Chapter 2
Production Planning and Scheduling: Interaction and Coordination

Yiwei Cai, Erhan Kutanoglu, and John Hasenbein

2.1 Introduction

In many organizations, production planning is part of a hierarchical planning, capacity/resource allocation, scheduling and control framework. The production plan considers resource capacities, time periods, supply and demand over a reasonably long planning horizon at a high level. Its decision then forms the input to the more detailed, shorter-term functions such as scheduling and control at the lower level, which usually have more accurate estimates of supply, demand, and capacity levels. Hence, interaction between production planning and production scheduling/control is inevitable, not only because the scheduling/control decisions are constrained by the planning decisions, but also because disruptions occurring in the execution/control stage (usually after schedule generation) may affect the optimality and/or feasibility of both the plan and the schedule. If the overall performance of the production system is to be improved, disruptions must be managed effectively, with careful consideration of both planning and scheduling decisions. This chapter focuses on the interaction between production planning and scheduling, emphasizing the coordination of decisions, with special emphasis on making robust decisions at both levels in the face of unexpected disruptions. We provide examples and realistic scenarios from semiconductor manufacturing.

To capture the interaction between production planning and scheduling, we suggest an intermediate model between the two levels. One can view this as a lower-level planning model or a higher-level scheduling model, but ultimately it provides a middle ground between the two levels of the decision-making framework. In many systems, the longer-term, aggregated production plan is used to facilitate scheduling. This is usually achieved by creating specific work orders or jobs of different product types that collectively resemble the output required by the production plan and generating a release schedule for the jobs. For example, semiconductor manufacturers

E. Kutanoglu (✉)
Graduate Program in Operations Research and Industrial Engineering,
Department of Mechanical Engineering, The University of Texas at Austin, Austin,
TX 78712, USA
e-mail: erhank@mail.utexas.edu

K.G. Kempf et al. (eds.), *Planning Production and Inventories in the Extended Enterprise*, International Series in Operations Research & Management Science 152,
DOI 10.1007/978-1-4419-8191-2_2, © Springer Science+Business Media, LLC 2011

rely on detailed simulation-based models to fine tune the release schedule (also referred to as “wafer starts”), which ultimately determines the product mix in the wafer fab. The scheduling function typically dispatches the jobs according to their perceived or assigned priorities to align the processing sequence of the jobs with the production plan while using the latest information on job and machine availabilities. Dispatching inherently utilizes local information (typically one-job, one-machine at a time) to make a decision. It is very hard, if not impossible, to calculate the effects of an individual dispatching decision on the long-term system performance, or even on the performance of the upstream and downstream machines on the shop floor. The idea of an intermediate model between planning and scheduling is to provide additional useful information to the scheduling system, whether it is dispatching based or otherwise.

With the interaction and coordination between production planning and scheduling being the main theme of this chapter, we first review the literature, focusing on a selected set of production planning and scheduling-based papers. In Sect. 2.3, we present two versions of our approach that attempt to fill the gap between the classical planning and dispatching-based scheduling models using an intermediate decision model. Section 2.4 describes our computational study with a simplified reentrant system that represents a small wafer fabrication facility. Section 2.4 also discusses the implementation issues that must be addressed to properly compare the proposed approach with the conventional planning-to-scheduling approach, using this mini-fab model. In Sect. 2.5, we present the experimental results focusing on insights that can be obtained from our preliminary experiments. We conclude with a summary and future research directions in Sect. 2.6.

2.2 Literature Review

We first review the production planning models that are representative of the existing models in the literature. This review is not exhaustive by any means, and our main focus is on the interaction between planning and scheduling, not necessarily on the other aspects of the models, such as batching or setups, that may be potentially critical. We also try to give examples and applications from the literature on semiconductor manufacturing to make the discussion more concrete, but the methods discussed in this section also pertain to more discrete parts manufacturing systems. Due to increasing competition and the rapid development of technology, manufacturing managers, especially those in semiconductor industry focus strongly on cycle time, which is defined as the time between the release of an order to the shop floor and its completion time of that order. Long cycle times imply a high work-in-process (WIP) level, and thus high inventory costs. Therefore, we choose cycle time as the main performance measure with which we evaluate the effectiveness of the coordination between planning and scheduling. This choice further limits our literature review to studies that emphasize cycle time as a potential issue to be addressed between planning and scheduling.

2.2.1 Production Planning Models

An extensive literature on production planning has been developed over almost five decades. In this section, we focus on only a few of these optimization models. Interested readers are referred to the chapter by Missbauer and Uzsoy (2010) in the first volume of this handbook which reviews the basic formulations that are most commonly used in academic research and industrial practice.

A capacitated Material Requirement Planning (MRP)-based model is proposed by Horiguchi et al. (2001). The goal is to calculate a planned release date for each order during each of its visits to a bottleneck station, and to estimate when the order will be completed. The authors aggregate the times available across machines over discrete time periods (time buckets) that are used to incorporate capacity factors. The model explicitly considers capacity only for specified near-bottleneck stations, and assumes that all other stations have infinite capacity, which is different from the conventional MRP approach. They perform two experiments. One examines the effect of the predictability of the capacity model. In their paper, predictability is defined as the deviation of the realized completion time in the simulation model from the predicted completion time in the planning model. The results show that finite capacity planning gives better predictability than dispatching rules such as Critical Ratio (Rose 2002). The lot with the lowest value has the highest priority. The second experiment tests the effects of using a "safety capacity" in planning, that is, the reduction of the planned capacity of a given station by some amount to keep processing capacity in reserve to deal with unexpected events such as machine breakdowns. Their results show that increasing the safety capacity reduces tardiness and improves predictability, without adversely affecting other performance measures.

There are a wide variety of linear programming-based planning models for production planning. Hackman and Leachman (1989) propose a general production planning framework based on a linear programming model. They take into consideration specific components such as processing and transfer time in order to provide an accurate representation of the production process. However, the time delays in the model do not capture the load-dependent nature of the lead times. Thus, the aspects of production captured in the model are limited. In addition, the LP formulation accommodates noninteger values for cycle times as well as planning time buckets of unequal length. Expanding the model in Hackman and Leachman (1989), Hung and Leachman (1996) incorporate time-dependent parameters representing partial cycle times from job release up to each operation into the LP planning model. Furthermore, they provide a framework that iteratively updates the plan through an LP model that develops a plan for a given set of lead times and a simulation model that evaluates the system performance for a given production plan. They estimate the cycle times from the simulation results and show that they can achieve better results by iterating between the LP and the simulation model. It is well known that the relationship between cycle time and machine utilization is nonlinear. Therefore, the iterative LP-simulation process provides a good way to approximate such a nonlinear relationship. The process stops when satisfactory agreement in cycle times is achieved. Hung and Leachman's experiments with deterministic and random machine breakdowns indicate that the difference between the LP and simulation cycle times can be reduced to 5% or less within a few iterations. However, other researchers have not been able to easily replicate such results, and our own research indicates that similar results are difficult to achieve. In particular, the convergence behavior appears to be unpredictable, and is certainly not well understood.

Some planning models that try to capture the relationship between cycle time and utilization without resorting to simulation make use of so-called clearing functions (Graves 1986). Clearing functions express the expected throughput of a machine in a planning period as a function of the expected WIP inventory at the machine over the period. (Our focus on load-dependent cycle times is mainly due to the observation that the load as a function of releases determined as part of the plan affects cycle times that result in scheduling. There are recent studies that try to capture the dependency between the load level and/or utilization and cycle times. In that sense, the underlying approaches can also be viewed as "hybrid" models attempting to link planning and scheduling.) Missbauer (2002) considers clearing functions for an M/G/1 system. He uses a piece-wise linear approximation for the clearing function to model the effective capacity for bottleneck stations, and considers fixed, load-independent time delays between bottleneck stages to represent the delays at nonbottleneck machines. His planning model determines the release plan and uses a short-term order release policy to select specific orders for release into the job shop. The product mix is not considered in their clearing function, i.e., the clearing function only depends on the total planned production quantity, which means the total output of a station can be allocated arbitrarily to different products.

Asmundsson et al. (2006) propose a clearing function-based planning model, which explicitly considers the product mix. They approximate the clearing function using an empirical approach, together with two sets of constraints enforcing flow conservation for WIP and finished goods inventory (FGI). There is no need for explicit cycle time parameters in their model. Due to the product mix, different products may have different capacity needs (capacity allocation), and a particular difficulty is estimating the throughput as a function of the product mix currently represented in the WIP. To overcome this, they assume that all products see the same average cycle time, which allows them to use a convex combination of the capacity allocation parameters to approximate the WIP levels of different products, which in turn leads to approximated clearing functions. Exploiting the concavity of clearing functions, they use outer linearization to approximate the functions, which results in an LP model. The objective in their model is to minimize the total production cost, the WIP cost, the FGI holding cost, and the raw material cost. The approximation of the clearing function is also done by simulation with several randomly generated realizations of the demand profile, which are evaluated using the release schedules obtained from the fixed cycle time production planning model of Hackman and Leachman (1989). They perform extensive experiments to evaluate the benefit of the clearing function-based model. Different dispatching rules are used to compare planned throughput and actual throughput. Based on these experiments, one of their conclusions is that:

> "If planning is done properly, the role of a detailed schedule can be viewed as rescheduling the jobs to adhere to the original production plan that has been distorted by equipment failures and other unpredictable occurrences. Although it is unlikely that the scheduler can restore the original plan in every instance, its ability to do so is highly dependent on the planning algorithm's ability to represent the shop floor dynamics correctly."

