Scheduling Theory and Applications

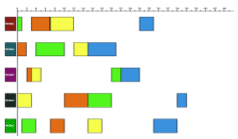
Prof. B.M.T. Lin (林妙聰)

Department of Information and Finance Management

Institute of Information Management

National Chiao Tung University, TAIWAN

May 2008



Outline

- Introduction and Overview
- Success Stories
- Production Scheduling
  - Single Machine
  - Parallel Machine
  - Shops and Flexible Assembly Systems
  - Supply-Chain Scheduling, Scheduling on the Internet and Green Logistics
- Workforce Planning and Timetabling
- Linking Real-world Applications with Academic Research
- Research Methods

Source:
M. Pinedo, *Planning and Scheduling in Manufacturing and Services*, Springer, 2005, New York.

S.I. Gass and A.A. Assad, *An Annotated Timeline of Operations Research: An Informal History*, Kluwer Academic, 2005, New York.

The Role of Scheduling

- Allocation of limited resources to tasks over time
- A decision-making process that has as a goal of the optimization of one or more objectives

Success Story:

The US Army National Guard Simulators Location and Routing Problem (Murty and Djang, 1999)

Decisions

- Home Base Selection: Select home bases for the mobile trainers from among the suitable sites
- Secondary Training Site Selection: Select the necessary secondary training sites so that each armory is within MTD of at least one of the home bases or secondary training sites
- Routing of Mobile Trainers
- Scheduling of Training Sessions

Objectives

- Mobile Trainer Fleet Mileage
- Platoon-Bus Mileage
- Number of Secondary Training Sites

Solution Models

- *p*-Median Problem for Home Base Selection
- Set-Cover Problem for Secondary Training Site Selection
- Vehicle Routing Problem for Routing of Mobile Trainers and Scheduling of Training Sessions

Results

- Reduction of mobile trainer mileages from 321,000 miles to 72,850 miles
- Cost saving mobile trainer mileage is $123M
- A one-time cost saving accrues from a reduction in the number of armory facility upgrades is $3M

Success Story:

SLIM: Short Cycle Time and Low Inventory in Manufacturing at Samsung Electronics

(Leachman, Kang and Lin, 2002)

The Problem

- SEC is a victim of its own success due to long the cycle time the fabrication process
- The original paradigm for scheduling production lots: urgency or due dates of lots to be scheduled
- Solution: Switch the paradigm to focusing on the capacity of each individual device

Results and Economic Benefits

- Cycle time required by fabricating 64MB RAM was reduced from 80 or more days to 27 days
- Total sales revenue of DRAM was $21.9 billions, among which $954 million was gained through the reduction of cycle time
- Total benefit 1 billion
- Market share increased from 18 ~ 22 percents
- President Y. W. Lee: *SLIM is essential to the success of SEC in the semiconductor business*.

More Applications

- College/University Class(room) Scheduling
- NBC’s Ads Scheduling
- Crew Recovery and Pilot Training at Continental Airlines
- Delta optimizes continuing-qualification-training schedules for pilots
- Scheduling Umpire Crews for Professional Tennis Tournaments
- Scheduling the Beef Fabrication Process at Swift & Company
- Scheduling of Refuel at NY Nuclear Power Plant
- Sequence-dependent Scheduling at Baxter International

Framework (Deterministic Models)

- Framework: three-field notation $\alpha|\beta|\gamma$
- α: machine configuration
- β: job characteristics and/or constraints
- α: objective function

Framework (Deterministic Models)

- Jobs
  - Static data:
    - p_{ij}: processing time of job j on machine i.
    - r_{ij} : release date of job j on machine i
    - d_j : due-date/deadline of job j
    - w_j: weight (importance indicator) of job j
  - Dynamic data:
    - starting time (S_{ij})
    - completion time (C_{ij})

Framework (Deterministic Models)

Machine Configurations

- Single machine (1)
- Identical machines in parallel (*Pm*)
- Machines in parallel with different speeds (*Qm*)
- Unrelated Machines in parallel (*Rm*)
- Flow shop (*Fm*)
- Flexible flow shop (*FFs*)
- Open shop (*Om*)
- Job shop (*Jm*)
- Supply chain models

Framework (Deterministic Models)

