Chapter 14
The Ongoing Challenge: Creating an Enterprise-Wide Detailed Supply Chain Plan for Semiconductor and Package Operations

Kenneth Fordyce, Chi-Tai Wang, Chih-Hui Chang, Alfred Degbotse, Brian Denton, Peter Lyon, R. John Milne, Robert Orzell, Robert Rice, and Jim Waite

14.1 Introduction

In the mid-1980s, Karl Kempf of Intel and Gary Sullivan of IBM independently proposed that planning, scheduling, and dispatch decisions across an enterprise's demand-supply network were best viewed as a series of information flows and decision points organized in a hierarchy or set of decision tiers (Sullivan 1990). This remains the most powerful method to view supply chains in enterprises with complex activities. Recently, Kempf (2004) eloquently rephrased this approach in today's supply chain terminology, and Sullivan (2005) added a second dimension based on supply chain activities to create a grid (Fig. 14.1) to classify decision support in demand-supply networks. The row dimension is decision tier and the column dimension is responsible unit. The area called global or enterprise-wide central planning falls within this grid.

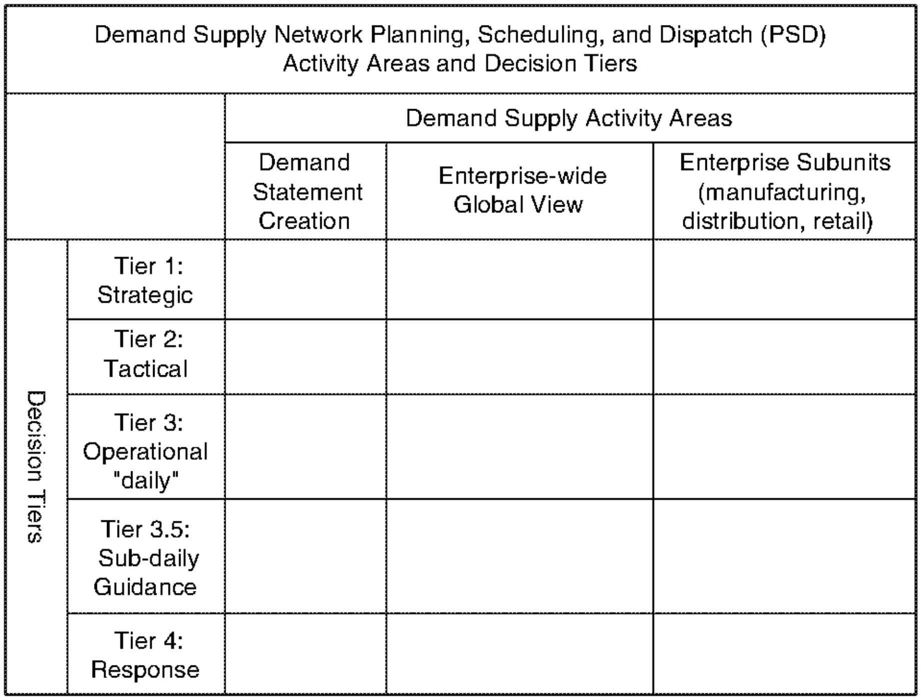
14.1.1 Decision Tiers

Demand-supply network (or supply chain) decisions in the semiconductor industry typically fall into one of four decision tiers (row dimension): strategic, tactical, operational, and response (dispatch). The categories are based on the planning horizon, the apparent width of the opportunity window, and the level of precision required in supporting the information.

The first decision tier, *strategic scheduling*, is typically driven by the lead time required for business planning, resource acquisition, and new product introduction. This tier can often be viewed in two parts: *very long-term* and *long-term*. Here, decision makers are concerned with a set of problems that are 3 months to 7 years into the future. Issues considered include, but are not limited to, what markets the firm

K. Fordyce (✉)
IBM Corporation Strategic Systems Department, 227 Evergreen Lane, Hurley, NY 12443, USA
e-mail: fordyce@us.ibm.com

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_14, © Springer Science+Business Media, LLC 2011



Demand Supply Network Planning, Scheduling, and Dispatch (PSD) Activity Areas and Decision Tiers				
		Demand Supply Activity Areas		
		Demand Statement Creation	Enterprise-wide Global View	Enterprise Subunits (manufacturing, distribution, retail)
Decision Tiers	Tier 1: Strategic			
	Tier 2: Tactical			
	Tier 3: Operational "daily"			
	Tier 3.5: Sub-daily Guidance			
	Tier 4: Response			

Fig. 14.1 Grid representation of enterprise-wide supply chain planning

will be in, general availability of tooling and workers, major process changes, risk assessment of changes in demand for existing products, required or expected incremental improvements in the production process, lead times for additional tooling, manpower and planning.

The second tier, *tactical scheduling*, deals with problems the company faces in the next week to 6 months. Estimates are made of yields, cycle times (CTs), and binning percentages. Permissible material substitutions are identified. Decisions are made about scheduling starts or releases into the manufacturing line (committing available capacity to new starts). Delivery dates are estimated for firm orders, available "outs" by time buckets are estimated for bulk products, and daily going rates (DGRs) for schedule-driven products are set. The order/release plan is generated or regenerated, and (customer-requested) reschedules are negotiated.

The third tier, *operational scheduling*, deals with the execution and achievement of a weekly plan. Shipments are made, serviceability levels are measured, and recovery actions are taken. Optimal capacity consumption and product output are computed.

The fourth tier, *real-time response system*, addresses the problems of the next hour to a few weeks by responding to conditions as they emerge in real time. It also accommodates variances from availability assumed in the plan creation and commitment phases. Within the demand-supply network, real-time response is often found in two predominant areas: manufacturing dispatch (which assigns lots to tools) and order commitment (available to promise, or ATP). In manufacturing dispatch scheduling (DS), decisions concern monitoring and controlling of the actual manufacturing flow and instructing the operator what to do next to achieve current manufacturing goals. The goal of most ATP applications is to provide a commit date to a customer order as quickly as possible. Although it may not respond in real time, its goal is to modify the current match between assets and demand to provide a real-time commit to an order placed by a customer.

Within semiconductor manufacturing, the decisions made across the tiers are typically handled by groups with one of the three responsibilities: establishing product demand, maintaining an enterprise-wide global view of the demand-supply network, and ensuring that subunits (such as manufacturing location, vendor, warehouse) are operating efficiently. Although ideally all planning would be central, in practice the level of complexity precludes this. Capacity or tool planning is a good example. At the enterprise level, capacity is modeled at some level of aggregation, typically viewing a tool set as a single capacity point. At the factory level, each tool, or potentially each chamber in a tool, is modeled.

14.1.2 Basics of Enterprise-Wide End-to-End Central Planning

The activity of concern in this chapter, called "enterprise-wide end-to-end central planning," falls into the second column in Fig. 14.1 (enterprise-wide global view) and straddles all the rows: strategic, tactical, and operational. This planning activity, which is a key requirement for successfully managing operations in any manufacturing industry, involves the coordination of supply and demand (current and future). In large-scale manufacturing systems such as semiconductor, this planning activity must handle the dual challenges of scope (complexity) and scale (size). In its simplest form, supply chain planning combines the below three "basic pillars" (Fig. 14.2):

- Business processes and organizational structure
- Data collection and storage mechanisms
- Analytical or modeling methods to execute the following steps

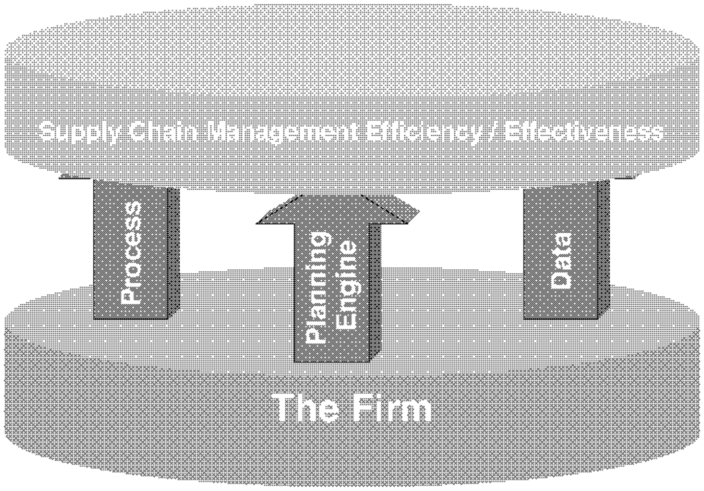


Fig. 14.2 Three basic pillars for supply chain planning

1. Create a demand statement
2. Gather and project assets to a decision point
3. Create an enterprise-wide end-to-end central plan by matching current and future assets with current and future demand (Fig. 14.3)

   3.1 To generate

   3.1.1 A projected supply linked with exit demand

   3.1.1.1 Including projecting supply without demand, capacity utilization, and pegging;

   3.1.2 Synchronization signals across the enterprise

   3.1.2.1 Including starts (or manufacturing releases), target outs, due dates, ship plans, stocking levels, lot priorities, planned substitutions, capacity utilization, etc.

