ELEMENTS OF SCHEDULING

Jan Karel Lenstra
Centrum Wiskunde & Informatica, Amsterdam

David Shmoys
Cornell University, Ithaca, NY

Scheduling seminar, March 31, 2021
schedulingseminar.com

HENRY LAURENCE GANTT
(1861–1919)

Henry Laurence Gantt was an industrial engineer and a disciple of Frederick W. Taylor. He developed his now famous charts during World War I to compare production schedules with their realizations. Gantt discussed the underlying principles in his paper "Efficiency and Democracy," which he presented at the annual meeting of the American Society of Mechanical Engineers in 1918. The Gantt charts currently in use are typically a simplification of the originals, both in purpose and in design.

Scheduling
Theory, Algorithms, and Systems

Second Edition

Michael Pinedo
New York University

Prentice Hall

Prentice Hall
Upper Saddle River, New Jersey 07458

Selmer M. Johnson

- 2-machine flow shop in $O(n \log n)$ time (1954)
- 42-city TSP, with Dantzig & Fulkerson (1954)
- generating permutations by adjacent transpositions (1963)

Wayne L. Smith

- $1||\Sigma C_j$ by SPT, $1||\Sigma w_j C_j$ by ratio rule (1956)

Early 1970's

operations research

- single machine, flow shop, job shop
- *classification scheme* (Conway, Maxwell & Miller 1967)

computer science

- single machine, parallel machines
- *complexity theory* (Cook 1971, Karp 1972)

mathematics

- *performance bounds* for LS and LPT on identical parallel machines (Graham 1966, 1969)

NATO ASI on Combinatorial Programming, Versailles, Sep. 1974

- Gene Lawler: *QAP; computational complexity*

Workshop on Integer Programming, Bonn, September 1975

- *complexity results* with Peter Brucker

Scheduling conference, Orlando, February 1976

- Marshall Fisher: *Lagrangian relaxation*
- Ed Coffman, Mike Garey, Dave Johnson: *bin packing*

Symposium on Algorithms & Complexity, Pittsburgh, April 1976

- Dick Karp: *probabilistic analysis*
- Nicos Christofides: *3/2-approximation for TSP with Δ≠*

DO77, Vancouver, August 1977

- *scheduling survey* of Graham, Lawler, L, Rinnooy Kan

S.C. Graves et al., Eds., *Handbooks in OR & MS, Vol. 4*

Chapter 9

Sequencing and Scheduling: Algorithms and Complexity

Eugene L. Lawler
University of California, Berkeley, CA, U.S.A.

Jan Karel Lenstra
Eindhoven University of Technology, Eindhoven, The Netherlands and CWI, Amsterdam, The Netherlands

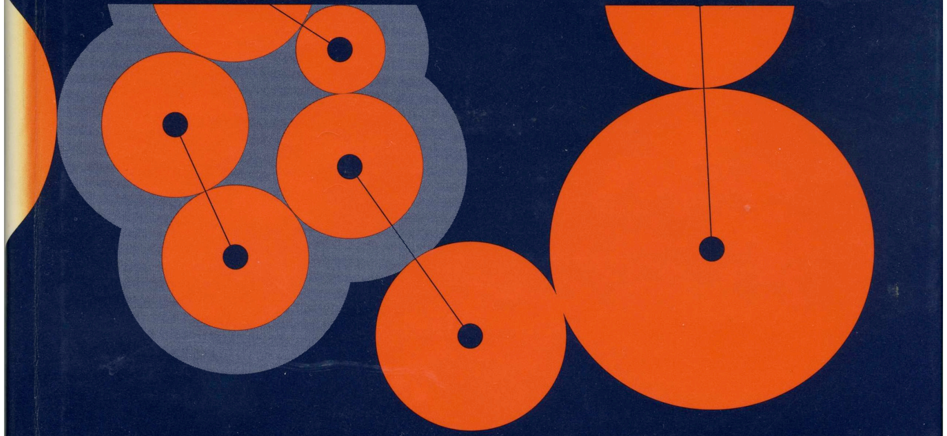
Alexander H.G. Rinnooy Kan
Erasmus University, Rotterdam, The Netherlands

David B. Shmoys
Cornell University, Ithaca, NY, U.S.A.