Such a conclusion shows that we need to consider the coordination between planning and scheduling to achieve better performance, which further motivates our study.

Pahl et al. (2005) give an extensive survey of planning models, which consider load-dependent cycle times. In addition to the use of clearing functions, there are other approaches. Interested readers are referred to Sect. 3 and the corresponding references in Pahl et al. (2005) for more details.

Model predictive control, or MPC (Qin and Badgwell 2003), is a method of process control that has been used extensively in processing industries (Kleindorfer et al. 1975). MPC encompasses a group of algorithms that optimize the predicted future values of the plant output by computing a sequence of future control increments. This optimization model is implemented through a rolling-horizon approach at each sampling time. MPC attempts to model the dependence between the sequence of predicted values of the system output, and the sequence of future control increments. With knowledge of the system model, disturbance measurements, and historical information of the process, the MPC model calculates a sequence of future control increments that must satisfy appropriate constraints. Vargas-Villami and Rivera (2000) propose a two-layer production control method based on MPC. Extending this work, Vargas-Villami et al. (2003) propose a three-layer version. The first layer, called the adaptive layer, is used to develop a parameter estimation approach. The second layer, the optimizer, solves an MILP model by branch and bound to generate a good-quality production plan. The third layer (direct control) uses dispatching to control the detailed discrete-event reentrant manufacturing line in a simulation model. The computational results show that the method is less sensitive to initial conditions than "industrial-like" policies examined by Tsakalis et al. (2003). Furthermore, the three-layer approach with the adaptive parameter estimation model achieves reduced variation at high production loads as compared to the two-layer approach. They did not perform any cycle time comparisons with existing methods, but point out that an MPC-based model could be a promising tool for planning.

Jaikumar (1974) proposes a methodology, which decomposes the planning and scheduling problem into two subproblems. The first problem is a long range planning problem, which maximizes the profit subject to resource constraints. The Lagrange multiplers obtained in the first problem are used in the objective function of the second short range scheduling model. They propose a heuristic algorithm to reduce the second model to a sequential allocation of production facilities to products.

2.2.2 Scheduling Models

Apart from planning models, there are models and studies that focus on scheduling only. Leachman et al. (2002) summarize their effort, dubbed Short Cycle Time and Low Inventory in Manufacturing (SLIM), to improve scheduling at Samsung Electronics. They try to control the production line by managing the WIP level at particular bottleneck stations. A dispatching-based method is used to achieve target WIP levels at the bottleneck stations. An important task is to determine appropriate target WIP levels. Based on an overall target cycle time for each product, the total buffer time is calculated as the difference between the target cycle time and theoretical (or raw) cycle time. The total buffer time is then proportionally allocated to each bottleneck step. The resulting buffer time allocation is in turn used to estimate the target WIP levels using Little's Law. By using different strategies according to the characteristics of different production stages (bottleneck, batching, nonbottleneck steps, etc.) and prioritizing the jobs in different stages to meet target WIP levels (and hence bottleneck utilization) as closely as possible, they show that one can reduce average cycle times.

In contrast to the fab-wide approach in Leachman et al. (2002), other authors focus on bottleneck steps only, with explicit controls the WIP level. Lee and Kim (2002) try to implement WIP control at bottleneck steps to balance a production line in semiconductor manufacturing. Assuming a given target throughput rate, they calculate the buffer time and the associated target WIP level. The work focuses on short-term scheduling for the steppers, which are usually the bottleneck machines in most wafer fabrication facilities. One of the two proposed MILP models minimizes the total weighted deviation from the target WIP level, and the other maximizes the total wafer production for all steppers in the hope that this will lead to high utilization of bottleneck machines.

Kim et al. (2003) use a similar idea to determine a single-shift schedule for the steppers for a given WIP status. Using an MILP formulation, they try to maintain the WIP levels close to the "desired" levels so that the flow of material through the factory is balanced. The objective is to meet the predetermined WIP targets. Three proposed heuristics to solve the underlying MILP model can find schedules within 5% of the optimum values in a reasonable amount of time.

Queueing theory is useful in scheduling in several ways. One way is to analyze the stability (that is, boundedness of average WIP) of scheduling policies. Another way is to model manufacturing systems with multiclass queueing networks to develop scheduling policies. Generally speaking, queueing theory is based on long-term steady-state analysis and may not be optimal in a finite period. However, the following two papers implement queueing theory using fluid models that focus on transient analysis and are a reasonable approximation for short periods of time. Dai and Weiss (2002) develop a fluid-relaxation-based heuristic to minimize makespan in job shops. In a fluid model, discrete jobs are replaced with flow of a continuous fluid and machines are replaced with valves that affect the flow rate of the of fluid. The proposed online (dispatching-based) heuristic uses safety stocks for WIP and tries to keep the bottleneck machine busy at almost all times, with the idea that the nonbottleneck machines are paced accordingly. The heuristic is constructed in three steps: (1) reduce the job shop problem to a reentrant line scheduling problem, which has the same lower bound; (2) define an infeasible backlog schedule that keeps the bottleneck machine busy (here the schedule is infeasible because a machine is allowed to start work on a job step even if the previous step of the job on a different machine has not been completed); (3) introduce safety stocks to make the backlog schedule feasible.

Similarly, Bertsimas et al. (2003) use a fluid model to solve a job shop scheduling problem with the objective of minimizing the holding cost. The proposed algorithm uses the optimal fluid solution as a guide. Comparison with several other commonly used heuristics shows that the proposed algorithm outperforms the other heuristic methods.

This section provided a brief review of planning and scheduling models with a focus on the interaction between the two models. We note that most of the planning models in the literature do not consider WIP allocation across stations, while many scheduling models consider WIP level explicitly. Therefore, we propose an approach in Sect. 2.3 to explicitly consider WIP allocation in an intermediate model called high-level scheduling, and try to control WIP allocation in a way which facilitates the coordination between planning and scheduling.

2.3 Coordination of Planning and Scheduling

2.3.1 Overall Approach

In most manufacturing companies, the planning and scheduling functions belong to two different departments, as planning is viewed as a tactical activity and scheduling as more operational. Sometimes it is necessary to separate planning and scheduling because it is almost impossible to obtain a comprehensive system-wide solution that encompasses both planning and scheduling concerns. Such a model would have to provide detailed decisions for each machine in each period. In general, it is impossible to solve such a model since it is inherently too complex with too many constraints and variables. Hierarchical decomposition into planning and scheduling provides an easy way not only to obtain reasonable solutions to both subproblems, but also to generate decisions aligned with the current organizational structure between planning and scheduling functions.

However, the conventional hierarchical separation of these two functions may cause several problems. One drawback is that a solution that is good at the planning level might not be easy to implement as a detailed schedule; the plan may not be even feasible when the scheduling issues are explicitly considered. One reason behind this is that the dynamics of the production system are modeled at an aggregate level, and detailed execution may be infeasible even if aggregate constraints are satisfied. Another issue is that objectives are usually different between planning and scheduling. The planning function focuses more on how to meet the demand, and reduce inventory and backorders, while scheduling emphasizes more operational measures such as minimizing cycle times and maximizing bottleneck utilization. Ideally, if the hierarchical decomposition were done properly, the objectives of the two levels would be aligned. However, due to the complexity of the overall problem, the computational effort involved in solving the two levels to optimality, and the differences in preferences between the organizational areas representing the two levels, this is not the case in the real world.

In this chapter, we test an idea that seeks to overcome the drawbacks of such separation of planning and scheduling. The idea is to introduce an "intermediate" module between planning and scheduling, which overall modifies the conventional hierarchical approach. The goal is to solve the discrepancy in objectives between production planning and detailed scheduling. In the following, we define "normal planning" as a planning model that does not consider WIP levels explicitly, and a "high level scheduling" model as one that does. In this approach, we have a normal planning model (denoted by "P") and a high level scheduling model (denoted by "H") that both feed into detailed scheduling (denoted by "D") (See Fig. 2.1). To provide a framework for our discussion, we represent each "stage" in the process with an associated model: the planning model, the high-level scheduling model, and the detailed scheduling model. In fact, in the following discussion, the first two models are linear programming problems and the last one is a simulation model of the system that represents the actual implementation of the plan and schedule.

First, we explain the proposed version of the P–H–D approach: The *planning model* tries to meet the demand while minimizing inventory and backorder costs. The output of the planning model specifies how much of each product should be produced by the end of each period. Then, we use that output as a modified demand profile, which is input to the high level scheduling model, which explicitly tries to minimize the average WIP level, and thus the average cycle time. The *high level scheduling model* determines the release policy and processing targets for each product in each period, which form the input to detailed scheduling. Figure 2.1 compares the proposed P–H–D approach with the P–D approach.