   3.2 Typically, this is an iterative process

   3.2.1 That consists of a set of model runs under different settings, such as

   3.2.1.1 With and without capacity.

   3.2.1.2 With and without new projected supply,

   3.2.1.3 With and without new forecasted demand,

   3.2.1.4 Where different runs occur at different times during the week.
4. Execute the plan, that is,

   4.1 Send signals to each core enterprise organization (such as manufacturing, storage, vendors, etc.)

   4.1.1 Converting this signal into more detailed guidelines for each organization,

   4.1.2 Executing the detailed manufacturing or transport activities.

   4.2 Send projected supply to ATP which

   4.2.1 Handles incoming requests for product,

   4.2.2 Makes tradeoffs or reallocation as needed and defined by the business rules.
5. Repeat

This flow is summarized in Fig. 14.4.

Step 3 is often referred to as the enterprise-wide best-can-do (BCD) matching, or the central planning engine (CPE). The core task of the CPE is to deploy modeling methods to match assets with demand across an enterprise to create a projected supply linked with demand and synchronization signals. Assets include, but are not limited to, manufacturing starts (or releases), work in progress (WIP), inventory, purchases, and capacity (manufacturing equipment and manpower). Demands include, but are not limited to, firm orders, forecasted orders, and inventory buffer. The CPE has four core components, which

1. Represent the (potential) material flows in production, business policies, constraints, demand priorities, current locations of asset, etc., and relate all this information to exit demand.
2. Capture asset quantities and parameters (CTs, yields, binning percentages, etc.).

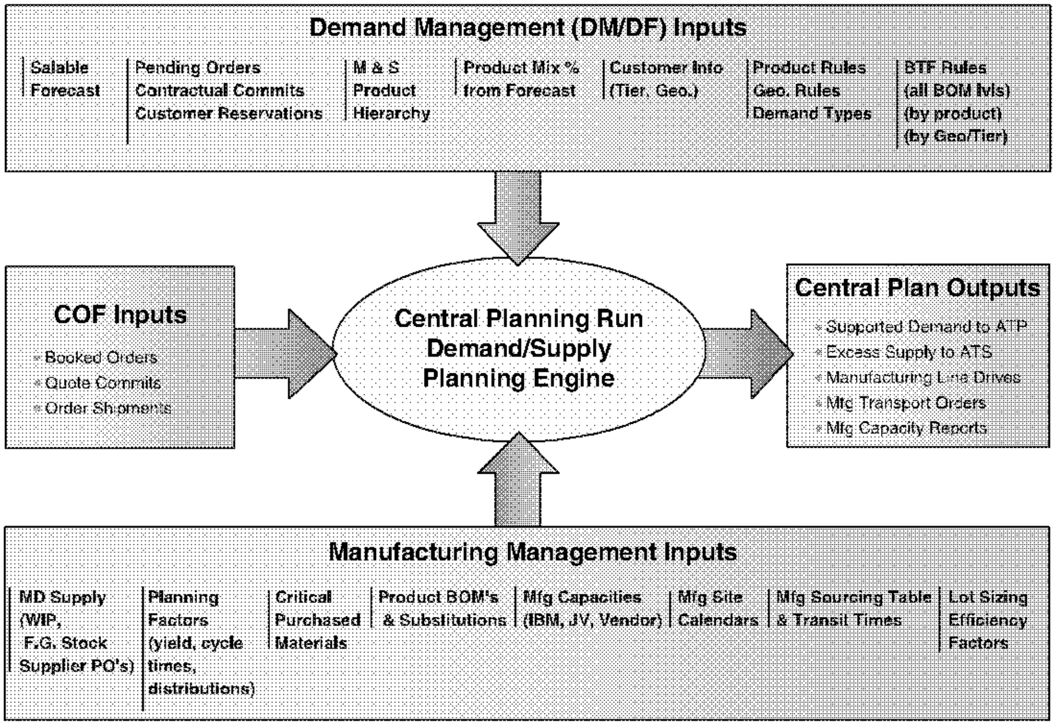

Demand Management (DM/DF) Inputs
Salable Forecast
Pending Orders
Contractual Commits
Customer Reservations
M & S Product Hierarchy
Product Mix % from Forecast
Customer Info (Tier, Geo.)
Product Rules
Geo. Rules
Demand Types
BTF Rules (all BOM lvls) (by product) (by Geo/Tier)
COF Inputs
• Booked Orders
• Quote Commits
• Order Shipments
Central Planning Run Demand/Supply Planning Engine
Central Plan Outputs
• Supported Demand to ATP
• Excess Supply to ATS
• Manufacturing Line Drives
• Mfg Transport Orders
• Mfg Capacity Reports
Manufacturing Management Inputs
MD Supply (WIP, F.G. Stock Supplier PO's)
Planning Factors (yield, cycle times, distributions)
Critical Purchased Materials
Product BOM's & Substitutions
Mfg Capacities (IBM, JV, Vendor)
Mfg Site Calendars
Mfg Sourcing Table & Transit Times
Lot Sizing Efficiency Factors

Fig. 14.3 Typical data inputs and outputs for enterprise-wide central planning engine