Sequencing and scheduling as a research area is motivated by questions that arise in production planning, in computer control, and generally in all situations in which scarce resources have to be allocated to activities over time. In this survey, we concentrate on the area of deterministic machine scheduling. We review complexity results and optimization and approximation algorithms for problems involving a single machine, parallel machines, open shops, flow shops and job shops. We also pay attention to two extensions of this area: resource-constrained project scheduling and stochastic machine scheduling.


C 255
095731
'91
ms to publisher
20
20
HÚSZ FORINT
HORVÁTH E. DEL.
NAGY ZOLTÁN SC.

Wiley-Interscience Series in Discrete Mathematics and Optimization



Combinatorial Optimization

William J. Cook
William H. Cunningham
William R. Pulleyblank
Alexander Schrijver

elementsofscheduling.nl

Preliminaries

Scheduling problems

Algorithms and complexity

Single machine

Minmax criteria

Weighted sum of completion times

Weighted number of late jobs

~~Total tardiness and beyond~~

Parallel machines

Minsum criteria

Minmax criteria, no preemption

Minmax criteria with preemption

~~Precedence constraints~~

Multi-operation models

Open shops

Flow shops

Job shops

More scheduling

Stochastic scheduling models

Scheduling in practice

Elements of Scheduling

collected and edited by Jan Karel Lenstra & David Shmoys

with contributions by Eugene Lawler, Charles Martel, Michael Pinedo, Maurice Queyranne, Alexander Rinnooy Kan, Andreas Schulz, Marc Uetz, David Williamson & Gerhard Woeginger

elementsofscheduling.nl

Illustration: Emile Aarts, *The job shop scheduling problem* (1997)

$P|prec,p_j=1|C_{\max}\leq 3$ (L, Rinnooy Kan 1978)

$R||C_{\max}\leq 2$ (L, Shmoys, Tardos 1990)

$O,F,J||C_{\max}\leq 4$ (Williamson, Hall, Hoogeveen, Hurkens, L, Sevast'janov, Shmoys 1997)

. . .

August
~~July~~ 20 1977.

On ~~july~~ august 20 1989 the $P \stackrel{?}{=} NP$ question will have been settled, this notion including undecidability

For: [signature]

Against: Eugene L Lawler

~~Best~~: 12 bottles of champagne

α|β|γ: *shorthand notation* or *research program*?

Challenges and developments

- analyze behavior of methods used in practice
- online methods
- learning from data

Looking forward

- evolving settings for scheduling in practice
- known unknowns
- learning from data

Data-driven scheduling & resource allocation

- cloud computing (scheduling & resource allocation)
- data center management (scheduling traffic in network)
- ride-share resource allocation

Models of an unknown future

- online analysis
  - arrivals over time
  - clairvoyance vs. non-clairvoyance
  - worst-case viewpoint
- stochastic models
  - queueing models (for arrivals)
  - probabilistic resource requirements
  - average-case (or w.h.p.) viewpoint

A very simple example

- $1|pmtn|\Sigma C_j$
  - Shortest Processing Time (SPT) rule optimal offline (no premptions)
  - but suppose the processing time is revealed only by running job
  - stochastic analysis
    - each job has a random variable processing time $\boldsymbol{p_j}$ (with ICR)
    - realization of random variable
    - optimal policy is to schedule in nondecreasing E[$\boldsymbol{p_j}$] order
  - nonclairvoyant analysis
    - each job has an unknown processing time
    - adversary determines processing times knowing algorithm
    - but possibly not knowing random coin tosses used by algorithm
    - round robin algorithm is 2-competitive
    - no (randomized) c-competitive algorithm exists with $c < 2$
    - [Motwani, Phillips, Torng 1994] [S, Wein, Williamson 1995]

A very simple example (continued)