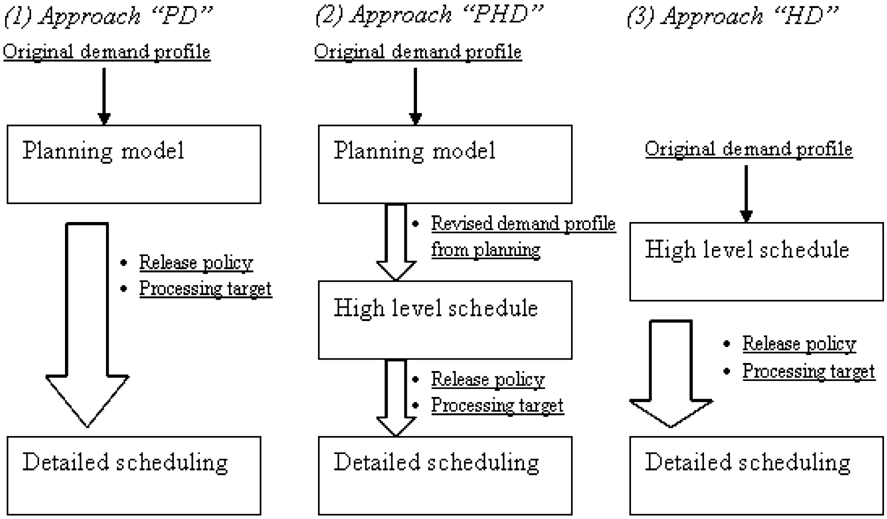

Fig. 2.1 Three approaches for planning and scheduling

Consideration of the three potential levels in the overall process gives rise to another structure, in which the planning step is skipped: the original demand data is fed directly to the high-level scheduling model whose output is input to the detailed scheduling model. Intuitively, using a high level schedule without a planning model consideration may achieve lower cycle times, but may result in higher costs since it pays more direct attention to WIP levels than to costs. We now discuss our mathematical formulations for the planning and high level scheduling models.

2.3.2 *Planning Model*

For the planning model, we use a fixed cycle time version of a well-known linear programming model found in Hung and Leachman (1996). This model assumes that the next planning horizon is divided into equal length time periods (or time buckets, e.g., representing individual shifts). The planning horizon is assumed to be long enough to capture varying demand levels (especially across product types) over time. We assume that we have accurate forecasts for demand levels for each product type, say, in every week, i.e., every 14 shifts. (To be consistent with the length of the time bucket used in the model, the demand profile used in the experiments has nonzero demand at the end of every 14 shifts). We finally assume that the production process is divided into production stages or steps, each of which represents a unique operation to be performed on a particular machine group (station). We first introduce the notation that supports the model.

2.3.2.1 Sets

I: set of products, indexed by i

T: set of time periods (say, shifts), indexed by t

K: set of processing steps $= \{1, ..., \kappa\}$, indexed by k, where κ is the number of steps, assumed to be the same for all products

M: set of stations, indexed by m

2.3.2.2 Input Parameters

$p_{i,m,k}$: processing time of product i on station m at step k (say, in minutes)

c_m: available running time of station m in one shift ($c_m = 12$ h for all m in our model)

$d_{i,t}$: demand for product i in shift t (in number of jobs), assumed to be nonzero every 14 shifts with the availability of weekly forecasts and 2 shifts per day

$f_{i,k}$: average partial cycle time for product i to finish step k (estimated from simulation results or historical data), i.e., the average difference between the time when the job is released and the time it finishes step k(in shifts)

u: length of one shift (12 h)

$q_{i,k}$: smallest integer greater than $f_{i,k}$, i.e. $q_{i,k} = \lceil f_{i,k} \rceil$
b_i: unit backorder cost for product i, set to a large number to discourage backorders
h_i: cost for holding one unit FGI of product i for one shift
$\delta_{i,k}$: coefficients used in constraints (2.4) for product i at step k (we explain this term later in detail)

2.3.2.3 Decision Variables (All Variables Are Nonnegative)

$D_{i,t,k}$: amount of product i that depart from step k in shift t (in number of jobs)
$R_{i,t}$: amount of product i released in shift t (in number of jobs)
$B_{i,t}$: backorder for product i in shift t (in number of jobs)
$I_{i,t}$: inventory of product i at the end of shift t at the end of production line (in number of jobs).

2.3.2.4 Model Formulation

$$\min \sum_{i \in I} \sum_{t \in T} (\mathrm{e}^{-t} R_{i,t} + h_i I_{i,t} + b_i B_{i,t})$$

subject to

$$\sum_{i \in I} \sum_{k \in K} p_{i,m,k} D_{i,t,k} \leq c_m \forall t \in T,\ m \in M \tag{2.1}$$

$$D_{i,t,\kappa} - I_{i,t} + I_{i,t-1} - B_{i,t-1} + B_{i,t} = d_{i,t}\ \forall i \in I,\ 1 < t < |T| - 1 \tag{2.2}$$

$$D_{i,t,\kappa} - B_{i,t-1} + B_{i,t} = d_{i,t} \forall i \in I,\ t \geq |T| - 1 \tag{2.3}$$

$$\delta_{i,k} R_{i,t-q_{i,k}} + (1 - \delta_{i,k}) R_{i,t-q_{i,k}+1} = D_{i,t,k} \forall i \in I,\ k \in K. \tag{2.4}$$

In this model, the objective is to minimize the total costs of releases, inventory, and backorders, across all products and shifts. The discounted raw material release costs are used to release raw material into the factory as late as possible so as to indirectly manage the WIP in the factory. Constraints (2.1) limit the capacity of each station with the given amount of time. Constraints (2.2) and (2.3) ensure that the end product demand in each period is either met by finished product inventory or backlogged.

Constraints (2.4) capture the dynamic properties of the cycle time. From simulation results or historical data, we can estimate $f_{i,k}$, which is the average partial cycle time for product i to finish step k (from the beginning of first step of the product until and including step k). Thus, we consider the production process as a fluid model, and estimate the relation between the released quantity and the departing quantity by backtracking the production flow along the time horizon. Figure 2.2 shows the details of the relationship between $R_{i,t}$ and $D_{i,t,k}$.

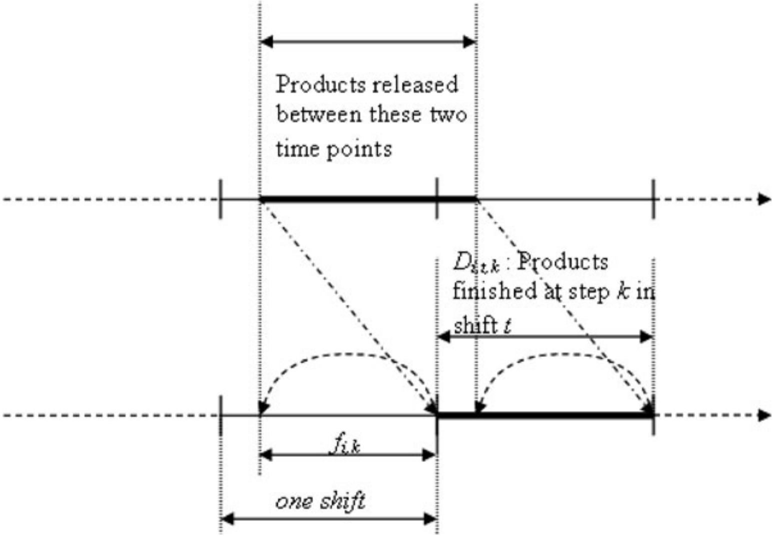
Products released between these two time points
$D_{i,t,k}$: Products finished at step k in shift t
$f_{i,k}$
one shift

Fig. 2.2 Partial cycle times, and the relationship between $R_{i,t}$ and $D_{i,t,k}$

All products of type i finished at step k in shift t, denoted by $D_{i,t,k}$ should have been released $f_{i,k}$ time units ago. Thus, by backtracking we can determine the time when these products are released. In Hung and Leachman (1996), a formula is provided to address all the cases where the cycle time is either longer or shorter than the length of the time bucket. Since we use a constant length for each time bucket, and in our mini-fab model (see Sect. 2.4.1) the partial cycle times are less than the length of the time bucket (shift), $D_{i,t,k}$ is composed of two parts. Thus, the relation between the products released, $R_{i,t}$, and $D_{i,t,k}$ is as follows:

$$D_{i,t,k} = \delta_{i,k} R_{i,t-q_{i,k}} + (1 - \delta_{i,k}) R_{i,t-q_{i,k}+1} \; \forall i \in I, \; 1 \le t \le |T| - 1, \; k \in K,$$

where the portions of releases to be completed by step k are estimated by $\delta_{i,k}$ and $(1 - \delta_{i,k})$. Here, we use $\delta_{i,k}$'s proportional to time: $\delta_{i,k} = (f_{i,k} \bmod u)/u$, $\forall i \in I$ and $\forall k \in K$.

2.3.3 High Level Scheduling Model

As mentioned before, the purpose of the high level scheduling model is to find a balance between the cycle times and the inventory/backorder costs. Thus, its objective function and constraints should take both factors into consideration. Although it is hard to model cycle times directly, we know from Little's Law that they are proportional to WIP levels for a fixed throughput. Thus, we consider the WIP level instead and try to represent minimization of cycle times by minimizing the WIP levels at the end of each shift.