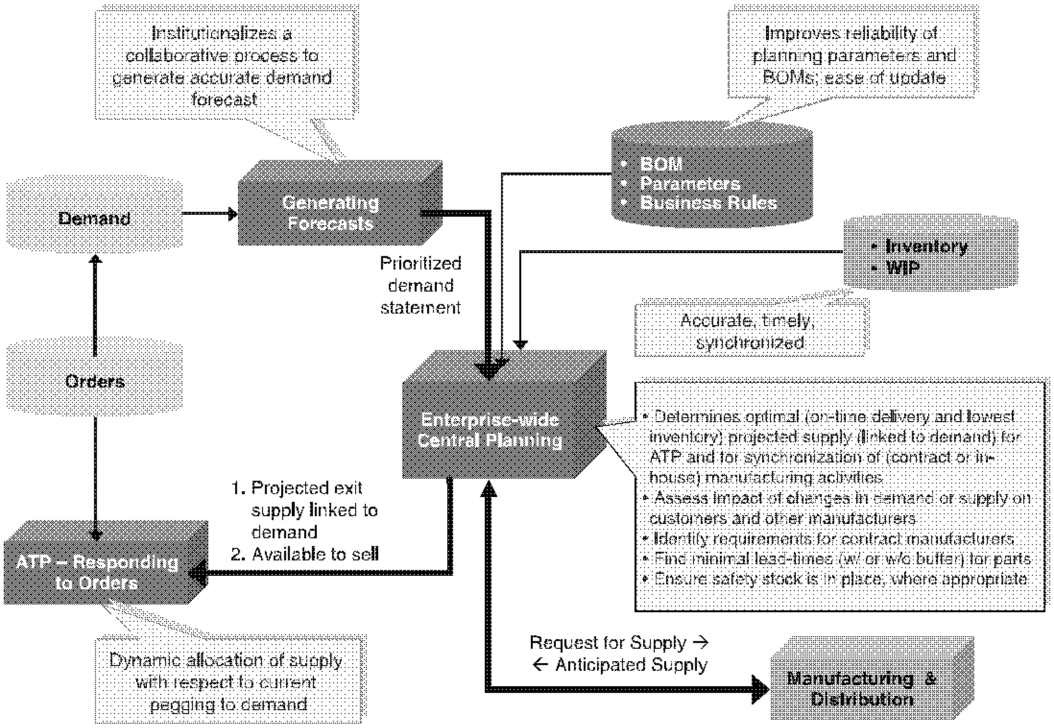

Institutionalizes a collaborative process to generate accurate demand forecast
Improves reliability of planning parameters and BOMs; ease of update
• BOM
• Parameters
• Business Rules
Demand
Generating Forecasts
• Inventory
• WIP
Prioritized demand statement
Accurate, timely, synchronized
Orders
Enterprise-wide Central Planning
• Determines optimal (on-time delivery and lowest inventory) projected supply (linked to demand) for ATP and for synchronization of (contract or in-house) manufacturing activities
• Assess impact of changes in demand or supply on customers and other manufacturers
• Identify requirements for contract manufacturers
• Find minimal lead-times (w/ or w/o buffer) for parts
• Ensure safety stock is in place, where appropriate
1. Projected exit supply linked to demand
2. Available to sell
ATP – Responding to Orders
Request for Supply →
← Anticipated Supply
Manufacturing & Distribution
Dynamic allocation of supply with respect to current pegging to demand

Fig. 14.4 Information and decision flows in a successful demand-supply network

3. Search and generate a supply chain plan, relate the outcome to demand, and modify the plan to improve the match.
4. Display and explain the results.

This is the focus of this chapter.

Historically, the concept that an enterprise needs a reasonably tightly coupled central planning process to be successful in the market took hold in the middle 1990s (Shobrys 2003). Within the semiconductor industry, the work at Harris Semiconductor (Leachman et al. 1996) and Digital Equipment Corporation (Arntzen et al. 1995) certainly elevated awareness. The work by Tayur et al. (1998) noted the growing importance of quantitative models, and the work by Lee et al. (1997), Swaminathan and Smith (1998), and Lin et al. (2000) made it clear that real improvements in organizational performance were possible from centralized planning. To be clear, we are not saying the concept did not exist before and certain businesses had some success at centralized control. Glover et al. (1979) produced, at a minimum, a clear ancestor to today's supply chain planning models (including using different models depending on whether strategic or tactical planning was being done). The ten ways that a material requirements planning (MRP) can fail (Woolsey 1979) still ring true today. Fogarty and Hoffman (1983) identified some core requirements. Duchessi (1987) identified the importance of a feasible production plan and the applicability of knowledge-based techniques to this problem. The papers by Hackman and Leachman (1989), Uzsoy et al. (1992, 1994), and Graves et al. (1995) clearly identified the emerging need and interest. Norden (1993) articulated the general trend in applying quantitative methods from well-structured operations to the more speculative aspects of strategy and policy formation. In some respects, little is new after Orlicky (1975). It is clear that the mid-1990s saw the "launch" of the supply chain management (SCM) industry, consisting of vendors, consultants, analysts, watch groups, specialized reports, and numerous internal efforts. The pace remains full speed ahead in 2007; see, for example, the supply chain operations reference (SCOR) model (www.supply-chain.org).

Much of the work in SCM from the mid-1990s until 2004 or 2005 focused on the creation of a centralized planning process and a centralized data view of the status of the organization's manufacturing and distribution activities. Less emphasis was placed on the third pillar of the SCM triangle (Fig. 14.2) – the decision model or the analytics – the enterprise-wide detailed level CPE. Certainly, basic matching engines were produced and used, but most relied on heuristics despite the promise shown by linear programming (LP) in the Harris work (Leachman et al. 1996) to handle the complexity of demand-supply networks in semiconductor manufacturing.

It is true that if you do not have your processes and data in place, the ability of your "CPE" to handle business complexity does not matter. However in 2007, just having your data and processes in place is no longer enough. Firms now compete on analytics (Davenport 2006) and many of those complexities, such as alternative bill of materials (BOMs) or customer request and commit dates, are core competencies for firms and cannot be "leaned" away without damaging the firm's competitive position. Kempf (2004) observed: "the tradition of referring to this supply-demand network as simply supply-chain grossly understates the actual complexity." The ability to handle these complexities resulted in a more effective supply chain plan.

14.1.3 IBM Microelectronic Division's SCM Transformation

Between 1992 and 1999 (Promoting 2005; Fordyce 1998, 2001), the microelectronics industry went through a dual transformation in core technology and use (or market). On the technology side, chip size, speed, and versatility took quantum leaps. For example, IBM pioneered copper circuits, RISC-based CPU processors, silicon germanium and silicon-on-insulator technologies, and innovative insulation techniques for copper circuits. The market for microelectronic devices expanded from an initial base in computers to a wide range of products, such as cell phones, car security systems, advanced GPS-based trackers, greeting cards, and aids for the physically challenged. Microelectronic devices truly pervade the world. This dual expansion has transformed manufacturing from making large quantities of just a few parts to varying quantities of many different parts. To quantify this change, in the mid-1980s IBM Microelectronics had about 100 active part numbers (PNs) with demand; the current number is about 6,000.

Like other industries, microelectronics is also under tremendous pressure to be more responsive to customers. Today's IBM Microelectronics Division – a tightly coupled set of manufacturing facilities employing a centralized SCM process, which subweekly determines which orders can be met when for the entire Division and can respond rapidly to customer requests to place or change orders as well as manage what-if scenarios – was only a dream in 1994.

In 1994, IBM manufacturing facilities located worldwide to produce wafers, devices, modules, and cards were semi-independent of one another at best. They supplied the parts for downstream manufacturing facilities for mainframes, workstations, printers, networks, and storage equipment. Their geographic linkages were strong. Typically, a facility in Europe would provide component parts for an IBM box plant in Europe. Supply chains consisted of individual manufacturing facilities linked directly with the box plants they supported with no concept of centralized control. Each manufacturing facility produced far fewer products and therefore managed far fewer products than it does today.

During the early 1990s, IBM Microelectronics Division needed to compete in the business of supplying components to original equipment manufacturers (OEM), while continuing to provide the IBM box facilities with state-of-the-art (and often custom) components, reducing costs and improving customer (internal and external) satisfaction. To accomplish these goals, IBM Microelectronics had to transform from a loose confederation of manufacturing facilities into a unified division. A major reengineering effort was launched in 1994 to completely restructure the Division's SCM applications (Lyon et al. 2001), covering processes, data, and planning models. Note that sometimes SCM is referred to as "customer order fulfillment" (COF).

14.1.4 IBM's Decision to Invest in a "Smart" Planning Engine & Supply Chain Efficiency Frontier

After an extensive review in 1994 (Sullivan 2007), the IBM Corporation made a clear business decision in 1995 as part of its reengineering effort (Lyon et al. 2001; Sullivan 2005) that a "simple" planning engine would not meet the long-term needs of its microelectronics business. IBM chose to invest to build a "smart" CPE as it was a key to being at the frontier of "supply chain efficiency."

In macroeconomics and financial engineering, a well-established concept is the "efficiency frontier" (also called production possibility frontier). In macroeconomics, if an economy is operating below its efficiency frontier, an increase in one product as well as in other goods can be obtained by increasing overall economic efficiency. If an economy is operating at its efficiency frontier, any increase in one product requires reducing the quantity of all other goods produced in the same period. Obviously in practice, there are multiple dimensions to this curve which can be education, transportation, or medicine. In finance, a similar pattern exists between risk and return: more risk, higher return; less risk, less return.

A similar concept can be applied in supply chains. As in macroeconomics, the efficiency frontier has multiple dimensions, consisting of inventory, on-time delivery (OTD), customer satisfaction, profit, revenue, stability, growth of market, etc. For illustrative purposes, we will have two dimensions (Fig. 14.5): (a) improvements in inventory performance (e.g., turns, excess, shelf life), and (b) improvements in OTD (e.g., meeting request date, finding acceptable commit date, meeting commit date).