Processing Characteristics and Constraints

- Release dates
- Sequence-dependent setup times/costs
- Preemptions
- Precedence constraints
- Breakdown and maintenance constraints
- Machine eligibility constraints
- Permutation
- Blocking/No-wait

Framework (Deterministic Models)

Processing Characteristics and Constraints

- Recirculation/Reentrant
- Workforce constraints
- Routing constraints
- Material handling constraints
- Make-to-stock and make-to-order
- Transportations

Framework (Deterministic Models)

Performance Measures and Objectives

- Throughput and makespan objectives $C_{\max}$
- Total (weighted) completion time: $\Sigma w_j C_j$, ΣC_j
- Due-date related objectives
  - Maximum lateness, $\max\{C_j - d_j\}$
  - Total (weighted) tardiness, $\Sigma w_{ij} \max\{0, C_j - d_j\}$
  - Total late work, $\Sigma \max\{p_j, \max\{0, C_j - d_j\}\}$
  - (Weight) Number of tardy jobs, $\Sigma w_j U_i$ ΣU_i
  - Earliness-Tardiness $\Sigma \alpha_j E_j + \Sigma \beta_j T_j$
- Setup costs
- Work-in-process (WIP) inventory costs
- Finished goods inventory costs
- Transportation costs

Example Problems

- $1 \mid r_i \mid \Sigma U_i$
- $1 \mid \text{prec} \mid \Sigma w_i C_i$
- $Pm \mid \text{tree} \mid L_{\max}$
- $Fm \mid \mid C_{\max}$
- $Om \mid r_{ij} \mid C_{\max}$

Service Models

- Activities are concerned: meeting, flight leg, game, appointment, personnel position
- Typically no goods to inventorize
- The amount of resources may vary over time

Operational Characteristics and Constraints

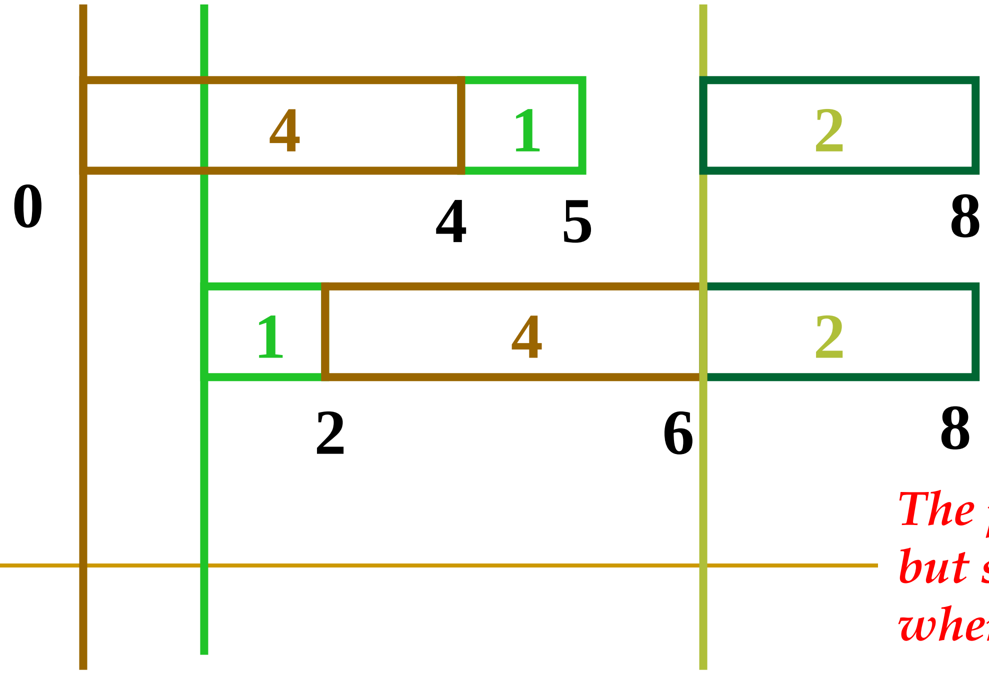
- Time windows (release dates and due dates)
- Capacity requirements and constraints
- Preemptions
- Operator and tooling requirements
- Workforce scheduling constraints

Performance Measures and Objectives

- Makespan
- Setup costs
- Earliness and tardiness costs
- Convenience costs and penalty
- Personnel costs

Single Machine

- Only a single server is available
- At any time, only one job/activity can be processed
- Consider the following example:
  - Each job has a processing time p_i and a release date r_i.
  - We want to find a feasible schedule such that the sum of completion times is minimized.



