms

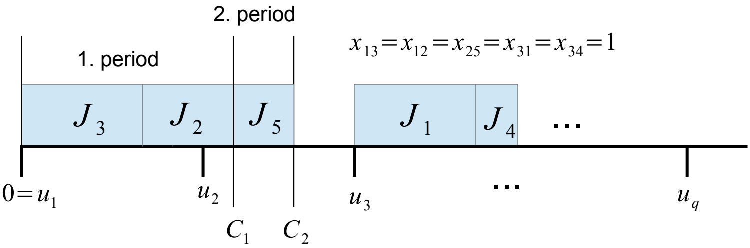
Summary of new results

1. FPTAS for $1|rm = 1, q = 2|C_{max}$
2. PTAS for $1|rm = 1, q = const|C_{max}$:
   - Use some ideas of Chekuri and Khanna (2006) for Multiple Knapsack Problem (MKP)
3. PTAS for $1|r_j, rm = 1, q = const|C_{max}$
4. PTAS for $1|r_j, rm = const, r - agr, q = const|C_{max}$

r-agr: there is a sequence π of jobs such that $a_{\pi(1)} \leq a_{\pi(2)} \leq \cdots \leq a_{\pi(n)}$ (componentwise)

Problem formulation

$x_{tj} = 1 \iff$ job j starts between u_t and u_{t+1}, otherwise 0.
C_t: completion time of those jobs starting in $[u_t, u_{t+1})$. $C_t > u_{t+1}$ is possible.
$b'_i := \sum_{\ell=1}^{i} b_\ell$



IP model

$$\min \max_{\ell=1}^{q} \left(u_\ell + \sum_{t=\ell}^{q} \sum_{j \in \mathcal{J}} p_j x_{tj} \right) \tag{1}$$

$s.t.$

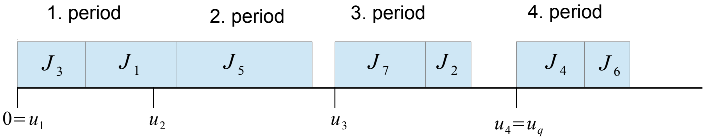
$$\sum_{j \in \mathcal{J}} a_j \left(\sum_{t=1}^{\ell} x_{tj} \right) \leq b'_\ell, \quad \ell = 1, \dots, q \tag{2}$$

$$\sum_{t=1}^{q} x_{tj} = 1, \quad j \in \mathcal{J} \tag{3}$$

$$x_{tj} \in \{0, 1\}, \quad t = 1, \dots, q; \quad j \in \mathcal{J} \tag{4}$$

PTAS: Preliminaries

- For any ε, we aim at $(1 + O(\varepsilon))$ approximation
- Notice that the number of jobs longer than εp_{sum} is at most $1/\varepsilon$. We schedule them in all possible ways.
- For any assignent $\bar{x}_{tj}^{B}$ of "big jobs" to supply periods, we schedule the remaining "small jobs" with minimum makespan
- After u_q the schedule is trivial $\rightarrow$ schedule as many small jobs as we can before u_q (respect resource constraints!)



Model for scheduling the small jobs

$$OPT_{\bar{x}^B} := \max \sum_{t<q, j\in\mathcal{S}} p_j x_{tj} \quad (7)$$

$s.t.$

$$\sum_{j\in\mathcal{S}} a_j \left(\sum_{t=1}^{\ell} x_{tj} \right) \leq \tilde{b}_\ell(\bar{x}^B), \quad \ell \leq q-1 \quad (8)$$

$$\sum_{j\in\mathcal{S}} p_j x_{tj} \leq \max\{0, u_{t+1} - C_t(\bar{x}^B)\} + \varepsilon p_{sum}, \quad t \leq q-1 \quad (9)$$

$$\sum_{t=1}^{q-1} x_{tj} \leq 1, \quad j \in \mathcal{S} \quad (10)$$

$$x_{tj} \in \{0,1\}, \quad t \leq q-1; \quad j \in \mathcal{S} \quad (11)$$

Connection with the Multiple Knapsack Problem

The MKP problem

- There are m knapsacks, each having its own size
- There are n items each having a size and a profit
- Items must be allocated to knapsacks such that
  - No knapsack is overfilled
  - The total profit of selected items is maximized

Connection with MKP:

- Two types of knapsack constraints, one is nested, the other is "standard"
- The items are the small jobs, and the knapsacks constraints are (8) and (9)
- Recycle some ideas from the PTAS of Chekuri and Khanna (2006) for MKP

How to schedule small jobs?

