Planning and scheduling models

Logistics

Giovanni Righini

UNIVERSITÀ DEGLI STUDI
DI MILANO

Planning

Production planning is a typical operation at a tactical decision level in production logistics.

Decide the amount of products to be produced on a medium term horizon (one week - one year).

Different levels of planning can be nested for different time periods, from longer to shorter term plans.

Production planning may involve:

- a single product or multiple products
- a single period or multiple periods
- fixed and variable production costs
- constraints on the lot size
- constraints on resource availability
- deadlines or due dates.

Objectives:

- minimize the production cost;
- maximize the expected profit.

The optimal production mix problem

The problem assumes a set P of products and a single period.
Production requires a set R of resources.
For each product $j \in P$ an expected unit profit c_j is known.
For each resource $iinR$ an available amount b_i is known.
A given technological coefficient a_{ij} indicates the amount of each resource needed to produce a unit of each product.
The objective is to maximize the total expected profit.

A (continuous or integer) variable x_j indicates the production level for each product $j \in P$.

$$\text{maximize } z = \sum_{j \in N} c_j x_j$$

$$\text{subject to } \sum_{j \in N} a_{ij} x_j \leq b_i \qquad \forall i \in R$$

$$x_j \geq 0 \text{ (integer)} \qquad \forall j \in P$$

Multi-period planning

When several consecutive periods are considered, the common assumption is that a warehouse of limited capacity is available to stock an amount of products that can be distributed or sold later.

Flow conservation constraints are needed to ensure the consistency of produced and stored amounts in consecutive periods.

$$x_t + s_{t-1} = d_t + s_t$$

where

- $x_t \geq 0$ indicates the production in period t;
- $s_t \geq 0$ indicates the amount stored in the warehouse at the end of period t;
- d_t indicates the demand satisfied in period t.

The lot-sizing problem

The problem assumes a single product and a set T of potential production periods.

A demand d_t is known for each period $t \in T$.

A fixed production cost f_t and a variable production cost c_t are given for each period $t \in T$.

A maximum production capacity q_t for each period and a warehouse capacity Q are also given.

A unit inventory cost h_t is known for each period $t \in T$.

The objective is to satisfy the demand at minimum total cost.

The lot-sizing problem

A binary variable y_t indicates whether period t is used or not.
A continuous or integer variable $x_t \geq 0$ indicates the production in period t.
A variable $s_t \geq 0$ indicates the amount stored at the end of period t.

$$\text{minimize } z = \sum_{t \in T} (f_t y_t + c_t x_t + h_t s_t)$$

$$\begin{aligned}
\text{subject to } & x_t \leq q_t y_t && \forall t \in T \\
& s_{t-1} + x_t = d_t + s_t && \forall t \in T : t > 1 \\
& s_0 + x_1 = d_1 + s_1 && \\
& x_t \geq 0 \text{ (integer)} && \forall t \in T \\
& s_t \geq 0 && \forall t \in T \\
& y_t \in \{0, 1\} && \forall t \in T
\end{aligned}$$

Assignment problems

At a more detailed decision level, production lots must be assigned to production units or teams.

This originates assignment problems between a set N of tasks/jobs and a set M of agents/machines.

The decision is typically represented by binary assignment variables:

$$x_{ij} = \begin{cases} 0 & \text{if task } i \in N \text{ is not assigned to machine } j \in M \\ 1 & \text{if task } i \in N \text{ is assigned to machine } j \in M \end{cases}$$

Assignment costs/times are given for each possible assignment (i, j) between a task $i \in N$ and a machine $j \in M$.

The linear assignment problem

A set N of jobs and a set M of agents are given, with $|N| = |M|$.
An assignment cost c_{ij} is given for each $i \in N$ and $j \in M$.
Assign a job to each agent and an agent to each job, minimizing the total assignment cost.

Binary variables x_{ij} are used to represent the selected assignments.

$$\text{minimize } z = \sum_{i \in N, j \in M} c_{ij} x_{ij}$$

$$\text{subject to } \sum_{i \in N} x_{ij} = 1 \qquad \forall j \in M$$

$$\sum_{j \in M} x_{ij} = 1 \qquad \forall i \in N$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in N, \forall j \in M$$

The generalized assignment problem

A set N of jobs and a set M of agents are given.
An assignment cost c_{ij} and a resource consumption a_{ij} are given for each $i \in N$ and $j \in M$.
A capacity (amount of available resource) b_j is given for each machine $j \in M$.
Assign jobs to machines, minimizing the total assignment cost and complying with capacity constraints.

Binary variables x_{ij} are used to represent the selected assignments.

$$\text{minimize } z = \sum_{i \in N, j \in M} c_{ij} x_{ij}$$

$$\text{subject to } \sum_{i \in N} a_{ij} x_{ij} \leq b_j \qquad \forall j \in M$$

$$\sum_{j \in M} x_{ij} = 1 \qquad \forall i \in N$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in N, \forall j \in M$$

Scheduling problems

Scheduling problems are typically short-term decisions, arising at an operational level, when operations are considered in full detail and time is taken into account: for each lot to be produced on a machine a starting time must be decided, according to a suitable job sequence: the schedule.

A set N of jobs and a set M of machines are given.
Each job $i \in N$ can be characterized by

- a processing time p_i
- a release date r_i
- a due date d_i
- a deadline D_i
- a weight w_i
- a delay cost c_i

An available time horizon T is given.
The objective is to schedule all jobs, minimizing a function of their completion times.

Scheduling problems

Scheduling problems are typically very difficult to solve to proven optimality.

They involve assignment decisions to assign jobs to machines (binary variables).

They involve sequencing decisions, that can be represented in several ways.