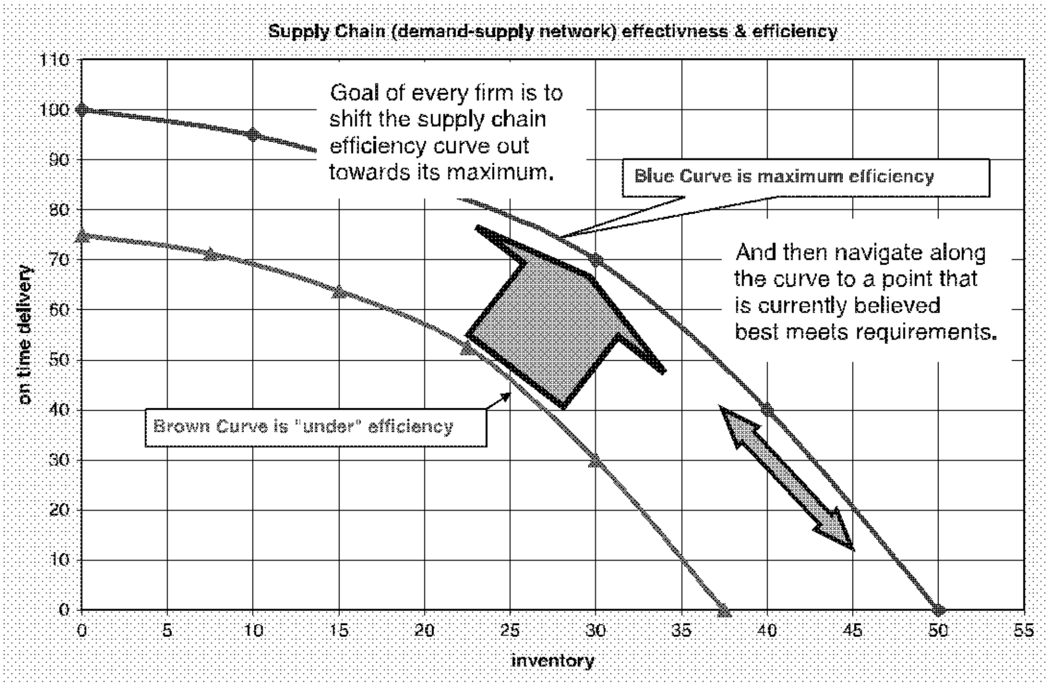
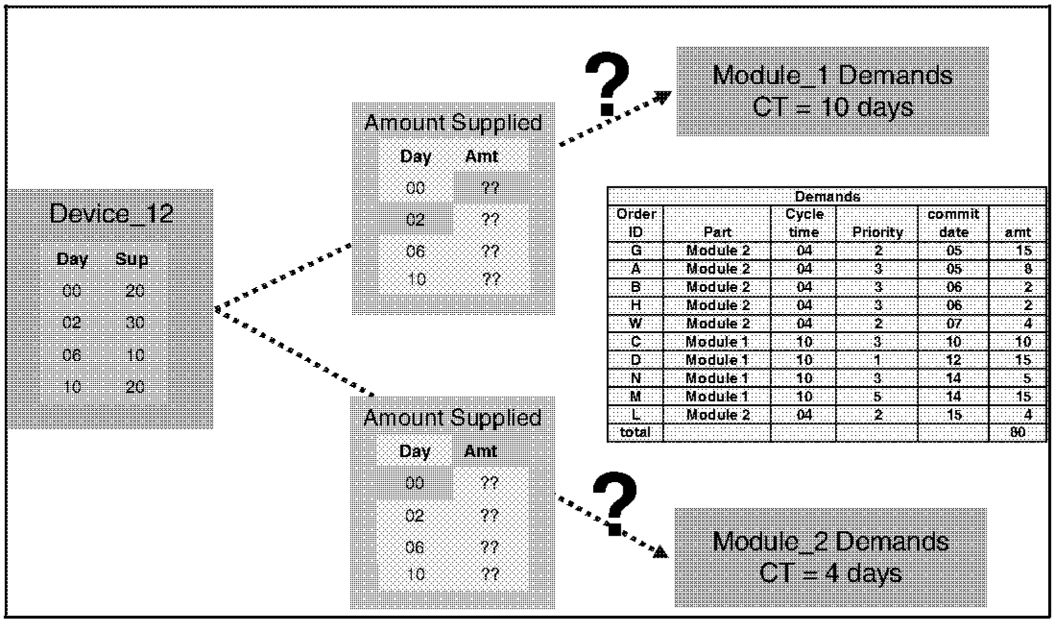


Fig. 14.5 Two-dimensional efficiency frontier: Inventory and on-time delivery



Device_12

Day	Sup
00	20
02	30
06	10
10	20

Amount Supplied

Day	Amt
00	??
02	??
06	??
10	??

Amount Supplied

Day	Amt
00	??
02	??
06	??
10	??

Demands					
Order ID	Part	Cycle time	Priority	commit date	amt
G	Module 2	04	2	05	15
A	Module 2	04	3	05	8
B	Module 2	04	3	06	2
H	Module 2	04	3	06	2
W	Module 2	04	2	07	4
C	Module 1	10	3	10	10
D	Module 1	10	1	12	15
N	Module 1	10	3	14	5
M	Module 1	10	5	14	15
L	Module 2	04	2	15	4
total					80

Fig. 14.6 Simple example of allocating supply to meet demand

Just as in macroeconomics, if a supply chain is at its efficiency frontier, improving inventory requires reducing OTD. As Shobrys and Fraser (2003) make clear, few supply chains are operating at their efficiency frontier. Therefore, the goal of each firm is to first shift the curve outward towards the efficiency frontier; then, the firm will navigate along the curve with great precision to the point that it believes best meets current market requirements.

A smarter mathematical model (planning engine) can directly improve a firm's performance. Let us look at the following simple example (Fig. 14.6).

Assume a firm makes two products: Module_1 and Module_2, and both modules require Device_12 as their only component part. The time it takes to produce Module_1 and Module_2 is 10 and 4 days, respectively. We will assume that once a unit of Device_12 is allocated to either Module_1 or Module_2, work begins immediately to make the module. If 3 units of Device_12 are allocated to Module_1 on day 2, then 3 units of Module_1 are available to meet demand on day $12(= 2 + 10)$. Similarly, if 4 units of Device_12 are allocated to Module_2 on day 6, then 4 units of Module_2 are available to meet demand on day $10(= 4+6)$. The anticipated supply of Device_12 is as follows:

Anticipated supply device 12		
Day	Supply	Cum sup
00	20	20
02	30	50
06	10	60
10	20	80

And the anticipated demand is as follows:

Demands					
Order ID	Part	Cycle time	Priority	commit date	amt
G	Module_2	04	2	05	15
A	Module_2	04	3	05	8
B	Module_2	04	3	06	2
H	Module_2	04	3	06	2
W	Module_2	04	2	07	4
C	Module_1	10	3	10	10
D	Module_1	10	1	12	15
N	Module_1	10	3	14	5
M	Module_1	10	5	14	15
L	Module_2	04	2	15	4
total					80

The primary task of the CPE in this simple scenario is to allocate the anticipated supply of Device_12 to produce Module_1 and Module_2 so as to best meet demand and minimize inventory. Below, two different search mechanisms come up with different allocations.

Table 14.1 is interpreted as follows. The first set of columns under "Demands" repeat the demand information. It is the order identifier, the type of part, the cycle time to produce the part, the priority of the demand, the commit date to meet this demand and the amount of the demand. The second set of columns under "Method 1" is the allocation of supply of device_12 to produce either Module_1 or Module_2 to meet demand. The row for demand D for Module_1 can be read as follows: 15 units from the supply of Device_12 available on day 00 are allocated to meet this demand. Fifteen units of Module_1 comes to stock (is completed) on day 10 and is allocated to demand D. Since the commit date for demand D is day 12 and the supply is available on day 10, the supply is 2 days earlier (this is the delta column). The OTD score is 0. The algorithm for scoring is: (a) if the demand is met on time or early, the OTD score is 0; (b) if the demand is late the smaller of the number of days late and −5 is divided by the demand priority (this caps the "days late" at 5 and weighs it inversely to the demand priority). The OTD scoring mechanism is designed only to be illustrative.