In this model, the relationship between releases and departures is captured in the same manner as in the planning model. The main difference is that we explicitly capture WIP levels and their distribution across stations and stages of production in detail. In this model, we manage to keep a certain level of WIP at bottleneck station to prevent its starvation, which may lead to reductions in the throughput rate. These levels are formulated using a WIP control constraint that allows the WIP to be within a tolerance around the target value. To enable the model to build up WIP to meet the target level, the LP model enforces the WIP control constraints only after a certain number of periods elapsed. With this in mind, we introduce additional notation.

2.3.3.1 Sets

K_B: set of steps in the bottleneck station

2.3.3.2 Input Parameters

ϵ^+, ϵ^-: upper and lower tolerance of WIP control (currently set at $\epsilon^+ = 1.1$ and $\epsilon^- = 0.9$)
v_i: target cycle time for product i (in minutes)
γ_i: total raw processing time for product i (in minutes)
y_i: total buffer time for product i, i.e. the difference between the v_i and γ_i (in minutes)
$z_{i,k}$: target WIP level of product i at station k (in number of jobs)
$\zeta_{i,k}$: the average cycle time for product i to travel to the kth bottleneck step from its previous bottleneck step (in minutes)
λ_i: average demand rate for product i (in jobs per minute)
t_s: the last shift in which the WIP control constaints are not enforced; the high level scheduling model is assumed to be in steady state after shift t_s

2.3.3.3 Decision Variables

$W_{i,t,k}$: WIP level of product i at step k at the end of shift t (in number of jobs).

2.3.3.4 Model Formulation

The calculation of the target WIP level is adapted from the method proposed by Leachman et al. (2002), which uses Little's Law (as will be explained in detail later). They use target WIP levels as criteria for dispatching. Here, we incorporate the target WIP levels into the high level scheduling model. The target WIP calculation allocates the buffer time to bottleneck steps in proportion to the partial cycle time between two consecutive bottleneck steps. Cycle times are estimated from a simulation model of the mini-fab in our experiments, but practice they can be obtained from historical data. One can calibrate the estimated cycle times through multiple simulation runs, but here we obtain cycle times in a single run. We first compute the buffer time as the difference between the target cycle time and the theoretical (or raw) cycle time (which consists of only processing times) for each product:

$$y_i = v_i - \gamma_i \ \forall i \in I.$$

To allocate this overall buffer time (slack) into bottleneck steps, we compute the time between two consecutive bottleneck steps of each product:

$$\zeta_{i,k} = f_{i,k} - f_{i,k'} \ \forall i \in I,\ k,\ k' \in K_B,$$

where k and k' are two consecutive bottleneck steps.

Finally, we set the target throughputs equal to the average demand rates, allocate the overall buffer time and convert the allocations to target WIP levels as follows:

$$z_{i,k} = \lambda_i \cdot y_i \cdot \frac{\zeta_{i,k}}{\sum\limits_{k' \in K_B} \zeta_{i,k'}} \ \forall i \in I,\ k \in K_B.$$

The final mathematical model is as follows:

$$\min \sum_{i \in I} \sum_{t \in T} \left(\sum_{k \in K} W_{i,t,k} + h_i I_{i,t} + b_i B_{i,t} \right)$$

subject to

$$\sum_{t \in T} \sum_{k \in K} p_{i,m,k} D_{i,t,k} \leq c_m, \forall t \in T,\ m \in M \tag{2.5}$$

$$D_{i,t,\kappa} - I_{i,t} + I_{i,t-1} - B_{i,t-1} + B_{i,t} = d_{i,t}, \forall i \in I, 1 < t < |T| - 1 \tag{2.6}$$

$$D_{i,t,\kappa} - B_{i,t-1} + B_{i,t} = d_{i,t}, \forall i \in I,\ t \geq |T| - 1 \tag{2.7}$$

$$R_{i,t} - D_{i,t,1} + I_{i,t,1} = W_{i,t,1}, \forall i \in I,\ t \geq 1 \tag{2.8}$$

$$D_{i,t,k-1} - D_{i,t,k} + I_{i,t,k} = W_{i,t,k}, \forall i \in I,\ t > 1,\ k > 1 \tag{2.9}$$

$$\sum_{i \in I} W_{i,t,k} \leq \epsilon^+ \cdot \sum_{i \in I} z_{i,k}, \forall t \in T,\ k \in K_B \tag{2.10}$$

$$\sum_{i \in I} W_{i,t,k} \geq \epsilon^- \cdot \sum_{i \in I} z_{i,k}, \forall t \in t,\ k \in K_B \tag{2.11}$$

$$\delta_{i,k} R_{i,t-q_{i,k}} + (1 - \delta_{i,k}) R_{i,t-q_{i,k}+1} \geq D_{i,j,k}, \forall i \in I,\ 1 \leq t \leq t_s\ k \in K \tag{2.12}$$

$$\delta_{i,k} R_{i,t-q_{i,k}} + (1 - \delta_{i,k}) R_{i,t-q_{i,k}+1} = D_{i,j,k}, \forall i \in I,\ t_s < t \leq |T| - 1\ k \in K. \tag{2.13}$$

The main difference between this model and the previous planning-based one is that we now have WIP control constraints for the bottleneck steps. Here, we first calculate the target WIP level for each step in the bottleneck station, and then use the target WIP level in the LP model. Leachman et al. (2002) use a dispatching rule to execute WIP control for individual buffers (product and step), while in our LP model we control the total WIP level across all products at the same step in the bottleneck station. The reason is that we think the purpose of setting target WIP levels is to keep feeding the bottleneck station, so we only need to track the total WIP level across all products at the same step in the bottleneck station instead of controlling WIP levels for individual buffers.

2.4 Experimental Study

To test the idea of high level scheduling, we perform several experiments on a three-station six-step hypothetical production system, which represents a small wafer fabrication facility (mini-fab). Below, we first describe the mini-fab system, and then present the experimental results. These experiments focus on (1) evaluating the overall merit of the P–H–D approach, (2) testing the impact of different cost settings, and (3) testing the impact of machine breakdowns at nonbottleneck stations on the cycle times and cost. For testing, we run the mathematical models that represent the planning and/or high-level scheduling problems with the same inputs, and then evaluate the system performance (cycle time, WIP levels, costs) with a simulation model that imitates the implementation of decisions made by the mathematical models. For this, the simulation model relies on a dispatching-based methodology that tries to follow the release and production targets set by the outputs of the mathematical models. Thus, the dispatching logic in the simulation model acts as a detailed scheduling system.

2.4.1 Mini-Fab Model

The three-station six-step mini-fab model is depicted in Fig. 2.3. There are two different products, each of which must complete six operational steps. Each step is to be performed by a machine at one of the stations. The process flows (routings), which are the same for both products, are also shown in Fig. 2.3.

Table 2.1 shows the basic settings of the raw processing times for each product at each step (in minutes). The last column shows the total raw processing time (RPT) for both products. We consider a time horizon of 150 days, each day with 2 shifts of 12 h, leading to a 300-shift long horizon. All the processing times are assumed to be deterministic.

In the experiments with machine breakdowns, the first station's machines may fail. Times between failures follow an exponential distribution with a mean of 42 h. The repair times follow an exponential distribution with a mean of 45 min. The base

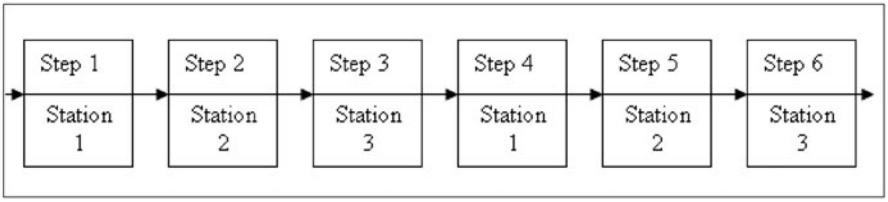

Fig. 2.3 Six-step three-station mini-fab model

Table 2.1 Raw processing time for the mini-fab model

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Total RPT
Product 1	47.5	30	75	40	52.5	30	275
Product 2	38	24	60	32	42	24	220

Table 2.2 Traffic intensity (expected utilization) for all stations

Traffic intensity	Station 1	Station 2	Station 3
No machine breakdowns	0.78	0.74	0.94
With machine breakdowns	0.80	0.74	0.94

demand rates are 55 jobs per week for product 1 and 44 jobs per week for product 2. The backorder cost is $50 per unit per shift, and the inventory cost is $1 per unit per shift. We vary these values in the next section to evaluate impacts from different factors.

Table 2.2 gives the traffic intensities (or utilizations) with and without machine breakdowns for all stations. We observe that station 3 is an overall bottleneck, and even when breakdowns are considered, the bottleneck station does not change.

2.4.2 Simulation Settings

To evaluate the performances of the different approaches in combining planning and scheduling, we use a simulation model that imitates the execution of the planning/scheduling decisions using a dispatching-based methodology. Here, the simulation model represents the actual system, where the planning/scheduling decisions are executed at the detailed scheduling level, during which the actual costs and performances are incurred and measured. The simulation model first creates production jobs according to the release schedule obtained in the mathematical model being used. It then dispatches the jobs at the machines with some level of adherence to the planning and scheduling decisions. Although there are several ways to carry out dispatching for a given plan/schedule, we apply two different dispatching rules: First-in-first-out (FIFO) and “target following.” FIFO dispatches the job that arrives at the station earliest, when a machine at the station becomes free, and is used primarily as a benchmark. The target following rule tries to follow the planning or high level scheduling production targets (captured by the optimal values of departure variables in both models) as closely as possible. Since the solution to the high-level scheduling LP model provides the processing targets for each product at each step by the end of each shift, the target following rule gives the highest priority to the buffer (product/step combination), which is the most behind (or least ahead) of its cumulative processing target.