Solution value 1
= 4+5+8

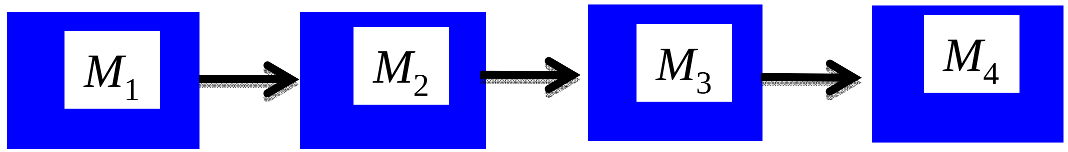
Solution value 2
= 2+6+8

The problem is NP-hard, but solvable in polynomial time when preemption is allowed

Parallel Machines

- A job/activity can be processed by any of the machines
- Machines may have different processing speeds/capabilities
- Most of the standard objectives are NP-hard

Flow Shops



- Flow shop is a set of (*m*) machines that are arranged in a pipeline fashion
- Each job J_i has m operations in which operation O_{ij} must be processed on machine M_j with processing time p_{ij}
- Processing of all jobs must follow the route $M_1, M_2, \ldots, M_m$

Job Shops

- A set of (m) machines are available
- Each job J_i has m or less operations
- Different jobs may have different processing routes
- e.g. job ←→ patient

SCM Scheduling

Medium-term Planning + Short-term Scheduling
Inventory cost, transportation cost, tardiness/shortage cost, …
are involved

Scheduling on the Internet

Green Logistics

Interval Scheduling, Reservation, Timetabling and Workforce Scheduling

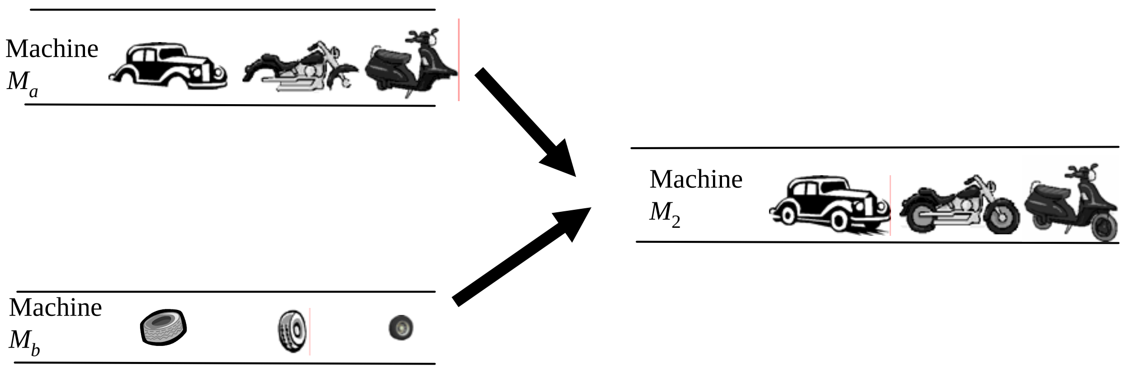
Off-day scheduling

Crew scheduling

Sports Scheduling, Train Timetabling

Linking Real-world Applications with Academic Research

A Model for Fire Engine Manufacturing



Redevelopment Project in Boston

H	h_1	h_2	h_3	h_4	h_5	h_6	h_7
a_i	4	8	5	21	7	19	18
b_i	7	5	15	17	16	12	11
δ_i	3	-3	10	-4	9	-7	-7

Sequence $S = h_1\, h_3\, h_5\, h_4\, h_6\, h_7\, h_2$

t	0		1		2		3		4		5		6		7
V_t	7		10		20		29		25		18		11		8
δ_i		h_1		h_3		h_5		h_4		h_6		h_7		h_2	