Observe that method 2 improves both OTD (−4.67 compared to −6.50) and inventory (39 compared to 78) in comparison with method 1, moving the firm closer its efficiency frontier. *The "smarter" planning engine makes a direct impact on the firm's performance.*

Below illustrates a third allocation where OTD is improved (−4.33 compared to −4.67) but inventory is increased (43 compared to 39). This is an example of moving along the curve of efficiency frontier. The smart engine is able to follow the firm's business rules and position the firm at the point on the efficiency frontier, where it currently believes is optimal. The planning engine institutionalizes the corporate rules and insures that they are known and followed (Table 14.2).

Table 14.1 Different allocations resulted from different search mechanisms

Comparing Two different Methods to Allocate Device_12 to the production of Module_1 and Module_2																	
Demands						Method 1						Method 2					
ID	Part	Cycle time	Priority	commit date	amt	alloc date	all amt	stk date	stk amt	delta	otd score	alloc date	all amt	stk date	stk amt	delta	otd score
D	Module 1	10	1	12	15	00	15	10	15	02	0.00	**02**	15	12	15	00	0.00
G	Module 2	04	2	05	15	02	15	06	15	–01	–0.50	**00**	15	04	15	01	0.00
W	Module 2	04	2	07	4	00	4	04	4	03	0.00	**02**	4	06	4	01	0.00
L	Module 2	04	2	15	4	02	4	06	4	09	0.00	**06**	4	10	4	05	0.00
A	Module 2	04	3	05	8	02	8	06	8	–01	–0.33	02	8	06	8	–01	–0.33
B	Module 2	04	3	06	2	02	2	06	2	00	0.00	02	2	06	2	00	0.00
H	Module 2	04	3	06	2	06	2	10	2	–04	–1.33	**02**	2	06	2	00	0.00
C	Module 1	10	3	10	10	06	10	16	10	–06	–1.67	06	10	16	10	–06	–1.67
N	Module 1	10	3	14	5	10	5	20	5	–06	–1.67	10	5	20	5	–06	–1.67
M	Module 1	10	5	14	15	10	15	20	15	–06	–1.00	10	15	20	15	–06	–1.00
										on time delivery score	–6.50					on time delivery score	–4.67
										inventory days	78					inventory days	39

Table 14.2 Alternate allocations on the efficiency frontier

Method 3 to allocate device_12 to module_1 and module_2												
Demands ordered by priority, date, qty							Method 3 Results					
ID	**Part**	**Cycle time**	**Priorit y**	**commit date**	**amt**	**need date**	**alloc date**	**all amt**	**stock date**	**stk amt**	**delta**	**otd score**
D	Module 1	10	1	12	15	02	02	15	12	15	00	0.00
G	Module 2	04	2	05	15	01	00	15	04	15	01	0.00
W	Module 2	04	2	07	4	03	02	4	06	4	01	0.00
L	Module 2	04	2	15	4	11	**10**	4	14	4	01	0.00
A	Module 2	04	3	05	8	01	02	8	06	8	–01	–0.33
B	Module 2	04	3	06	2	02	02	2	06	2	00	0.00
H	Module 2	04	3	06	2	02	**06**	2	10	2	–04	–1.33
C	Module 1	10	3	10	10	00	**10**	10	20	10	–10	–1.67
N	Module 1	10	3	14	5	04	**00**	5	10	5	04	0.00
M	Module 1	10	5	14	15	04	10	15	20	15	–06	–1.00
									on time score			–4.33
									inventory days			43

IBM's conclusion was that a smarter planning engine translated directly into improved supply chain efficiency and consistent performance. It was this understanding and general sense of urgency that kept the supply chain transformation, including the work on the planning engine, on track.

The result was the creation of the IBM Advanced Supply Chain planning team in January 1995, consisting of decision science and computer science professionals along with business analysts who had formerly had responsibility for operational planning. This team remains active today. One of this team's key missions was to build and deploy a CPE that could handle the scope and scale of the demand-supply networks at the IBM Microelectronics Division. This planning engine would need to operate at PN, order number, and lot level detail, be completely data driven, run either as a batch job at night or by planners (not as the personal tool kit of operations research professionals), and provide support for operational, tactical, and strategic planning and scheduling. A key challenge within this mission was to blend LP with heuristics in a manner that was transparent to the planners.

As we will describe in detail later, the complexity associated with managing demand-supply networks for semiconductor manufacturing makes it an ideal candidate for LP technology. However, the scale of the enterprise problem is too large to be solved by a single LP model. Simple heuristics such as greedy algorithms can handle the scale. But they fail to handle even such simple complexities as binning and typically underutilize expensive assets. To build a CPE, the IBM team developed and implemented a series of significant advances, including an LP model for binning optimization that could be invoked by a heuristic, an advanced heuristic, which provides highly effective decisions with regards to such items as demand priorities and binning (invoking the binning LP model), an advanced LP-based algorithm, which handles complex trade-offs and nonlinearities such as lot sizing and preemptive demand priorities, an overall solution structure, which dynamically blends both decision technologies and partitions the solution process in a manner that adds to the quality of the solution, and mechanisms to fully automate the process. The CPE is capable of solving enterprise-scale SCM problems with both function and speed. Its partitioning logic automatically classifies product structures and solves the most complex ones using large LPs, the moderately complex ones using small LPs, and the relatively simple ones using heuristics. The CPE achieves all this without any human intervention. The core of this work is described in the following sections.

On the professional side, this work has been recognized by IBM and INFORMS (Lyon et al. 2001; Denton et al. 2006). Perhaps, the best testimony came from Jerry Dundon (senior VP SCM, Analog Devices, Inc.) after the IBM CPE was successfully installed at Analog: "Much has been written about improving supply chain operations through reducing demand volatility and improving forecast accuracy. We have put equal focus on understanding and modeling the much-neglected supply side of the equation. Having a robust, credible model of supply capability allows us to be more flexible and responsive to inevitable changes to demand" (IBM white paper 2006).

The rest of this chapter proceeds as follows: basics of producing semiconductor-based microelectronic devices, overview of the core requirements of a CPE, challenges of complexity, description of the core solution mechanism in the IBM CPE, future work extending the big bang, and a conclusion.

14.2 The Basics of Producing Semiconductor-Based Microelectronic Components

Semiconductor manufacturing is a complex process involving everything from growing silicon crystals, manufacturing silicon wafers upon which integrated circuits (ICs) are built, to the actual placement and soldering of chips to a printed circuit board (Sullivan 1990; Kempf 1994, 2004; Leachman et al. 1996; Denton et al. 2006). Typically, the core manufacturing flow is wafers to devices to modules to cards (Fig. 14.7).

14.2.1 Wafer Fabrication

A wafer is a round, thin piece of silicon that looks like a CD. The goal of the wafer fabrication process is to build a set of ICs on the wafer surface according to a specific circuit design. The manufacturing process begins with cutting raw wafers from a silicon ingot. Next, circuit components (such as transistors or resistors) are sequentially built on the wafer surface and then interconnected to form ICs. At a high level, IC fabrication requires many repetitions of four essential steps: deposition (depositing special materials on the wafer surface), photolithography (forming

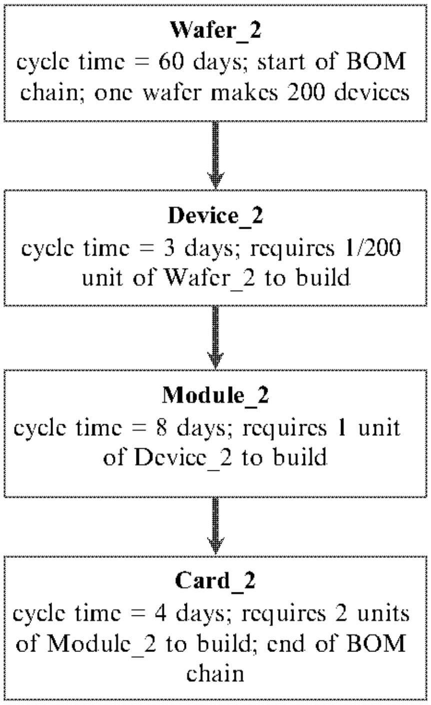

Wafer_2
cycle time = 60 days; start of BOM chain; one wafer makes 200 devices
Device_2
cycle time = 3 days; requires 1/200 unit of Wafer_2 to build
Module_2
cycle time = 8 days; requires 1 unit of Device_2 to build
Card_2
cycle time = 4 days; requires 2 units of Module_2 to build; end of BOM chain