The simulation run length is determined by the completion of all jobs that are released during the time horizon, 300 shifts. To obtain stable statistics on cost and other performance measures such as cycle times, the first and last 100 jobs are discarded. For the experiments with machine breakdowns (i.e. the scenarios in which the first station may break down), we run 50 replications with stochastic machine failure times and repair times to obtain the statistics.

2.4.3 Implementation

There are several issues to be addressed regarding the simulation experiments, especially in terms of how the planning and scheduling decisions from the mathematical models are converted to inputs for simulation testing. One issue is how to convert the release policy to a usable input for simulation. The LP model solution produces a noninteger numbers of jobs to be released during each shift, which should be converted to integer values before they are used in simulation. We first round the noninteger release values in the LP solution to integer values. However, due to the run length of 300 shifts, the difference (denoted by Δ) between the sum of the rounded values and the sum of the originally fractional releases can sometimes be significant. As a result, if we use the rounded value for the release policy in the simulation, we may incur unnecessary backorders or extra inventory. To overcome this problem, we first calculate Δ, and then try to evenly distribute this difference over the 300 shifts. For example, suppose $\Delta = 30$ (i.e., the rounded values lead to an accumulated release of 30 more jobs over 300 shifts than the fractional values do). Then we would ideally release one additional job every 10 shifts for an even distribution over 300 shifts. We follow this ideal distribution whenever we can, for example when there is already a scheduled release in a shift and the even distribution causes an additional job release in that shift. However, for shifts with no scheduled releases, this may cause a release of one job by itself. Hence, if the even distribution of Δ suggests releasing a job by itself in a shift with no originally scheduled release, we release the extra job in the next shift with a nonzero release in the original LP solution. This way, we can minimize the discrepancy between the rounded release values and the original ones, and follow the original release schedule as closely as possible.

A similar issue must be resolved while coordinating the planning and the high level schedule. However, we do not address the noninteger issue here, because the output result from the planning model is not a "real" demand profile, but rather a guideline that will allow the high level schedule to meet the actual demand. Thus, the fractional solution from the planning model should be adequate for this purpose.

Therefore, when the planning model provides its decisions, i.e. how many jobs need to be produced at the end of each shift, to the high level schedule as the demand profile, it retains the original fractional values.

Finally, we discuss the criteria used to evaluate the simulation results. A common performance measure in practice is cycle time, which measures how much time a product spends in the system. Average cycle time is a typical aggregated measure of the cycle time performance for each product category. In simulation, we collect the cycle times for finished jobs of each product and divide the average cycle time for each product by its raw processing time to obtain the so-called X-factors. A product-based X-factor in a way measures how many multiples of the raw processing time a product spends in the system, on average, before completion. For the one product experiments, it is obvious that a lower X-factor indicates a better performance. When we compare different approaches or parameters in the two-product experiments, it is possible that one product's X-factor is smaller and the other's is larger across different approaches and models. To facilitate a reasonable overall comparison, we compute the weighted average of the individual X-factors, with the weights being set equal to the product demand rates, thus producing a demand-weighted X-factor as a common measure across both products.

2.5 Experimental Results

Before we present the experimental results, we describe our terminology in this part of the chapter. In the charts that present the simulation results in the following discussion, we use the notation in Fig. 2.1. "P–D" means the traditional planning approach feeding detailed scheduling simulation; "P–H–D" means planning followed by high-level scheduling which feeds detailed scheduling; "H–D" means high-level scheduling feeding detailed scheduling without prior planning. In detailed scheduling, "F" means using FIFO as a dispatching rule, and "T" means using the target following dispatching rule. Aggregate performance measures obtained from simulation are shown as a function of the two arguments – the approach that feeds detailed scheduling (whether it is P–D, P–H–D, or H–D), and the dispatching rule used in detailed scheduling (whether it is F or T). The two major performance measures are the weighted average X-factors and total costs. "X" shows the product 1 X-factor in the one-product setting, and the demand-weighted average X-factor in the two product setting. "C" denotes the total inventory and backorder costs.

To understand how the system performance changes, we vary certain parameters and observe their effects on the relative performances of the tested planning/scheduling approaches. In particular, we vary the unit inventory costs and demand variation across time periods, and examine different scenarios for machine breakdowns. The mini-fab setting explained in Sect. 2.4 is our base case, and in all the following experiments, we only change one parameter at a time and keep other parameters unchanged.

2.5.1 One Product Results

First, we test the approaches with one-product experiments. In these experiments, we only have demand for product 1. In order to keep the traffic intensity (or utilization) consistent with the two-product settings, we change the demand profile in the base case setting for product 1 to 90 jobs every 14 shifts, which results in a traffic intensity of 93%.

In these experiments, we keep the backorder cost fixed at $50 per unit per shift and modify the inventory cost per unit per shift according to Table 2.3.

Figure 2.4 shows the impact of varying the unit inventory cost on X-factor and total costs. We see that the proposed P–H–D approach and the H–D approach always produce lower X-factors than the planning model alone (with either dispatching rule). We also find that the changes in inventory cost affect the overall X-factors level very little. The total costs generally increase as the unit inventory costs increase, regardless of the approach, with small differences across alternative approaches. To see whether there is any interaction between the approaches and the components of the total costs (inventory and backorder), we plot their levels separately in Fig. 2.5. We see that as the unit inventory cost increases, we keep less inventory and incur more backorders for a given approach, as expected. Since the P–H–D and H–D approaches try to control the WIP levels, they have a little more inventory and much fewer backorders than the traditional P–D approach. We also observe that an increase in the unit inventory cost (and a decrease in the relative cost of backorders) causes significantly sharper increases in the conventional P–D approach, regardless of the dispatching rule used for detailed scheduling. The experiments using the FIFO dispatching rule with the P–H–D and H–D approaches also show that FIFO is in general worse than the target following dispatching rule. We only report the performance of P–D and FIFO as a benchmark to make the graphs more legible.

Table 2.3 Change in inventory cost per unit per shift

Experiment index	1	2	3	4
Unit inventory cost for product 1	1	2	4	6

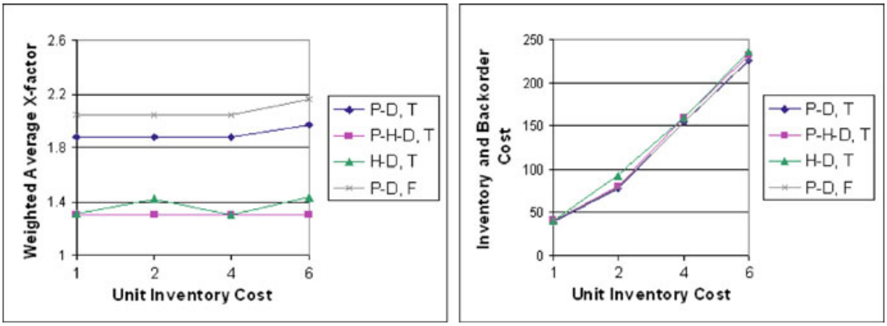

Fig. 2.4 Impact of varying unit inventory costs on X-factor and total costs for single product experiments

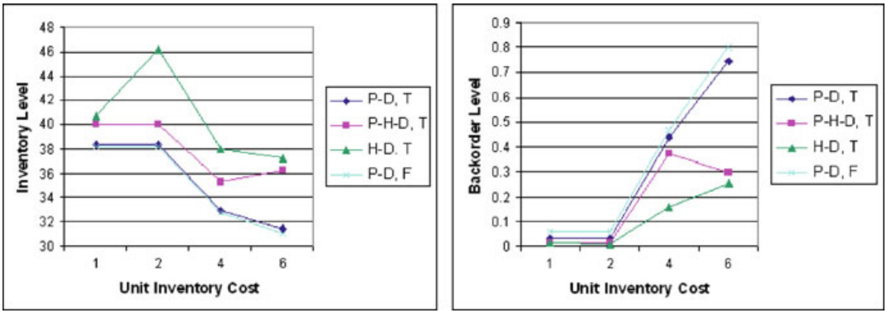

Fig. 2.5 Impact on inventory and backorder quantity of inventory cost change for single product

Table 2.4 Demand distributions for the single product experiments

Experiment index	1	2	3	4
Demand (Uniform)	90	(88, 93)	(85, 95)	(83, 98)

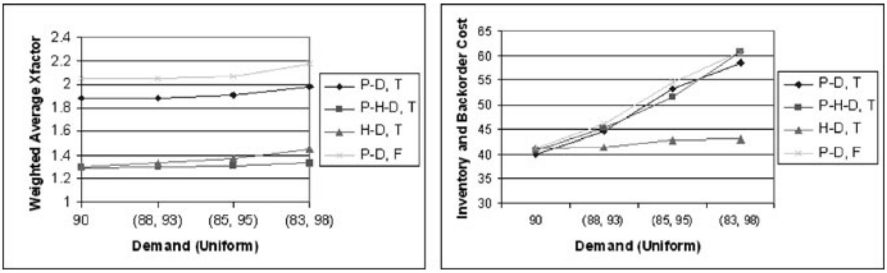

Fig. 2.6 Impact on X-factor and cost of demand variance

2.5.1.1 Demand Variation

In actual manufacturing systems, demand may vary over time. This is especially true in the semiconductor industry, where the technology evolves so fast that it affects the demand levels for different categories of products differently. In the experiments in this section, we modify the variability of the demand. Hence, for each week, we generate a random demand value drawn from a discrete uniform distribution. We change variability of demand over time by extending the underlying range of the distribution as shown in Table 2.4.