1. 'Guess' the optimum $OPT_{\bar{x}^B}$

   $$\text{Test each member of } G := \{p_{\max}^{S(\bar{x}^B)}(1+\varepsilon)^i \mid 0 \leq i \leq g\}$$

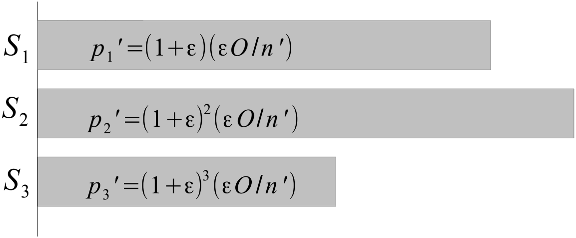
   where $g \leq \lfloor 2\varepsilon^{-1} \ln n \rfloor$
   $(p_{\max}^{S(\bar{x}^B)}(1+\varepsilon)^{g+1} > n \cdot p_{\max}^{S(\bar{x}^B)} \geq OPT_{\bar{x}^B})$

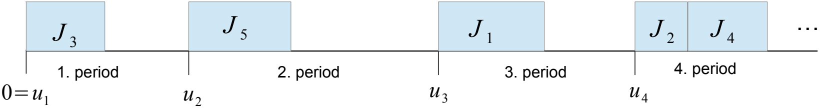
2. For each guess $\mathcal{O} \in G$, partition the set of small jobs into subsets $S_1, \ldots, S_h$:

   $$S_\ell := \{j \in S(\bar{x}^B) \mid (1+\varepsilon)^\ell(\varepsilon\mathcal{O}/n) \leq p_j < (1+\varepsilon)^{\ell+1}(\varepsilon\mathcal{O}/n)\}$$

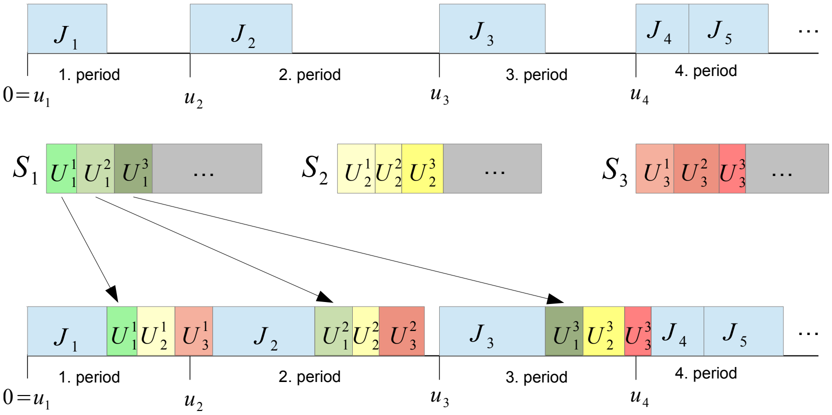
   where $h \leq \lfloor 4\varepsilon^{-1} \ln n \rfloor + 1$, for $(\varepsilon\mathcal{O}/n)(1+\varepsilon)^h > \mathcal{O}$

Partition the small jobs into sets S_l; each set is ordered in increasing resource req.





Schedule of big jobs



Complexity:

$$O(q^{1/\varepsilon} \cdot (2\varepsilon^{-1} \ln n) \cdot (n^2 \varepsilon^{-1} \log^2 n + (qn) \cdot n^{O(\varepsilon^{1-q} + \varepsilon^{-2})}))$$

Connection between the delivery tardiness (producer jobs), and the makespan minimization (consumer jobs) problems

Lemma

Given an instance $I = \{n, q, (p_j, a_j)_{j=1}^n, (u_\ell, b_\ell)_{\ell=1}^q\}$ *of the Delivery tardiness problem. Define an instance* $I' = \{n, q, (p_j, a_j)_{j=1}^n, (u'_\ell, b'_\ell)_{\ell=1}^q\}$ *of the Makespan minimization problem:*

$$\begin{aligned} u'_\ell &= u_q - u_{q+1-\ell} \\ b'_\ell &= b_{q+1-\ell} \end{aligned} \qquad \ell = 1, \ldots, q.$$

Then, if σ *is a sequence of jobs giving a maximum delivery tardiness of* $T^\sigma_{\max}$ *for* I*, then scheduling the jobs in reverse* σ *order gives a schedule of makespan* $u_q + T^\sigma_{\max}$ *for instance* I' *of the Makespan minimization problem.*