Redevelopment Project in Boston

H	h_1	h_2	h_3	h_4	h_5	h_6	h_7
n_i	4	8	5	21	7	19	18
a_i	7	5	15	17	16	12	11
δ_i	3	-3	10	-4	9	-7	-7

Sequence $S = h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_7$

t	0	1	2	3	4	5	6	7
V_t	7	10	7	17	X -			
h_{σ_t}		h_1	h_2	h_3	h_4			

Film Production Problem

actor i	Shooting day j 1	2	3	4	5	6	7	8	Holding Cost c_i
1	1	1	1	0	0	1	0	0	810
2	1	1	0	0	0	1	1	0	700
3	0	1	0	1	0	0	0	1	910
4	0	1	0	1	1	1	1	0	880
5	0	1	0	0	0	0	0	1	400
6	0	0	1	1	0	0	0	1	610

Total cost = 2*810+3*700+4*910
+880+5*400+3*610

Film Production Problem

actor i	Shooting day j								Holding Cost c_i
	8	2	3	4	5	6	7	1	
1	0	1	1	0	0	1	0	1	810
2	0	1	0	0	0	1	1	1	700
3	1	1	0	1	0	0	0	0	910
4	0	1	0	1	1	1	1	0	880
5	1	1	0	0	0	0	0	0	400
6	1	0	1	1	0	0	0	0	610

Total cost = 3*810+3*700+1*910
+880+610

Runway Scheduling at Rome Airport

Research Methods

- Modeling and Formulation
- Settling time complexity (NP-hardness Vs. polynomial solvability)
- Exact solution methods
  - Branch-and-bound algorithms (optimality properties, bounding functions, dominance rules)
  - Dynamic programming algorithms
- Approximation Approach
  - Heuristics: computational & analytical performances
  - Meta-heuristics/Local Search: computational study
  - Approximability/non-approximability
  - Approximation scheme

Q & A

Proof of NP-hardness

Theorem 1. *The* RPD_2^+ *problem is NP-hard in the strong sense.*

Proof. To establish the NP-hardness of the two-machine relocation problem, we present a polynomial-time reduction from the NMTS problem. Suppose that we are given two sets of integers $\{x_1, x_2, \ldots, x_n\}$, and $\{y_1, y_2, \ldots, y_n\}$, and a collection of targets $B_1, B_2, \ldots, B_n$. Without loss of generality, we assume $B_1 \le B_2 \le \cdots \le B_n$. We group all targets with the same value together in a subset. Assume there are k different target values and let m_i be the cardinality of ith subset. We get $B_1 = \cdots = B_{m_1} < B_{m_1+1} = \cdots = B_{m_1+m_2} < \cdots < B_{m_1+m_2+\cdots+m_{k-1}+1} = \cdots = B_{m_1+m_2+\cdots+m_k} = B_n$. Let $M = 1 + \max_{i=1,\ldots,k} m_i$. Set $\bar{B}_i = B_{m_1+m_2+\cdots+m_i}$ for all $1 \le i \le k$.

We define an instance I of RPD_2^+ as follows. In instance I, there are two machines, $2n$ basic jobs J_{1i}, $i = 1, 2, \ldots, n$, and J_{2i}, $i = 1, 2, \ldots, n$, and $k-1$ connecting jobs $\bar{J}_i$, $i = 1, \ldots, k-1$. All connecting jobs and jobs J_{1i}, $i = 1, 2, \ldots, n$, have to be executed on machine one, and jobs J_{2i}, $i = 1, 2, \ldots, n$, have to be executed on machine two. For each integer x_i, we create basic job J_{1i} with $\alpha_{1i} = 2Mx_i$ and $\beta_{1i} = 2Mx_i + 1$. For each integer y_i, we create basic job J_{2i} with $\alpha_{2i} = 2My_i$ and $\beta_{2i} = 2My_i + 1$. Note that

$$\beta_{1i} > \alpha_{1i} > ME/2 \quad \text{for all } i = 1, \ldots, n. \tag{5}$$