Linear ordering

Linear ordering variables (binary):
$x_{ij} = 1$ if and only if i precedes j, for each pair of distinct jobs i and j.

Linear ordering constraints:

$$x_{ij} + x_{ji} = 1 \qquad \forall i \neq j \in N$$
$$x_{ij} + x_{jk} + x_{kl} = 2 \qquad \forall i \neq j \neq k \in N$$
$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in N$$

Time-indexed formulations

Time-indexed variables (binary):
$x_{it} = 1$ if and only if job i starts at time t, $\forall i \in N$ and $\forall t \in T$.

The time axis is discretized: the granularity affects the number of variables.

Time-indexed formulation constraints:

$$\sum_{t=1}^{T-p_j+1} x_{jt} = 1 \qquad \forall j \in N$$

$$\sum_{j \in N} \sum_{s=\max\{1,t-p_j+1\}}^{t} x_{js} \leq 1 \qquad \forall t = 1, \ldots, T$$

$$x_{jt} \in \{0, 1\} \qquad \forall j \in N \; \forall t = 1, \ldots, T$$

Disjunctive constraints

Disjunction variables (binary):
$x_{ij} = 1$ if and only if job i starts before job j, $\forall i \neq j \in N$.

Starting time variables (continuous):
t_j indicates the starting time of each job $j \in N$.

Disjunctive constraints:

$$t_i \geq t_j + p_j + Mx_{ij} \qquad \forall i \neq j \in N$$
$$x_{ij} + x_{ji} = 1 \qquad \forall i \neq j \in N$$
$$t_i \geq 0 \qquad \forall i \in N$$
$$t_i \leq T - p_i \qquad \forall i \in N$$

Classification

In 1979 Graham et al. introduced a three fields notation to classify scheduling problems

$$\alpha|\beta|\gamma$$

where

- α describes the machines
- β describes the jobs
- γ describes the objective.

Classification: α

Some possible values for α are:

- 1: single machine
- *P* (identical): the processing time of each job does not depend on the machine;
- *Q* (uniform): the machines have different speed s_j and the processing time of job *i* on machine *j* is $p_{ij} = p_i/s_j$;
- *R* (unrelated): the values of p_{ij} are in general different and unrelated;
- *F* (flow shop): each job must be processed by each machine according to a fixed machine sequence which is the same for all jobs;
- *J* (job shop): as in flow shop but the machine sequence can be different for each job;
- *FJ* (flexible job shop): machines are replaced by work centers equipped with parallel identical machines;
- *O* (open shop): all jobs must be processed on all machines but without routing restrictions.

Classification: β

Some possible values for β are:

- r_j: with release dates;
- *prmp*: with preemption (and resume);
- with precedence constraints, represented by a digraph G with a node for each job:
  - *prec*: G is generic;
  - *chain*: G is a set of paths;
  - *tree*: G is an oriented tree;
- *brkdwn* (breakdown): with fixed unavailability periods for the machines;
- s: with set-up times (dependent or independent of the machine);
- $p_j = 1$: unitary processing times.

Classification: KPIs

Some typical KPIs in scheduling:

- C_j: completion time of job j;
- $L_j = C_j - d_j$: lateness of job j (it can be negative);
- $T_j = \max\{L_j, 0\}$: tardiness of job j (non-negative);
- $U_j = 1$ if $C_j > d_j$ and 0 otherwise.

Classification: γ

Some possible values for γ are:

- C_{max}: minimize the completion time of the last job (*makespan*);
- L_{max}: minimize the maximum lateness;
- $\sum C_j$: minimize the total completion time;
- $\sum w_j C_j$: minimize the total weighted completion time;
- $\sum (w_j) T_j$: minimize the total (weighted) tardiness;
- $\sum (w_j) U_j$: minimize the total (weighted) number of tardy jobs.

Single-machine scheduling

For some single-machine scheduling problems, the optimal solution can be computed by sorting the jobs according to specific ordering criteria.

Some well-known examples are:

- $1||\sum_j w_j C_j$
- $1|r_j, prmp|\sum_j C_j$
- $1||L_{max}$
- $1||\sum_j U_j$

Several single-machine scheduling problems can be solved in polynomial or pseudo-polynomial time by dynamic programming.

Sum of weighted completion times

$1||\sum_j C_j$ (unweighted case).

SPT (Shortest Processing Time) rule (Smith's rule): sort the jobs by non-decreasing processing time.

Proof: by exchange.
Complexity: $O(n \log n)$.

$1||\sum_j w_j C_j$ (weighted case).

WSPT (Weighted Shortest Processing Time) rule (Smith's rule): sort the jobs by non-decreasing values of the ratio w_j/p_j.

Proof: by exchange.
Complexity: $O(n \log n)$.

Preemption and unit weights

$1|r_j, prmp|\sum_j C_j$.

This problem has unit weights, release dates and preemption.
SRPT (Shortest Remaining Processing Time) rule: sort the jobs by non-decreasing residual processing time.

Proof: by exchange.
Complexity: $O(n \log n)$.

Maximum lateness

$1||L_{max}$.

EDD (Earliest Due Date) rule (Jackson's rule): sort the jobs by non-decreasing due dates.

Proof: by exchange.
Complexity: $O(n \log n)$.

Number of tardy jobs

$1||\sum_j U_j$.

EDD (Earliest Due Date) rule (Moore's algorithm):

- sort the jobs by non-decreasing due dates (EDD)
- start with an empty schedule
- for each job j in the EDD ordered list
  - append j to the schedule
  - if j is late, then select the longest job k in the schedule and remove it
- add all the removed jobs at the end of the schedule (in any order).