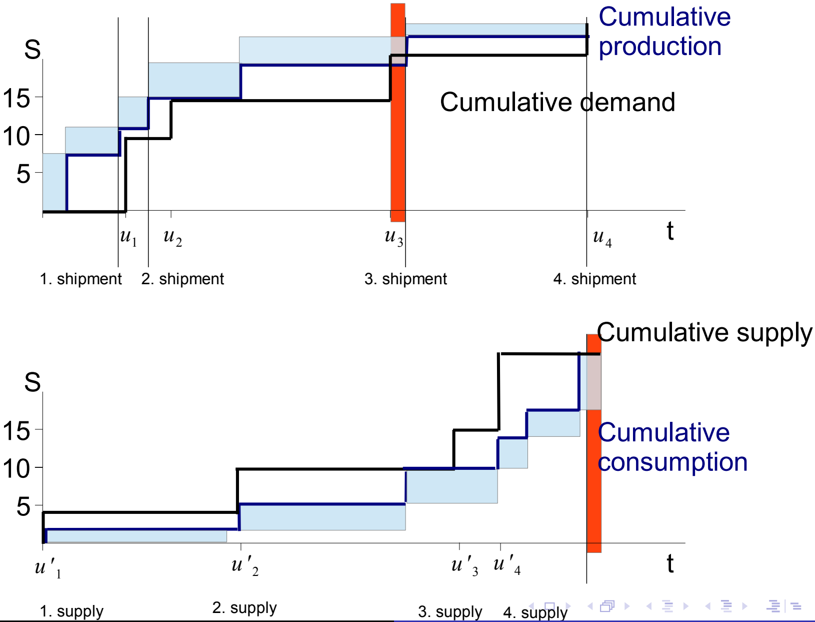
Connection ...

Lemma

Given an instance $I = \{n, q, (p_j, a_j)_{j=1}^n, (u_\ell, b_\ell)_{\ell=1}^q\}$ *of the Makespan minimization problem. Define an instance* $I' = \{n, q, (p_j, a_j)_{j=1}^n, (u'_\ell, b'_\ell)_{\ell=1}^q\}$ *of the Delivery tardiness problem:*

$$
\begin{aligned}
u'_\ell &= u_q - u_{q+1-\ell} \\
b'_\ell &= b_{q+1-\ell}
\end{aligned}
\qquad \ell = 1, \ldots, q.
$$

Then, if S is a schedule with a makespan of $C^S_{\max}$ for I, then scheduling the jobs in reverse order (without any delays among them) gives a schedule of maximum tardiness at most $C^S_{\max} - u_q$ for instance I' of the Delivery tardiness problem.


Cumulative production
S
15
10
5
Cumulative demand
u_1
u_2
u_3
u_4
t
1. shipment
2. shipment
3. shipment
4. shipment
Cumulative supply
S
15
10
5
Cumulative consumption
u'_1
u'_2
u'_3
u'_4
t
1. supply
2. supply
3. supply
4. supply

Consequences of the connection

Corollary

Let (I_D, I_M) be corresponding instances of the Delivery tardiness and the Makespan minimization problems. Then the optimum value $T^*_{\max}(I_D)$ of the Delivery tardiness problem equals $C^*_{\max}(I_D) - u_q$, the optimum value of the Makespan minimization problem minus u_q.

Theorem

The Delivery tardiness problem admits a FPTAS / PTAS for some q and r if and only if the Makespan minimization problem admits an FPTAS / PTAS for the same q and r.

Concluding remarks

Results on the way

- Approximation reductions between Knapsack and Scheduling problems with non-renewable resources (in both directions!)
- PTAS for fixed number of supplies and fixed number of resources
- PTAS for problems with release dates, and combination with fixed number of supplies and fixed number of resources

Open problems

- Does there exist a PTAS when the number of supplies / delivery due dates is part of the input?
- Extension: Multiple machines?
- Extension: Precedence constraints?

Thank you for your attention!

References

N. Boysen, S. Bock, M. Fliedner, Scheduling of inventory releasing jobs to satisfy time-varying demand: an analysis of complexity, *Journal of Scheduling*, Volume 16 (2013), 185–198.

C. Chekuri, S. Khanna: A Polinomial Time Approximation Scheme for the Multiple Knapsack Problem, *SIAM Journal on Computing*, Volume 35, No. 3 (2006), 713–728.

J. Carlier, *Problèmes d'ordonnancements á contraintes de ressources: algorithmes et complexitée*, Thèse d'état, 1984.

A. Grigoriev, M. Holthuijsen, J. van de Klundert: Basic Scheduling Problems with Raw Material Constraints, *Naval Research Logistics*, Vol. 52 (2005), 527–535.

M. Drótos, T. Kis, Scheduling of inventory releasing jobs to minimize a regular objective function of delivery times, *Journal of Scheduling*, Volume 16 (2013), 337–346.

P. Györgyi, T. Kis: Approximation schemes for single machine scheduling with non-renewable resources, *in print* (2013)