We use $\bar{\alpha}_i$, $\bar{\beta}_i$, $\bar{\delta}_i$ to denote the amount of required resources, amount of returned resources and net contributions of connecting job $\bar{J}_i$. For each connecting job $\bar{J}_i$, define $\bar{\alpha}_i = 2M\bar{B}_i + 2m_i$ and $\bar{\beta}_i = 2M\bar{B}_{i+1}$. Note that $\bar{\delta}_i = 2M\bar{B}_{i+1} - (2M\bar{B}_i + 2m_i) = 2M(\bar{B}_{i+1} - \bar{B}_i) - 2m_i \ge 2M - 2m_i > 0$. The inequality implies $\bar{\alpha}_i < \bar{\alpha}_{i+1}$ for all $i = 1, \ldots, k-2$. Moreover, we have

$$\bar{\alpha}_i = 2M\bar{B}_i + 2m_i > 2M\bar{B}_1 = 2MB_1 > ME. \tag{6}$$

Table 3. The case $p_i \in [1, 50]$ and $q_i \in [1, 100]$ with dominance rules.

n	LB_{LA}							LB_{TR}						
	Opt_cnt	Time			Node			Opt_cnt	Time			Node		
		Min	Avg	Max	Min	Avg	Max		Min	Avg	Max	Min	Avg	Max
10	10	0.01	0.04	0.07	192	897	1,734	10	0.00	0.00	0.01	6	135	654
15	10	0.91	8.06	28.72	11,833	113,976	419,410	10	0.00	0.02	0.09	6	26,080	9,157
20	10	17.23	417.66	697.82	157,161	3,718,077	6,029,032	10	0.00	0.46	2.23	391	29,937	146,346
25	1	1,290.98	1,290.98	1,290.98	7,789,575	7,789,575	7,789,575	10	0.00	0.30	2.82	9	16,020	145,400
30	0	—	—	—	—	—	—	10	0.02	62.15	566.90	2,009	2,082,991	17,822,202
35	0	—	—	—	—	—	—	10	0.03	47.29	169.57	2,904	1,701,164	5,223,273
40	0	—	—	—	—	—	—	8	0.00	170.54	630.04	232	5,836,917	21,175,595
45	0	—	—	—	—	—	—	6	0.13	558.24	1,407.22	9,442	17,275,578	41,658,122
50	0	—	—	—	—	—	—	5	0.02	296.02	1,057.09	1,317	9,232,193	32,209,774
55	0	—	—	—	—	—	—	5	0.03	425.89	1,780.13	1,567	11,513,190	48,201,711
60	0	—	—	—	—	—	—	3	57.73	617.15	947.48	2,038,125	16,939,198	25,731,039
65	0	—	—	—	—	—	—	1	414.83	414.83	414.83	10,491,953	10,491,953	10,491,953

Algorithm CB-TCT:

Initial conditions:

$h(n+1, u) = 0$ for all $u, 1 \leq u \leq n$.

Recursion:

$$h(i,u) = \min_{i \leq j \leq n} \left\{ h(j+1, u+1) + (j-i+1) \times \max_{1 \leq k \leq m} \left\{ (u+1)s_k + \sum_{l=1}^{j} p_{kl} \right\} \right\}$$

Goal:

Find $h(1, 0)$.

Approximation Algorithm and Non-approximability

Splitting Algorithm.

1. Let I be an instance of RP. Apply Rounding Algorithm to I and let σ be the derived schedule.
2. Let m be the number of machines and n_t be the number of jobs scheduled in interval $[t, t+1)$ for $t = 0, \ldots, Z(\sigma) - 1$.
   For all t such that $n_t > m$ **do:**
   Set $\tau := \lceil \frac{n_t}{m} \rceil$
   Schedule n_t jobs in intervals $[t, t+1), [t+1, t+2), \ldots, [t+\tau-1, t+\tau)$ such that each of the first $\tau - 1$ intervals contain m jobs.
   For each job J_i with $C_i(\sigma) \geq t+1$ set $C_i(\sigma) = C_i(\sigma) + \tau$.
3. Stop.

Theorem 5. *Splitting Algorithm is a* $(3 - \frac{2}{m})$*-approximation algorithm for the* RP *problem with m identical parallel machines.*

Proof. Let σ be the schedule obtained by Splitting Algorithm. We say that time interval $[t, t+1)$ is *complete* if exactly m jobs are allocated within this interval in schedule σ, otherwise time interval $[t, t+1]$ is incomplete. Let h and l be the numbers of complete and incomplete intervals correspondingly, i.e. $Z(\sigma) = l + h$.