Fig. 14.7 Simple flow for production of semiconductor-based package parts

circuit patterns by protecting only the deposited material which corresponds to circuit structures), etching (introducing chemical to remove the unprotected material), and ion implantation (modifying the wafer's conductive properties to complete the building of circuit components). These four steps are repeated for as many times as the design requires, resulting in a three-dimensional, layered structure on the two-dimensional wafer surface. In addition, toward the end of the fabrication, an iterative set of "metallization activities" occurs to connect the components. This repetitive use of the same core processes, and hence the same core equipment sets, is called "reentrant flow." For complex designs, it may take three to six hundred individual steps to complete the entire IC fabrication process. Cycle times range from 30 to 130 days, and yield (percentage of quality wafers) can show significant variation. There is also wide variation in the demand patterns for wafers. A wafer that is a component of a video game has large demands over a 6- to 12-month period. In contrast, usually wafers that support specialty products have sporadic demands and those used in sensing and testing devices have stable demands. Those involved in computers or control units have moderate demands over a reasonable period of time and then fall into an end of life phase that must cover warranty issues. Demand is stated either as a finished wafer or a planned start. For all of the complexity and uncertainty in wafer fabrication, all enterprise planning engines have a simple representation of this manufacturing process: a few key capacity points and a limited number of purely serial decision points.

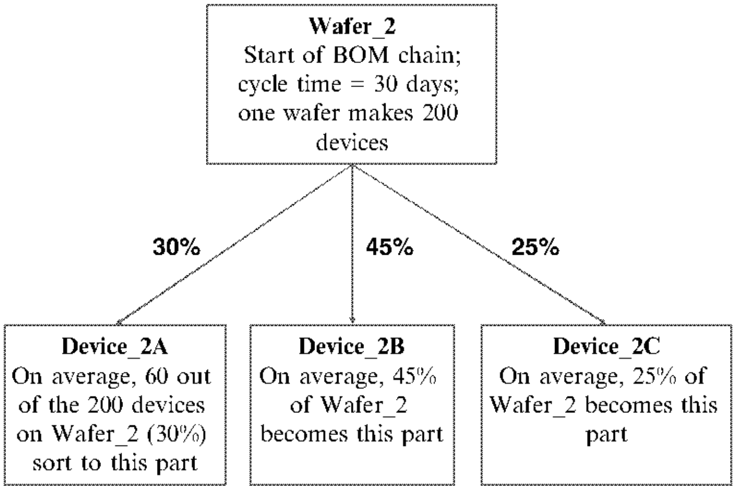

Wafer_2
Start of BOM chain; cycle time = 30 days; one wafer makes 200 devices
30%
45%
25%
Device_2A
On average, 60 out of the 200 devices on Wafer_2 (30%) sort to this part
Device_2B
On average, 45% of Wafer_2 becomes this part
Device_2C
On average, 25% of Wafer_2 becomes this part

Fig. 14.8 Simple binning or sorting situation from wafer to device

14.2.2 *Wafer Testing and Device Generation (Binning)*

Once the circuits have been built on the wafers, they are tested to determine the resultant yield of operational circuits (good or bad, speed, power consumption, etc.) and tagged for reference. The wafers are then diced (cut) into individual units and sorted or binned based on the prior testing creating what is referred to as the device or "die bank." This process is generally referred to as "binning" (illustrated in Fig. 14.8). Observe it is the device (not the wafer) that is packaged and placed into video games, cell phones, laptops, etc. A single wafer generates anywhere from 30 to 900 devices, and sometimes, the exact type of device is not determined until it is tested. That is, the testing not only determines whether the device is usable or not, but also determines its final part identity. Differences between devices from the same wafer typically occur as result of speed and power consumption. Wafer testing has a short cycle time (3 – 10 days) and involves a purely sequential process through test operations except for rework.

14.2.3 *Modules*

The device step is followed by a sequence of assembly steps to mount devices onto a substrate (called wire bonding) and package to make a module. Packaging protects the fragile device inside and makes it suitable to be incorporated in other electronic products. These modules are further tested to determine electromagnetic and thermal characteristics. As with devices, in some cases the final identity of the part may not be determined until the test activities are done. Again, the cycle times are short compared to fabrication times.

For some modules, a significant amount of complexity is introduced when the production process requires a sequence of packaging and testing steps and/or the

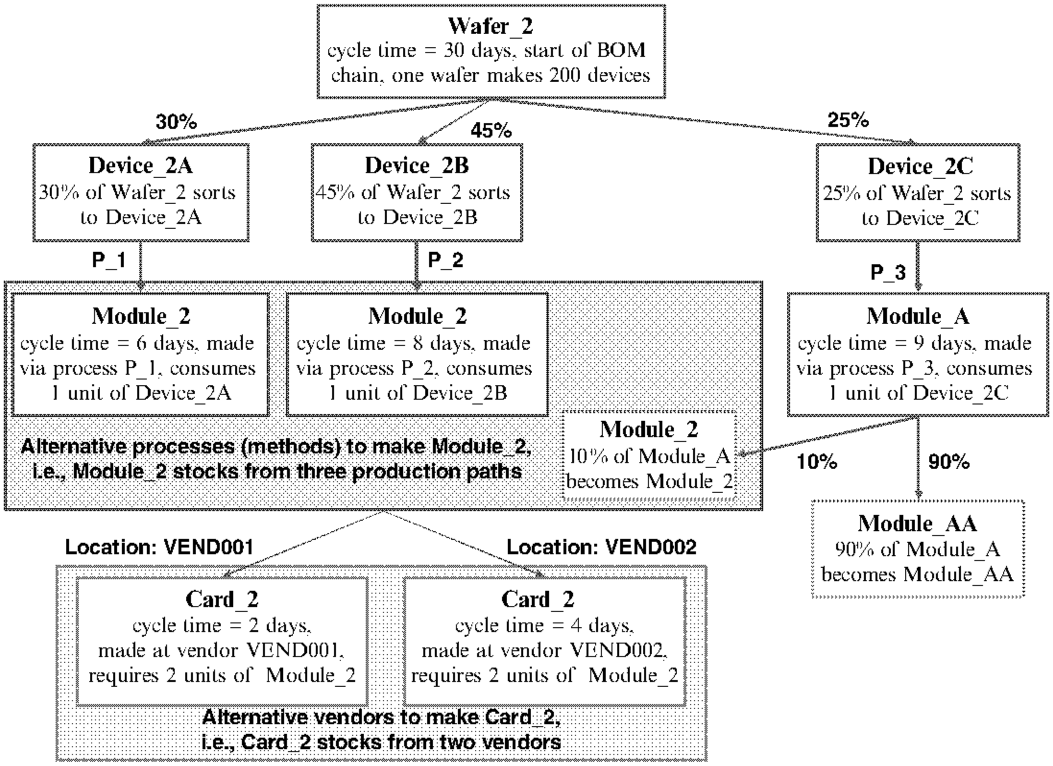

Wafer_2
cycle time = 30 days, start of BOM chain, one wafer makes 200 devices
30%
45%
25%
Device_2A
30% of Wafer_2 sorts to Device_2A
Device_2B
45% of Wafer_2 sorts to Device_2B
Device_2C
25% of Wafer_2 sorts to Device_2C
P_1
P_2
P_3
Module_2
cycle time = 6 days, made via process P_1, consumes 1 unit of Device_2A
Module_2
cycle time = 8 days, made via process P_2, consumes 1 unit of Device_2B
Module_A
cycle time = 9 days, made via process P_3, consumes 1 unit of Device_2C
Module_2
10% of Module_A becomes Module_2
Alternative processes (methods) to make Module_2, i.e., Module_2 stocks from three production paths
10%
90%
Module_AA
90% of Module_A becomes Module_AA
Location: VEND001
Location: VEND002
Card_2
cycle time = 2 days, made at vendor VEND001, requires 2 units of Module_2
Card_2
cycle time = 4 days, made at vendor VEND002, requires 2 units of Module_2
Alternative vendors to make Card_2, i.e., Card_2 stocks from two vendors

Fig. 14.9 Alternative processes to make Module_2, Module Binning, and Card_2

same module can be made from different processes that may or may not consume the same component part (a.k.a. alternate build paths). Figure 14.9 provides an example of this type of complexity.