Again, as seen in Fig. 2.6, the X-factors obtained by P–H–D and H–D are much lower than those obtained by P–D, regardless of the dispatching rule used with P–D. The H–D approach achieves the lowest total costs among the approaches tested. From these results, we see that increase in demand variation does not cause a significant increase in the X-factor levels for a given planning/scheduling approach, while the increase in total costs can be high, especially for P–D and P–H–D.

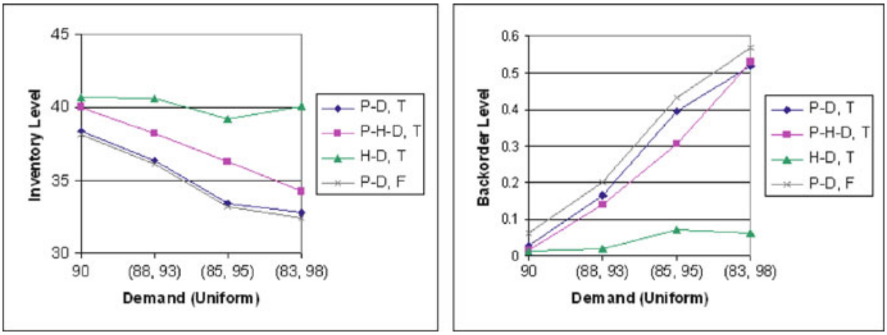

Fig. 2.7 Impact of demand variation on inventory and backorder quantities

Figure 2.7 shows the average inventory and backorder level. We find that the backorder level of the P–D approach increases much more sharply than in the other approaches as the demand variation increases, regardless of the dispatching rule used in conjunction with it. Since the P–H–D approach obtains the implied demand profile from planning and does not directly consider the original demand profile, it cannot perform well in controlling backorders as the demand variation increases. Although the P–H–D approach has higher backorder levels than the H–D approach, it has lower backorders than the P–D approach in three out of four experiments, and a comparable backorder level for the other one. However, the demand variation almost has no impact on the H–D approach; the inventory and backorder levels do not climb significantly while the demand variation increases. The reason could be that the H–D approach takes the original demand (and its variation) into account directly and, to some extent, anticipates the variations in demand. Thus, it performs better than the alternatives in the case where the demand variation is a dominating factor.

2.5.2 Two Product Results

We now present the results of the two-product experiments. Since the two products compete for capacity in this setting, the analysis is not straightforward. However, we still find some useful insights here.

2.5.2.1 Unit Inventory Cost

In this set of experiments, we evaluate the impact of varying unit inventory cost, which takes a value of \$1, \$2, \$4, or \$6 per job per shift. When we change the inventory cost for one product, we fix the inventory cost for the other to the base level, \$1 per job per shift. The changes are displayed in Table 2.5.

Figures 2.8 and 2.9 show the simulation results of these experiments varying unit inventory cost for product 1 and for that of product 2, respectively.

Table 2.5 Inventory cost per unit per shift

Experiment	1	2	3	4	5	6	7
Inventory cost for product 1	1	2	4	6	1	1	1
Inventory cost for product 2	1	1	1	1	2	4	6

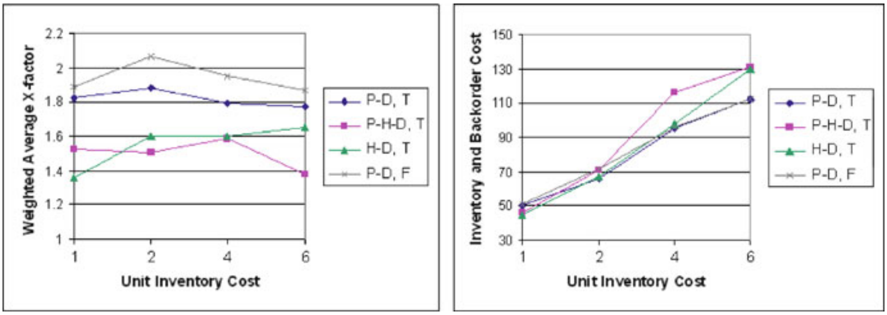

Fig. 2.8 Impact of unit inventory cost of product 1

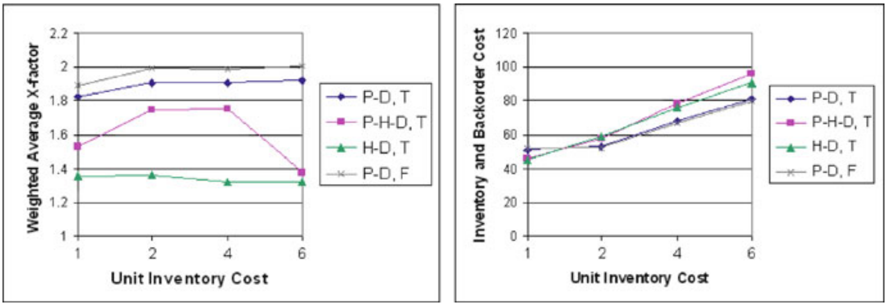

Fig. 2.9 Impact of unit inventory cost of product 2

As before, increasing unit inventory costs results in increasing total costs and relatively stable X-factor values for the tested approach. Comparing the different planning/scheduling approaches, the P–D approach with FIFO has the largest weighted average X-factor, and the P–D approach with target following the second largest. In general, the P–D approach with either dispatching rule has lower total costs, and the approaches with high level scheduling (P–H–D and H–D) have larger total costs. For all the experiments in which we modified the unit inventory cost for product 1, the weighted average X-factors are similar for the P–H–D and H–D approach. Overall, P–H–D and H–D are the best in terms of the X-factor and they are comparable to P–D with FIFO or target following dispatching.

2.5.2.2 Demand Variation

In the previous 2-product experiments, we have demand levels constant over time, at 55 units and 44 units per 14 shifts for products 1 and 2, respectively. To make it more realistic, we change shift demands over time, setting the means to the levels in the constant demand case. The distributions used to create individual shift demand levels are discrete uniform as in the one product experiment. We change the demand variation by extending the range of the uniform distribution, taking range values of 0 (no variation, fixed demand scenario), 5, 10, or 15. For example, when we set the variation range to 5, the demand distribution is discrete uniform between 53 and 58 (both inclusive, represented by Uniform(53, 58)) for product 1, and Uniform (42, 47) for product 2. These levels are described in Table 2.6.

Figures 2.10 and 2.11 show the impact of demand variation of product 1 and 2, respectively. Again, we have lower weighted X-factors in the P–H–D approach and the H–D approach. For the planning model, the cost increases when the demand variation increases for the first product. As in the one product case, it seems that the high level scheduling approach is robust with respect to the demand variation.

Table 2.6 Demand profiles with different distributions

Experiment	1	2	3	4	5	6	7
Product 1 demand (Uniform)	55	(53, 58)	(50, 60)	(45, 61)	55	55	55
Product 2 demand (Uniform)	44	44	44	44	(41, 47)	(38, 50)	(34, 53)

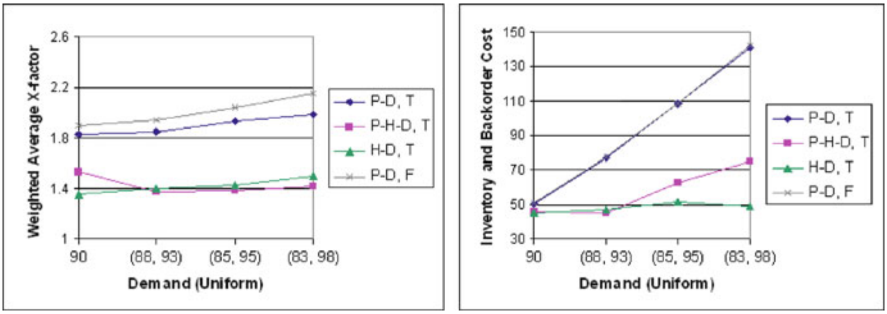

Fig. 2.10 Impact from demand variation of product 1

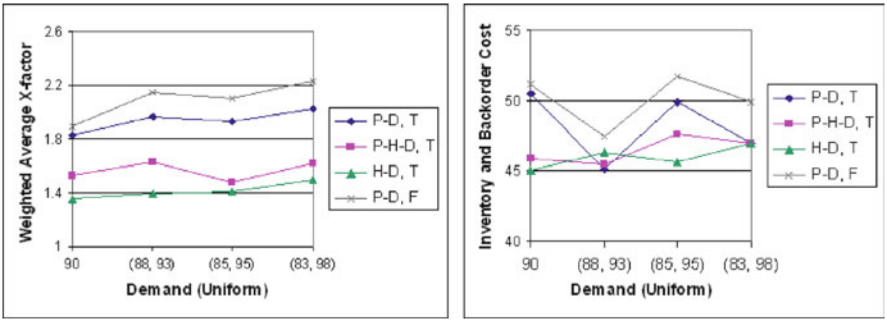

Fig. 2.11 Impact from demand variation of product 2

2.5.2.3 Machine Breakdowns

In the base experimental setting, described in Sect. 2.4.1, the time between machine failures and the time to repair follow exponential distributions. The mean time between failures (MTBF) at station 1 is 42 h, and the mean time to repair (MTTR) is 45 min, which produces an availability of 98.2% (long-term percentage of "up" time out of total time, i.e., MTBF/(MTBF+MTTR)). In this section, we evaluate the impact of varying levels of machine breakdowns. However, to keep the evaluation simple, we keep the traffic intensity of station 1 less than that of station 3, so that the bottleneck station does not change when we vary the downtime parameters. As before, when we modify one of the factors we keep the others at the base levels.