- $1|pmtn|\Sigma C_j$ [Kumar, Purohit & Svitkina NIPS 2018]
  - if there exists means to exactly predict p_j then online = offline
  - machine learning provides such a mechanism in many settings
  - but predictions are noisy (and worse)
  - simple algorithm is Shortest Predicted Processing Time
  - SPPT finds optimal schedule if predictions are perfect
  - but can be arbitrarily bad if not
  - *Question: how do we hedge against inaccurate predictions?*
  - Step 1: bound performance of SPPT as a function of prediction error
    - Let m be the total prediction error for an input with n jobs
    - SPPT is a $(1+2m/n)$-competitive algorithm
  - Step 2: blend SPPT and round robin (like processor sharing)
    - if an α-fraction of time run SPPT and a $(1-\alpha)$-fraction run round robin
    - and set α so that $\alpha(1+2m/n) = 2(1-\alpha)$ obtain constant competitive ratio

A (much) more intricate example

[Lattanzi, Lavastida, Moseley, & Vassilvitskii 2020]

- $R|p_{ij} \in \{p_j, \infty\}, r_j|C_{\max}$
- consider an online arrival model with m machines
- each job has processing requirement p_j
- but can be feasibly assigned to specified subset of machines
- [L,S, & Tardos] offline 2-approximation algorithm (w/o r_j)
- [Azar, Naor & Rom 1995] O(log m)-approximation algorithm
- no online algorithm can have better competitive ratio
- Insight: algorithm uses succinct prediction about input (more sophisticated than the input data)
- Predicted quantity: weight w_i for each machine i
- Weights can be learned by an online algorithm

A (much) more intricate example (continued)
[Lattanzi, Lavastida, Moseley, & Vassilvitskii 2020]

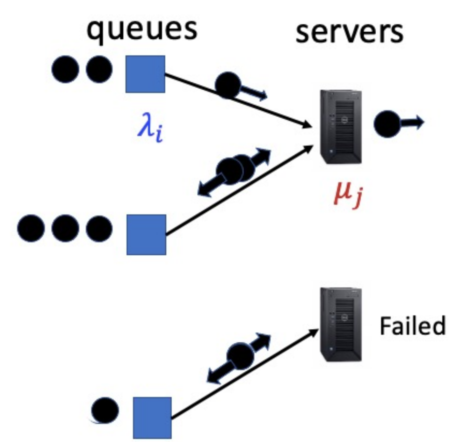
- $R|p_{ij} \in \{p_j, \infty\}, r_j|C_{\max}$
- Step 1: algorithm computes online fractional assignment
  - let *r* be *max*(predicted weight/"correct" weight)
  - online algorithm is O(log *r*)-competitive
  - intuition behind weights – fraction of job *j* assigned to machine *i* is ratio of machine *i*'s weight to total weight of feasible machines for *j*
- Step 2: algorithm rounds fractional assignment to {0,1}
  - online algorithm is $O((\log\log m)^3)$-competitive
  - algorithm is randomized and achieves makespan bound w.h.p.
- Can prove there exist weights for which proportional assignment has makespan within constant factor of optimum
- Also give online algorithm to "learn" weights
- Bottom line: $O((\log\log m)^3 \log r) = o(\log m)$ (lower bound)

A hint of a final example [Gaitonde & Tardos]

- builds on tradition of stochastic scheduling (queueing)
- and recent work on learning in games
- queues at routers compete for servers in rounds
- packets not sent (successfully) today, will be sent tomorrow

Model of Learning in a Queuing System

- Queue i gets new packets with a Bernoulli process with rate λ_i
- Server j succeeds at serving a packet with probability μ_j
- Each time step: each queue can send one packet to one of the servers to try to get serviced
- **Server can process at most one packet and unserved packets get returned to queue**
- Queues use no-regret learning to selfishly get the best service



Main Result: If servers have capacity for twice the arrival rate, and queues use no-regret learning, then queues stay bounded

Some concluding thoughts

- new models are needed for data-driven decision-making
- exciting interplay with machine learning
- bridges between deterministic and stochastic scheduling
- new opportunities for theory to guide and learn from practice