14.2.4 Cards

The modules are eventually combined onto printed circuit boards to make cards. Finally, the cards are tested and those that pass are used in the assembly of a wide range of electronic products such as PCs, printers, and CD players. The cards are often produced at multiple vendor locations (Fig. 14.9).

14.2.5 Total Journey

To create manageable enterprise wide supply chain models, we introduce some abstraction from the complex routing of lots through the production process. We project the continuous process onto a discrete set of PNs. We use a BOM that specifies the component(s) of each PN to generate a graphical representation of the components needed for finished products, the alternative paths to produce the same PN, and the binning or sorting activities that determine the final status of a PN (Fig. 14.10).

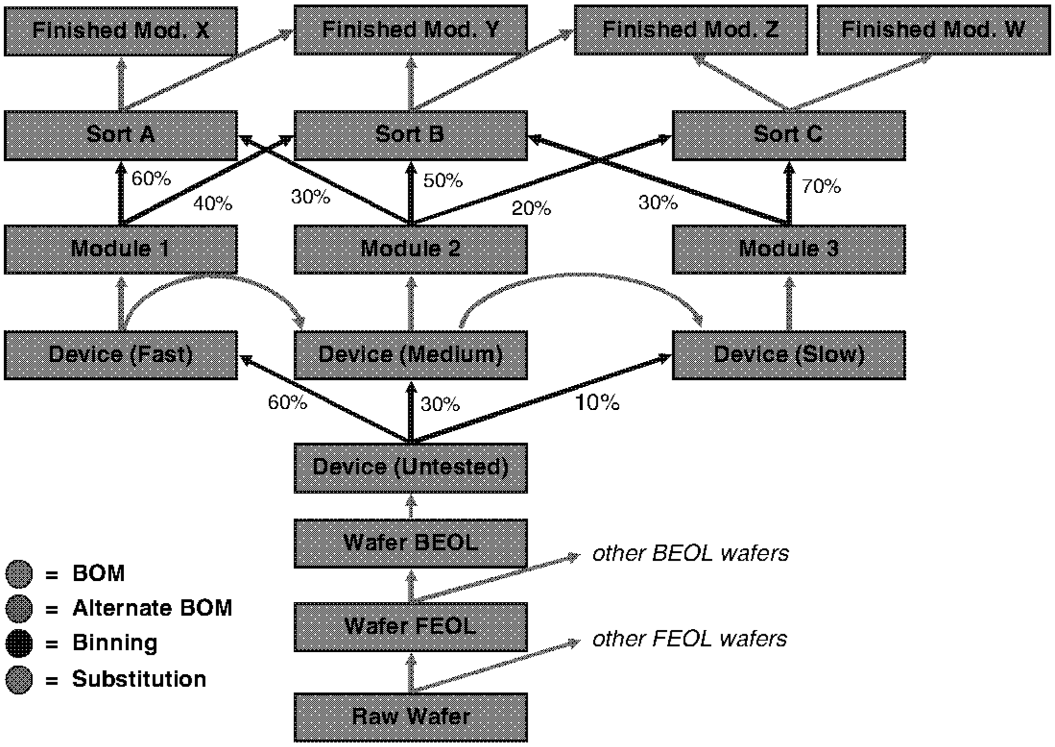


Fig. 14.10 Graphical representation of a bill of material structure

Observe the variety (complexity) inherent within the total "journey from sand to module or card." The first stage (fabrication of wafers and then testing and sorting into devices) has a long lead or cycle time, where the capital expense for equipment ranges from hundreds of millions to billions of dollars. There is no assembly at this stage; in fact, the number of PNs goes from a few to many. In stage two (modules) and three (cards), the cycle times are short and the process is assembly and test. Typically, more than one component is needed in the assembly process and there are multiple ways to produce the same part. The number of finished goods explodes compared to the original number of wafer parts. A manufacturing facility typically specializes in only part of the process. As an example of a common product flow: the wafer is produced at location A; it is tested and diced at location B; the module is assembled at location C1, C2, or C3; it is tested at location D1 and then sent to location E1 or E2 to be assembled onto a card or sent directly to a customer. Within this network, different PNs that support a wide range of customers are constantly competing for the same tool or machine capacity.

The complexity extends to variability across time as well. Most of the planning parameters and business preferences are date or time effective. This includes cycle times, yields, capacity required, capacity available, binning percentages, manufacturing processes, and so on. For example, in Fig. 14.9 the cycle time to produce Module_2 might reduce from 8 to 5 days in 3 weeks, and the entire process might be eliminated in 30 weeks.

To add to the complexity, *customer or exit demand may occur at any level* (card, module, device, wafer tested, and wafer untested), or be stated as required starts or allocation of capacity.

14.3 Overview of BCD Enterprise-Wide CPEs

The BCD enterprise-wide detailed CPE is the control point for the flow of material or product within an organization, and focuses on how to best meet prioritized demand without violating temporal, asset (WIP and inventory), or capacity constraints. A CPE application minimizes prioritized demand tardiness and some aspects of cost, establishing a projected supply and synchronized targets for each element of the supply chain.

The core of the CPE process is matching assets with demand, which refers to aligning assets with demand in an intelligent manner to best meet demands. The alignment or match occurs across multiple facilities within the boundaries established by the manufacturing specifications, process flows, and business policies. Assets include, but are not limited to, starts (manufacturing releases), WIP, inventory, purchases, and capacity (manufacturing equipment and manpower). Demands include, but are not limited to, firm orders, forecasted orders, and inventory buffer. The matching must take into account manufacturing/production specifications and business guidelines. Manufacturing specifications and process flows include, but are not limited to, build options, BOM, yields, cycle times, anticipated date on which a unit of WIP will complete a certain stage of manufacturing (called a receipt date), capacity consumed, substitutability of one part for another (substitution), the determination of the actual part type after testing (called binning or sorting), and shipping times. Business guidelines include, but are not limited to, frozen zones (no change can be made on supplies requested), demand priorities, priority trade-offs, preferred suppliers, and inventory policy. Many of the manufacturing specification and business guideline values will often change during the planning horizon (time effective).

The creation of a CPE plan requires a solver (sometimes referred to as a model or an engine) with the following core features:

1. Method(s) to represent the (potential) material flows in production, business policies, constraints, demand priorities, current locations of asset, etc., and relate all this information to exit demand.
2. Capture asset quantities and manufacturing specifications (parameters).
3. Search mechanism(s) to generate a balanced supply chain plan, relate the outcome to demand, and modify the plan to improve the match.
4. Display and explain the results of the BCD match.

The first task of any "BCD" CPE is to "flow material" and maintain a "feasible material flow" (see Graves et al. 1995 for a review of material flow control mechanisms). Simply put, the CPE must decide a sequence of manufacturing starts to produce finished goods, and for each start the CPE places into the plan, the required component parts and capacity must be available and the manufacturing activity must be permitted on that day (e.g., it is not a shutdown day). For example, in Fig. 14.7 if the CPE decides to manufacture 10 cards on day 10 to be completed on day 14 to meet a customer's demand, then on day 10 it must have 20 modules and tool/equipment capacity for the manufacturing process to consume. Typically, the CPE handles this requirement either implicitly with material balance equations or explicitly with explosion and implosion.

Explosion and implosion are the core processes of the CPE which either move work units (WIP or starts) forward (implosion) to project completed parts or backward (explosion) to determine starts required across the BOM supply chain following the appropriate manufacturing data such as cycle time, yield, capacity, and product structure. We typically use implosion to estimate what finished goods will be available to meet demand and explosion to estimate what starts are needed at what due dates to ensure meeting the existing demand on time.