We first vary the mean time between failures at station 1, and keep the mean repair time at 45 min. The MTBF changes according to Table 2.7.

Figure 2.12 shows the impact of different mean times between failures at station 1 on the weighted average X-factors for the tested planning/scheduling approaches. In these experiments, the P–H–D approach produces the lowest weighted X-factor, and its cost is comparable to the cost of the planning model. Although the H–D approach has a slightly higher weighted X-factor than P–H–D (still lower than P–D with FIFO or target following), it has lower total costs than the other three approaches when breakdowns are more frequent. As the mean time between failures increases (i.e., as the availability increases), the differences among the total cost of the various approaches decrease. Also, the weighted X-factor decreases as the disruptions are less frequent for all four approaches. Again, the FIFO dispatching rule with the planning model gives the worst performance.

As part of the experiments in this section, we also vary the MTTR at station 1 according to the values in Table 2.8. By keeping the MTBF at 42 h, we generate availability levels comparable to those in the MTTR experiments. Figure 2.13

Table 2.7 Mean times between failures at station 1

MTBF (h)	7.5	9	12.5	25	42
Station 1 availability	90%	91.6%	94%	97%	98.2%
Station 1 traffic intensity	0.87	0.85	0.83	0.81	0.8

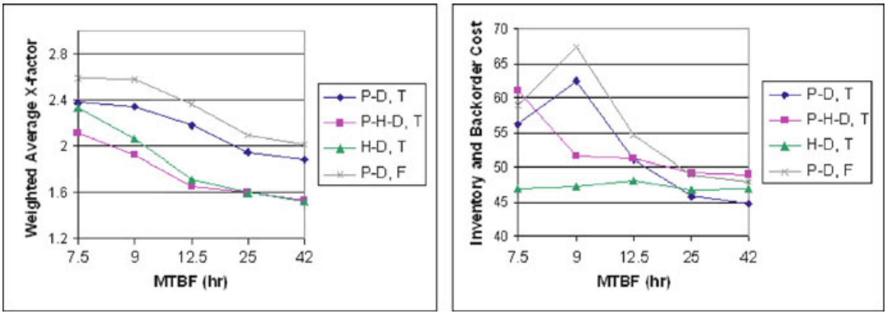

Fig. 2.12 Effect of mean time between failures at station 1

Table 2.8 Mean times to repair at station 1

MTTR(min)	252	210	150	75	45
Station 1 availability	90%	91.6%	94%	97%	98.2%
Station 1 traffic intensity	0.87	0.85	0.83	0.81	0.8

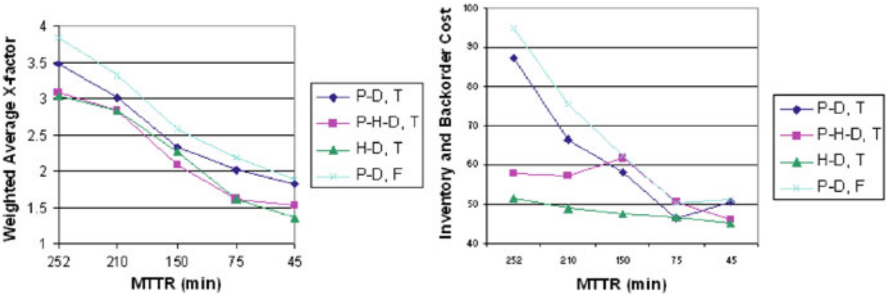

Fig. 2.13 Impact to weighted average X-factor and cost from mean repair time at station 1

shows the effect of different mean repair times on X-factor and total costs. In these experiments, the approaches with high-level scheduling (P–H–D and H–D, both with target following dispatching) produce lower weighted X-factors than P–D with FIFO and target following. Again, P–H–D and H–D have similar weighted X-factors. In total costs, these approaches yield lower values than the planning-only-based approaches (P–D), especially when the repair times are long and availability is low. Comparing this with the previous set of experiments, we find that the impact from the breakdown duration (MTTR experiments) is more pronounced than the impact from the frequency of breakdowns (MTBF experiments).

2.5.3 Gantt Chart Analysis

To analyze why P–H–D or H–D works better than P–D, we examine the Gantt charts representing the processing of jobs in some of the above experiments. As mentioned before, we choose the same random seed for all three approaches so that they see the same realization of machine breakdowns. Figures 2.14 and 2.15 are two typical time periods in one replication of the same setting, which corresponds to the second column in Table 2.7.

As we can see from Fig. 2.14, in the P–D approach, during the 1.05–1.08 ($\times 10^4$) min time interval, there are no new releases. Therefore, when the machine breaks down at station 1 around 1.12×10^4 min, there is not enough WIP at the bottleneck station 3, and the station becomes idle at around 1.13×10^4 min. However, for the P–H–D and H–D approach, the same disruption does not affect the bottleneck station at all.

In Fig. 2.15, although the machine break down in station 1 affects the bottleneck station in all three approaches, we can see clearly that the idle time of the bottleneck station in the P–H–D and H–D approaches is significantly smaller than the one

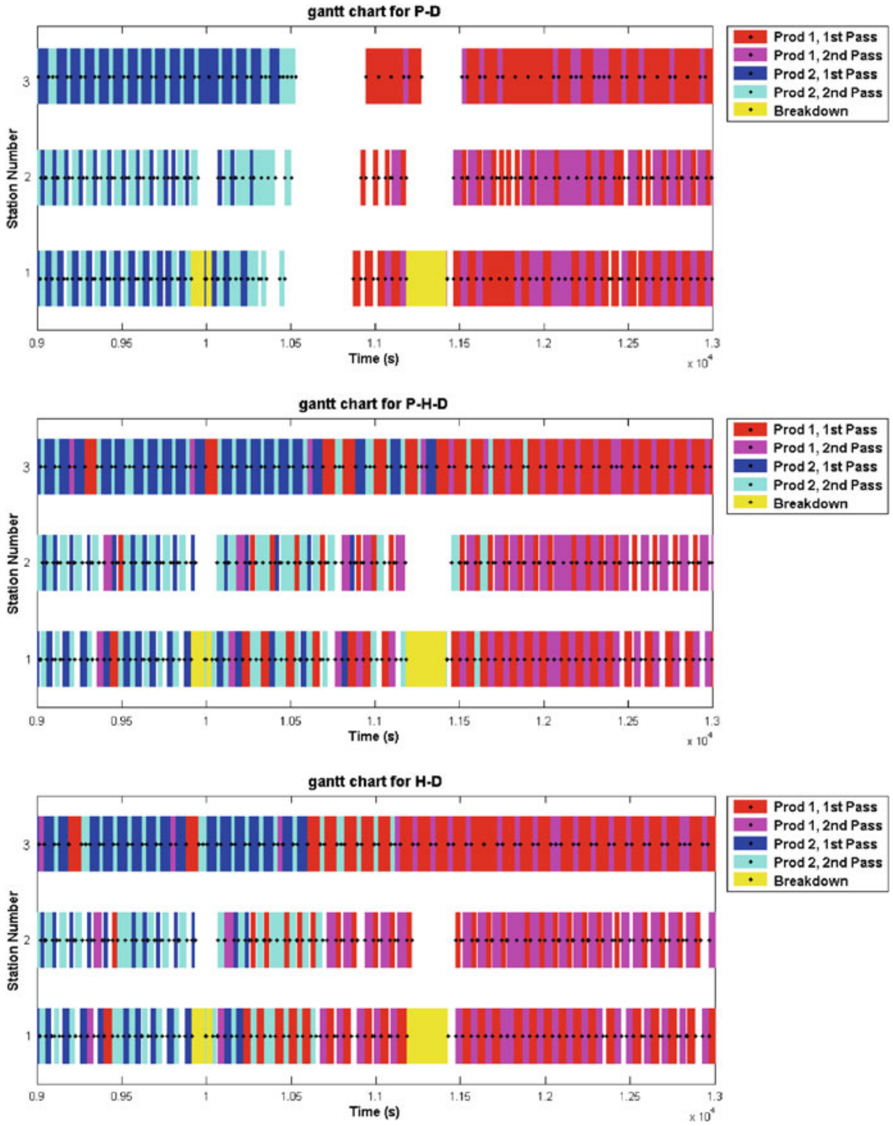

Fig. 2.14 Gantt chart for three approaches between 0.9 and 1.3 ($\times 10^4$) min

in the P–D approach. If we look at the Gantt chart along the whole time horizon, we find that situations such as those seen in Figs. 2.14 and 2.15 are very common. In the P–H–D and H–D approaches, jobs are "pushed" to the bottleneck station, thus some jobs can avoid the machine-break-down situation without being held at station 1. On the contrary, the total number of jobs released is almost the same for all three approaches since the demand profiles are the same. Therefore, the weighted average cycle times in the P–H–D and H–D approach are smaller than the cycle time achieved by the P–D approach.

