Aggregation – the key to integrating production planning and scheduling

J. Váncza[1] (2), T. Kis[1], A. Kovács[2]
[1]Computer and Automation Research Institute, Hungarian Academy of Sciences
[2]Budapest University of Technology and Economics, Budapest, Hungary

Abstract

In this paper we suggest an integrated planning and scheduling framework with a special emphasis on the link between these control levels. Our planning model is generated automatically by performing aggregation on *de facto* standard product and technology related data in the dimensions of time, resource capacities and operations. The method addresses make-to-order production environments. An industrial case study is also presented, demonstrating how our algorithms work on large-scale problem instances.

Keywords:
Production, planning, scheduling

1 INTRODUCTION

Production planning and scheduling (PPS) map the load of a factory to its capacities on different time horizons and levels of detail. Planning and scheduling problems differ in their timescale, the granularity of resource and activity models and in their optimization criteria.

The two levels of PPS are strongly coupled since planning sets the goals as well as the resource and temporal constraints for scheduling. On the other hand, scheduling is responsible for unfolding a plan into detailed resource assignments and operation sequences. No scheduling strategy can improve much on an inadequate plan, whereas a bad scheduling strategy that wastes resources may inhibit the fulfillment of a good plan. All this makes PPS extremely complex and hard to solve. At the same time, PPS calls for efficient decision support methods and intuitive, flexible models with fast, reliable solution techniques that scale-up well to large problem instances.

Departing from detailed product, resource and production technology related information, **aggregation** connects the two levels by composing distinct resources and operations into larger units. Thereby aggregation reduces the complexity of production planning when deciding over the flow of materials and the use of resources on a longer horizon. Aggregation addresses also the various uncertainties of production: it does not allow generating detailed plans for a future that will certainly be different from what we anticipate now. Note that this principle applies well not only in PPS but also in the early stage of product design [1] and engineering [2].

The idea of aggregate production planning was introduced almost fifty years ago, just with the motivation to respond to fluctuations in product orders by means of a clear-cut mathematical model that used a common measure of work required by the different orders [3]. A number of theoretical models followed: they merged discrete operations requiring the same resources into distinct aggregate activities [4]. However, if the parts loop over the same resources several times, this method may result in very complex temporal interdependencies of the activities. Consequently, temporal patterns used by planning could hardly be filled in with technological data [5] [6], and planning disregarded most temporal relations [4]. In practice, so-called material/manufacturing requirements planning systems (MRP/MRP II) do work with precedence relations, but assume that components and complete products can be produced with fixed lead times, without any direct regard to capacities and the actual load. None of these assumptions is realistic in make-to-order production environments that respond to fluctuating orders. No wonder that plans generated this way can barely be refined to executable detailed schedules.

2 PROBLEM STATEMENT

In this paper, we consider aggregation as the process that provides a mapping between the models of PPS problems. Hence, it is the solution of a **representation problem** that has an essential impact both on planning and on scheduling. The main requirements towards aggregation are as follows:

- Planning must respect the main temporal constraints of orders (e.g., due dates) and the resource capacity constraints of the factory.
- Production plans should be unfoldable into executable schedules. Planning must also handle precedence relations that ensue from complex product structures (e.g., assemblies) and technological routings.
- However, resource assignment problems with finite capacities and precedence constraints are in general extremely hard to solve. Due to aggregation, typical instances of planning problems should be of the size and complexity that can be solved efficiently.
- Planning and scheduling models should be built by using common product and production data (e.g., bills of materials (BOMs), routings, resource calendars). Open orders should be addressed at both levels.
- Albeit aggregation removes details – due to the differences between the planning and scheduling model – it may introduce new constraints that distort the original problem. The effect of such constraints should be kept as small as possible.

The clarification of these interdependencies makes PPS methods more transparent and efficient. Hence, advanced PPS methods could better be applied in supply chain logistics, too [7]. We emphasize that the actual PPS frameworks used in factories should meet the above general requirements.

In what follows, we introduce shortly our PPS approach (Sect. 3), analyze the role of aggregation, define an optimal activity model, and provide algorithms to construct such activities in Sect. 4. Further, we present an industrial case study, whose lessons are summed up in Sect. 6.