Note that after the splitting of "overloaded" unit time intervals, we get at most one new unit time interval that has less than m jobs. It follows that l does not exceed the length of schedule obtained by Rounding Algorithm. From Theorem 4 we have that $l \leq 2 \times \text{OPT}$. On the other hand, for any instance of the relocation problem with n UET jobs and m machines we get $\frac{hm+l}{m} \leq \text{OPT}$. The inequality follows from the fact that each complete interval contains exactly m jobs and each uncomplete interval contains at least one job. Combining the above facts, we thus get

$$\frac{Z(\sigma)}{\text{OPT}} \leq \frac{l+h}{\max\{\frac{l}{2}, \frac{hm+l}{m}\}}.$$

Approximation Algorithm and Non-approximability

Theorem 3. *Unless* $P = \mathrm{NP}$, *there is no* ρ*-asymptotic approximation algorithm for the* RP_∞ *problem for any* $\rho < \frac{4}{3}$, *even if* $\alpha_i \le \beta_i$ *and* $p_i = 1$ *for all* i.

Proof. We consider an instance I of the Partition problem: Given a finite set of integers $X = \{e_1, e_2, \ldots, e_n\}$ with $\sum_{e_i \in X} e_i = 2E$, is there a subset $X_1 \subseteq X$ such that $\sum_{e_i \in X_1} e_i = E$?

We define an instance I' of the RP_∞ problem as follows. Let $K > 0$ be a fixed integer. We introduce $n(K+1)$ basic jobs J_{ki} with $\alpha_{ki} = (E+1)^k e_i$ and $\beta_{ki} = (E+1)^k e_i + \frac{1}{n}$ for all $k = 0, 1, \ldots, K$ and $i = 1, 2, \ldots, n$. Note that for convenience parameter β_{ki} does not abide by the integer constraint. The violation can be resolved by scaling all parameters with a factor n. The jobs are grouped into $K+1$ sets of n jobs with the same first index, i.e. set $N_k = \{J_{k1}, J_{k2}, \ldots, J_{kn}\}$. We define K connecting jobs $\bar{J}_k$, $1 \le k \le K$, with $\bar{\alpha}_k = E(E+1)^{k-1} + 1$ and $\bar{\beta}_k = E(E+1)^k$. The amount of initial resources is $\Omega_0 = E$.

We use the notation of schedule concatenation as follows: $\sigma = N_0 N_1 N_2 \ldots N_l$ means that the jobs of set N_i are followed by the jobs of set N_{i+1} in schedule σ for all $i = 0, 1, 2, \ldots, l-1$. It is easy to see that in any feasible schedule σ, the jobs are processed in the following order $\sigma = N_0\{\bar{J}_1\}N_1\{\bar{J}_2\}N_2\{\bar{J}_3\}N_3 \ldots \{\bar{J}_K\}N_K$. Since the amount of initial resources is equal to E, only the jobs from set N_0 are available for processing at time zero. Moreover, since the net contribution of each basic job is equal to $\frac{1}{n}$, only the connecting job $\bar{J}_1$ becomes available when all jobs from set N_0 finish their processing. After finishing job $\bar{J}_1$, the amount of resources is equal to $E(E+1)$ and we can schedule only jobs from set N_1. This line of reasoning continues for the remaining jobs (Fig. 2). In schedule σ, at the time job J_i, $i = 1, \ldots, K$ is to be processed, it will consumes all available resources. It follows that no job can be processed in parallel with a connecting job J_i. Now consider the jobs from any set N_k, $0 \le k \le K$. They require $\sum_{i=1}^{n} \alpha_{ki} = \sum_{i=1}^{n} (E+1)^k e_i = 2E(E+1)^k$ units of resources. The amount of resources available after finishing jobs in $N_0\{\bar{J}_1\}N_1\{\bar{J}_2\}N_2\{\bar{J}_3\}N_3 \ldots \{\bar{J}_k\}$ is equal to $E(E+1)^k$. Since $\delta_{ki} = \frac{1}{n}$ and all e_i are integers we can schedule the jobs of set N_k in time interval of length 2 if and only if there exists a partition of the job set