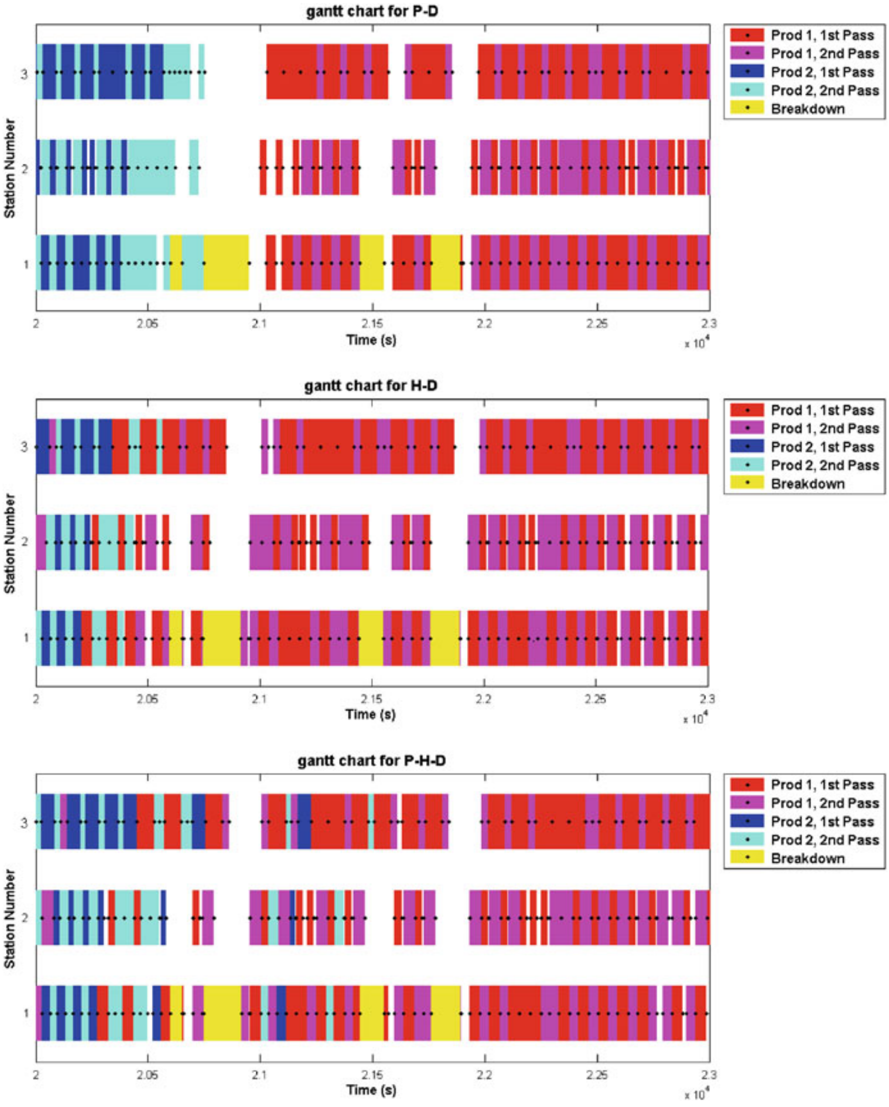

Fig. 2.15 Gantt chart for three approaches between 2 and 2.3 (×10^4)min

2.6 Summary

Production planning is a critical process for every manufacturing company since it directly affects the performance of detailed scheduling, which ultimately determines the overall performance of the manufacturing system. Despite this interdependency, in many manufacturing companies planning and detailed scheduling activities are separated, with a limited coordination between them. Usually, planning decisions obtained from a long-term model are fed into a scheduling model as restrictions, jobs to be released, and due dates. Detailed scheduling is typically handled through dispatching.

In this paper, we suggest incorporating an intermediate stage into the usual planning-scheduling hierarchy to seek coordination between planning and detailed scheduling. This approach consists of the usual planning model and a high level scheduling model, both of which feed dispatching-based detailed scheduling. The high level scheduling model explicitly controls the WIP over time at each stage in the system, thus providing a more specific guide to detailed scheduling. Our numerical results indicate that the proposed approach results in shorter cycle times (realized as a lower weighted X-factor) than the conventional two-stage approach of feeding planning results into a detailed scheduling algorithm. In most cases, the use of the high level scheduling model, either as an intermediate step between planning and detailed scheduling or as an initial step before detailed scheduling, results in lower inventory and backorder costs. This approach without a major planning step turns out to be especially suitable for situations with high demand variability. All these results indicate that if we consider more scheduling details in the planning level and/or at an intermediate level before detailed scheduling, we would have better performance on the shop-floor in terms of both cycle times and system costs including inventory and backorders.

In actual manufacturing systems, all planning and scheduling systems are implemented in a rolling horizon fashion. This is also true for the three-level planning/high-level scheduling/detailed scheduling approach. For example, the planning model would generate a plan for one quarter every month, and produce the first several months' demand profile, release schedule, and production targets, for the upcoming weeks. This information would then be released to the high level scheduling model. Then the high level scheduling model would generate a more granular (say by day or shift) release policy and processing targets, for each major processing step pertaining to the upcoming week or month. The detailed scheduling step would try to implement the high-level scheduling decisions made for the next few days on the shop floor. High level scheduling and planning steps can be rerun with some regular frequency (say every week and month, respectively). During the execution of detailed scheduling, the current system status, such as WIP level, machine availability, may provide feedback that would initiate more frequent runs of the high level scheduling and planning models. We believe the rolling horizon approach would improve the performance of the proposed approach that incorporates a high-level scheduling model. Simulation of such a rolling horizon approach requires significant effort and we leave this as a topic for future research.

References

Asmundsson J, Rardin R, Uzsoy R (2006) Tractable nonlinear production planning models for semiconductor wafer fabrication facilities. IEEE Trans Semicond Manuf 19:95–111

Bertsimas D, Gamarnik D, Sethuraman J (2003) From fluid relaxation to practical algorithms for job shop scheduling: the holding cost objective. Oper Res 51(5):798–813

Dai J, Weiss G (2002) A fluid heuristic for minimizing makespan in job shops. Oper Res 50(4):692–707

Graves S (1986) A tactical planning model for a job shop. Oper Res 34:552–533

Hackman S, Leachman R (1989) A general framework for modeling production. Manag Sci 35:478–495

Horiguchi K, Raghavan N, Uzsoy R, Venkateswaran S (2001) Finite-capacity production planning algorithms for a semiconductor wafer fabrication facility. Int J Prod Res 39:825–842

Hung YF and Leachman RC (1996) A production planning methodology for semiconductor manufacturing based on iterative simulation and linear programming calculations. IEEE Trans Semicond Manuf 9(2):257–269

Jaikumar R (1974) An operational optimization procedure for production scheduling. Comput Oper Res 1:191–200

Kim S, Yea S, Kim B (2003) Shift scheduling for steppers in the semiconductor wafer fabrication process. IIE Trans 34:167–177

Kleindorfer PR, Kriebel CH, Thompson GL, Kleindorfer GB (1975) Discrete optimal control of production plains. Manag Sci 22

Leachman R, Kang J, Lin V (2002) SLIM: Short cycle time and low inventory in manufacturing at samsung electronics. Interfaces 32(1):61–77

Lee Y, Kim T (2002) Manufacturing cycle time reduction using balance control in the semiconductor fabrication line. Prod Plann Contr 13:529–540

Missbauer H (2002) Aggregate order release planning for time-varying demand. Int J Prod Res 40(3):699–718

Pahl J, Voβ S, Woodruff D (2005) Production planning with load dependent lead times. Q J Oper Res 3:257–302

Qin SJ and Badgwell TA (2003) A survey of industrial model predictive control technology. Contr Eng Practice 11:733–764

Rose O (2002) Some issues of the critical ratio dispatch rule in semiconductor manufacturing. Proceedings of the 2002 Winter Simulation Conference, December 2002

Tsakalis K, Godoy JF, Rodriguez A (2003) Hierarchical modeling and control for re-entrant semiconductor fabrication lines: A mini-fab benchmark. pp. 578–587

Vargas-Villami F, Rivera D (2000) Multilayer optimization and scheduling using model predictive control: application to reentrant semiconductor manufacturing lines. Comput Chem Eng 24:2009–2021

Vargas-Villami F, Rivera D, Kempf K (2003) A hierarchical approach to production control of reentrant semiconductor manufacturing lines. IEEE Trans Contr Syst Technol 11(4):578–587