Abstract

Seek to explore the computability of planning models in continuous production.

Intro

A production planning problem considers processing items on machines in time horizon **T**. An operation plan (or schedule) should be made to satisfy orders in time. This problem is critical in supply chain management, and it has traditionally been divided into multiple parts, including S&OP, AP (aggregate planning), MRP, and scheduling at the lowest level. None of these above can provide a comprehensive solution to production plan, inventory and delivery.

In 2016, I worked with Texas Instrument on a project to reduce WIP levels in their semi-conductor facilities, see Zhang et al. (2017). The precise scope of the project lies between MRP and AP, as it focuses on materials production, and inventory levels. Later in 2019 I worked with a large ICT company on an enormous planning problem, which considers over 100,000 products with complex BOM in a planning horizon over 3 months. The project should provide the company with an integrated plan on inventory, order delivery, production, procurement, out-sourcing, and so on, which makes it encloses S&OP, AP, and even procurement plans.

It seems that supply chain operations has discarded the methodology defined by traditional management tools. To translate strategic plan into operations, we nowadays prefer to consider what has been treated separately as a whole. It is partly the comes from the development and success of large-scale mathematical programming.

The keyword is computability. In the TI project, most instances define weekly plans, and can be solved within 3,600s by open-source linear optimizer *CLP*. For the project I did in 2019, we use LP models with over 1 billion decision variables, the computation time goes up to 3~4 hours. But they are all linear optimization models. Using LP means you cannot have integer variable, disjunctive constraints, SOS, and so on, which prevents you from modeling operation rules **precisely**. At the same time, fractional solutions are not acceptable for many cases, and you will need a good rounding heuristic to play with the integer issues. Generally, extra efforts should be placed on LP-specific constraints, systematic tuned hyper-parameters, and the designed solving procedure since you cannot model prioritized "multi-objective" requirements in LP.

Compared to continuous production, discrete manufacturers would like to pay more attention on using models to solve production planning problems, especially for industries like semi-conductor, ICT, mobile phones and so on. The reasons could still be the problem of computability. For discrete production, one usually wants to minimize inventory levels to achieve Just-in-

Time style solutions. The ideal way is to model the production problem by using time-indexed variables; the description of constraints seems to be quite straightforward. It looks like lot-sizing, assignment, and network flows, and can be solved by LP; at least a fractional amount of production makes some sense. You can find detailed reference from Wolsey's planning book[1]. Furthermore, a time-indexed model tends to produce very discrete solutions. A production plan would not be continuous, which is not acceptable in continuous production. The continuous companies would like to minimize setup costs and to keep running the same production, and this goal somehow contradicts the nature of time-indexed models. In this case many rules and descriptions of the problem require MILP-like constraints like routing of jobs, processing time of items and so on. This makes the problem similar to JSP and thus NP-hard.

The Discrete Model

Generally, the goal of a planning systems is to find a solution to plan for production, and ultimately to reduce delivery insufficiency. It can be summarized from high to low by priority: a. minimize production shortage, b. minimize inventory, c. minimize production cost, setup cost, etc.

Notation

We use the following notations.

- set of items: $\mathbf{I} = \{1, 2, ..., N\}$
- set of machines: $\mathbf{M} = \{1, ..., m, ..., M\}$
- demand/order: d_{it}
- lead time: l_i
- suppose there exists resource restrictions on machine r_m
- parents: J_i, a group of items use i\$ as components with rate u_{ij}

Decision:

- production: x_{it}
- delivery: s_{it}

A MILP/LP model can be defined as follows: t' defines the start time of production that finishes @t

---

[1]Pochet and Wolsey (2006)

$$N_{i,t} + D_{i,t} - \sum_{j \in J_i} u_{ij} X_{j,t} = N_{i,t-1} + X_{i,t'}, t' \leq t$$

This above constraint is usually defined as **flow**, i.e., how items are produced and the inventory flows into the whole system. For discrete system, the flow actually follows the BOM[2], which are generally a group of trees.

For continuous production, an item is built from a set of operations; for each item i, the building procedure can be represented as a directed acyclic graph $G_i = \mathscr{G}(O_i, E_i)$, where $O_i \subseteq O$ is the needed operations each of which has to be processed on one of the machines, E_i is the set of directed edges defining precedences. The lead-time here is defined from the processing time by matrix $P_i = \{p_{om}\}$, where p_{om} is the time to finish o on machine m. We will talk about this in the next section.

References

Pinedo M (2012) *Scheduling* (Springer).

Pochet Y, Wolsey LA (2006) *Production planning by mixed integer programming* (Springer Science & Business Media).

Zhang C, Bard JF, Chacon R (2017) Controlling work in process during semiconductor assembly and test operations. *International Journal of Production Research* 55(24):7251–7275.

---

[2]bill-of-material