Optimization of Foundry Production Processes 3

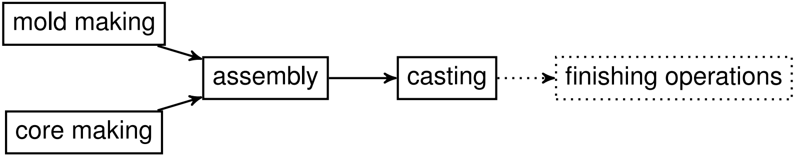
Anna Kelemen

Supervisor: Alpár Jüttner
Eötvös Loránd University

January 8, 2026

Foundry production process

- casting: shaping molten metal by pouring it into molds



Motivation and context

- Continuation of previous foundry scheduling projects
- Earlier objectives:
  - minimize the number of casting shifts
  - minimize idle time between consecutive operations
- New extensions in this project:
  - missing deadlines is allowed with penalty
  - material-type restriction in casting batches
- New objectives:
  - minimize the number of casting shifts
  - minimize total lateness

Problem summary

- Sets:
  - P: products
  - A: assembly shifts
  - C: casting shifts
- Capacities:
  - assembly shifts: time-based capacity
  - casting shifts: weight limit (4 tonnes)
- Product parameters:
  - weight w_p
  - deadline d_p
  - material type $t_p \in \{0, 1\}$
  - processing times for mold, core, and assembly tasks

Casting-only IP model

$$\min \; c_{\text{cast}} \sum_{c \in C} (y_{c0} + y_{c1}) + c_{\text{late}} \sum_{p \in P} L_p$$

$$\sum_{c \in C} x_{pc} = 1 \qquad \forall p \in P$$

$$\sum_{p : t_p = k} w_p x_{pc} \le W y_{ck} \qquad \forall c \in C, k \in \{0, 1\}$$

$$y_{c0} + y_{c1} \le 1 \qquad \forall c \in C$$

$$L_p \ge s_c - d_p - M(1 - x_{pc}) \qquad \forall p, c$$

Extended IP model

- Issue: the first model ignores assembly shift capacities
- Solution: cumulative assembly constraint

$$\sum_{p \in P} \sum_{c' \leq c} t_p^{tot} x_{pc'} \leq \sum_{a \in A(c)} T_a \quad \forall c \in C$$

- Improves global feasibility
- Drawback:
  - increased complexity and runtime

Full scheduling with column generation

- Full IP with all operations is too large to solve directly
- Use LP relaxation with column generation
- Decision variables represent complete schedules:
  - one schedule per product
  - includes all assembly shifts and the casting shift
- Schedules are generated dynamically
- Objective:
  - weighted sum of total lateness
  - number of casting shifts used

Master problem

$$y_{c0} + y_{c1} \leq 1 \qquad \forall c \in C$$

$$\sum_{s_{p_i} \in S_p} x_{s_{p_i}} = 1 \qquad \forall p \in P$$

$$\sum_{p \in P} \sum_{\substack{s_{p_i} \in S_p \\ a \in s_{p_i}}} t_p x_{s_{p_i}} \leq C_a \qquad \forall a \in A$$

$$\sum_{\substack{p \in P: \\ t_p = 0}} \sum_{\substack{s_{p_i} \in S_p: \\ c \in s_{p_i}}} w_p \, x_{s_{p_i}} - W \, y_{c0} \leq 0 \qquad \forall c \in C$$

$$\sum_{\substack{p \in P: \\ t_p = 1}} \sum_{\substack{s_{p_i} \in S_p: \\ c \in s_{p_i}}} w_p \, x_{s_{p_i}} - W \, y_{c1} \leq 0 \qquad \forall c \in C$$

$$\sum_{s_{p_i} \in S_p} d_{s_{p_i}} x_{s_{p_i}} - L_p \leq d_p \qquad \forall p \in P$$

Pricing problem

- Each constraint has a dual variable
- For each product, pricing finds a schedule with minimum reduced cost
- Minimum reduced cost of a schedule of product p:

$$\min_{s_{p_i} \in S_p} \left\{ \sum_{a \in s_{p_i}} t_p \beta_a + \sum_{c \in s_{p_i}} w_p \gamma_{c,t_p} + d_{s_{p_i}} \delta_p - \alpha_p \right\}$$

- Pricing is solved using dynamic programming

Iterative rounding

- LP solution may be fractional
- Iterative rounding procedure:
  - select top K schedule variables with highest value
  - temporarily fix each to 1
  - keep the best candidate
  - re-run column generation
- Repeat until all variables are integral

Two-phase algorithm

- Phase 1: integer program to fix casting shifts
- Phase 2: schedule assembly operations only
- Advantages:
  - smaller LP
  - faster runtime
  - near-optimal solutions
  - secondary optimization can be added

Handling infeasibility

- Causes:
  - insufficient assembly capacity
  - fractional feasibility in Phase 1
- Mitigation strategies:
  - reduce assembly capacities (e.g. to 80%)
  - forbid problematic casting shift combinations

$$\sum_{c \in C'} \sum_{p \in P} x_{pc} \leq |C'| - 1$$

  - restart Phase 1 and Phase 2

Comparison of the methods

- **Column generation + iterative rounding:**
  - solutions close to LP optimum
  - slack assignments rare with lateness allowed
  - fastest method for combined objectives
- **Two-phase algorithm:**
  - efficient for single-objective or small instances
  - allows feasibility-only or secondary optimization
  - slower than iterative rounding for combined objectives

Declaration of the use of AI

Artificial intelligence tools were used in the preparation of this project and presentation.

In particular, I used Gemini and ChatGPT for the following purposes:

- checking and improving the grammar and clarity of the text
- formatting and structuring the LaTeX source code

Thank you for your attention!