3 INTEGRATED PLANNING AND SCHEDULING

3.1 Project-based production planning

Recently we have suggested a novel approach to modeling and solving production planning problems in make-to-order production environments [8]. This method – as other project-based planners [5] [6] – unifies the capacity and the material flow oriented aspects of planning. Orders are modeled as **projects** that compete for limited resources. Projects have time windows set by their release dates and deadlines. Each project is represented as a network of **activities** that are linked by various precedence constraints. An activity may require several **resources** and the execution of a given amount of work. However, the intensity of executing an activity may vary over time; the activity can even be pre-empted. Activities are **aggregates**: they represent groups of manufacturing, assembly, etc. operations, some of which are executed simultaneously, some sequentially, and others independently of each other. This leads to a model in which not the lead times but only the work amounts of activities are fixed a priori.

The required resources are typically both machine and human resources that should be shared by activities of different projects. The resources may be distributed, geographically dispersed and may even belong to different organizations. The **capacities** of resources are limited and may vary over time.

Plans are generated on a medium term horizon with one week's time unit. The solution, i.e. a production plan specifies what portions of the activities have to be performed and how much external resources must be used in each time unit so that all temporal, precedence and capacity constraints are respected. This solution can be optimized according to various cost and due-date performance criteria. In the running example, the primary objective is to minimize the cost of external resource usage, while the secondary objective is to keep work-in-process (WIP) level as low as possible.

Our production planner works with customized, powerful mathematical programming methods. The solver developed specifically for the project model efficiently solves problems of real-life sizes [8].

3.2 Constraint-based job shop scheduling

Our short-term scheduler performs finite capacity scheduling with respect to detailed technological constraints. The scheduling horizon is as long as the time unit of the planner (i.e., one week), while the scheduling time unit is 0.1 hour. The operation sets to be scheduled are given by the aggregate activities that fall into a given time unit in the medium-term production plan. Typically, schedules are generated for the next few weeks only. If an activity covers several weeks, then its operations are distributed in this period proportional to the activity's intensities.

There are both individual (e.g., machine tools) and group resources (homogeneous machine groups, assembly stations, various pools of qualified workforce). Resource availability – that may vary shift-by-shift – is given by a calendar. Resource and time requirements, as well as the sequence of operations are described in the routings. Each operation requires a given combination of resources. E.g., a turning operation might require a turning centre and a machinist during the entire length of its processing. Operations have specific processing, setup and transportation times.

The solution is an assignment of starting times to operations such that all temporal, precedence and resource constraints are satisfied. The main objective is to minimize the maximal tardiness with respect to the due dates set by planning. We obtain such a solution using a **constraint-based scheduling** approach [9].

4 AGGREGATION OF PROJECT TREES

4.1 The project tree

Each project can be described by a rooted tree, the so called **project tree** (denoted by T), whose vertices represent manufacturing operations (see Fig. 1). Vertices with several children denote assembly operations, while those with a single child represent either machining operations or joining a purchased part to the workpiece. The execution of the project over time advances from the leaves towards the root that stands for the finishing operation of the final product. Edges represent strict precedence relations, i.e., the sons of an operation must all be completed before the operation itself could be started.

Each operation i of the project tree has a processing time p_i, setup time s_i and transportation time u_i. Resources needed by i are given by $R_i \subseteq R$ where R is the set of all available resources. We assume that transportation and setup is performed before the operation, but while the first needs the part only, setup requires solely the resources of the operation.

Fig. 1 below shows the parameters and structure of a sample project. The root operation #1 is a manual assembly that produces the final product. The project tree contains also other assembly operations (#3 and #8) while the other nodes correspond to various machining operations. For instance, operations #1 and #3 need the same human and machine resources, actually a welding cabinet with appropriate personnel.

i	p_i	s_i	u_i	R_i
1	12.0	3.0	0.5	M1, H1
2	6.0	1.0	0.5	M2, H2
3	9.5	3.5	0.5	M1, H1
4	8.0	1.0	0.2	M3, H2
5	6.0	1.0	0.2	M2, H2
6	9.0	1.0	0.2	M3, H2
7	2.5	1.0	0.5	M2, H2
8	5.0	2.0	0.5	M4, H1
9	9.0	1.0	0.5	M2, H2

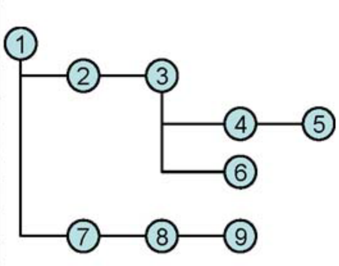

Figure 1: A sample project.

4.2 Activity models of the project

At aggregation, connected vertices of the project tree are contracted into components that define the activities of the planning model. This partitioning of the project tree is called the **activity model** of the project. If two operations of the project tree that are connected by a precedence