To review implosion and explosion, consider Fig. 14.7 again which represents a simple production flow. The first manufacturing activity is the production of Wafer_2. This manufacturing activity has a cycle time of 60 days, i.e., it takes on average 60 days to take a raw wafer and create a completed wafer with the part ID Wafer_2. The second activity is device production. Creating 1 unit of Device_2 requires 3 days and consumes 1/200th unit of Wafer_2. Module_2 consumes 1 unit of Device_2 and takes 8 days to produce. Finally, Card_2 consumes 2 units of Module_2 and takes 4 days to produce.

Referencing Fig. 14.7, implosion can be illustrated with the following example. Manufacturing estimates that 4 units of Device_2 will be available or completed on day 10. This is called a projected receipt. If manufacturing immediately uses these 4 units to produce Module_2, on day 18 (10 + "Module_2 cycle time" = 10 + 8 = 18) 4 units of Module_2 will be completed. Continuing the projection process, the 4 units of Module_2 are immediately used to create 2 units of Card_2, which will be available on day 22 (18 + "Card_2 cycle time" = 18 + 4 = 22). The implosion process enables manufacturing to estimate the future supply of finished goods.

Again referencing Fig. 14.7, explosion can be illustrated with an example. To meet demand for 1 unit of Card_2 on day 20, the plant must have two (completed) units of Module_2 available on day 16 (20 − "*Card_2* cycle time" = 20 − 4 = 16). This generates an exploded demand of 2 units of Module_2 with a due date of day 16. To continue the explosion process, to produce the 2 units of Module_2, the plant must have 2 units of Device_2 available on day 8 (16 − "Module_2 cycle time" = 16 − 8 = 8). Next, the device demand is exploded creating a demand for 2/200th units of Wafer_2 on day 5 (= 8 − 3). This exploded information creates the guidelines for manufacturing to meet existing demand. For example, the device department must start production of 2 units of Device_2 no later than day 5 to meet the demand for 1 unit of Card_2 on day 20. Since the cycle time to produce Wafer_2 is 60 days, it needs to have one already in production and close to completion.

Within the explosion and implosion process is a method called "demand pegging." This method links each allocation of an asset or creation of a start with either a specific exit demand, or, at a minimum, the demand class or priority (relative importance of demand) associated with the exit demand being supported. Using the explosion example described above, if the exit demand for 1 unit of Card_2 on day 20 has a demand class of 3, each exploded demand will carry that demand class with it. Therefore, the units of Module_2 that are started on day 8 will have a demand class of 3. Similarly, if 3 units of Card_2 are desired on day 20 for a customer with demand class 5, then 6 units of Module_2 to be started on day 8 will also have a demand class of 5. The total required starts picture on day 8 is 8 (2 + 6), with 2 units with demand class 3 and 6 units with demand class 5. If by chance, there is only enough capacity on day 8 to start 2 units of Module_2, they will be allocated to the more important demand (demand class 3).

14.4 Challenges of Scope and Scale

"The great 20th century revelation that complex systems can be generated by the relationships among simple components" (Goldman 2004) applies to supply chain planning (and almost all aspects of planning, scheduling, and dispatch) (Little 1992).

Although simply creating a feasible central plan which maintains material balance, observes date effectivity, obeys business rules, captures existing WIP and inventory, and does a rough job at meeting demand is by itself challenging, it is no longer sufficient for a firm to remain competitive. The failure to "create a more accurate assessment of supply" forces the firm to compensate with slack (Galbraith 1973) or inefficiencies that leaves it at a competitive disadvantage. The purpose of this section is to identify and describe some of the key challenges a CPE must handle to provide an accurate assessment of supply. By "handle," we mean provide a mathematical representation of the individual characteristics, the system or relationships, and a method to search for an intelligent, if not provably, optimal solution. In the following paragraphs, we describe the challenges of demand class; simple binning; complex binning, substitution, and alternative bill of materials (BSA); lot sizing; sourcing; fair share; and commit date versus request date.

Leachman et al. (1996), Kempf (2004), and Denton et al. (2006) also provide excellent reviews of the complexities/challenges within the production of semiconductor-based parts. Leachman focuses on binning and demand class; Kempf develops and explains the stochastic challenge (uncertainty in demand and supply); and Denton deals with demand class and lot sizing in detail.

14.4.1 Allocating Perishable and Nonperishable Assets Based on Demand Class

A fundamental decision found in most aspects of planning and scheduling is deciding which demand "gets to go first?" That is, when more than one demand needs

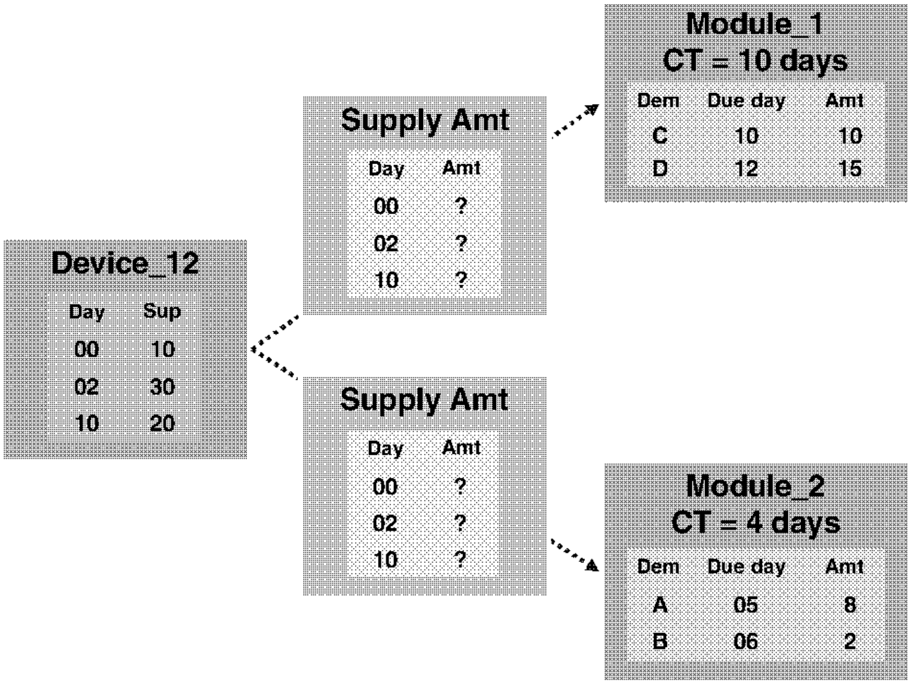


Fig. 14.11 Example of nonperishable asset allocation

either capacity (a perishable asset) or a component part (typically viewed as a nonperishable asset) and there is not sufficient supply to meet all immediate demand, the question is which demands get the asset and which have to wait. The CPE needs to allocate the asset based on the relative importance of the demand (indicated by demand class or priority) and the impact on delivering the finished goods on time.

Figure 14.11 shows a simple example of allocating supply of part (nonperishable asset) to meet demand. Module_1 and Module_2 are both made from Device_12 with cycle time 10 and 4 days, respectively. The demand for Module_1 is 10 units on day 10 and 15 units on day 12. The demand for Module_2 is 8 units on day 5 and 2 units on day 6. The key decision for the CPE is how to best allocate supply from Device_12 to the two modules (represented by the two boxes titled “Supply Amt” in Fig. 14.11).

One solution for the situation in Fig. 14.11 might be: (1) immediately allocate 8 of the 10 units of Device_12 on hand to meet demand A (8 units of Module_2 on day 5) 1 day early on day 4 (= 0 + 4); (2) immediately allocate the remaining 2 units of Device_12 on hand to meet demand B (2 units of Module_2 on day 6) 2 days early on day 4; (3) on day 2, allocate 10 units of the projected supply of 30 units of Device_12 to demand C (10 units of Module_1 on day 10) 2 days late on day 12 (= 2 + 10); (4) on day 2, allocate 15 units of the projected supply of 30 units of Device_12 to demand D (15 units of Module_1 on day 12) on time (12 = 2 + 10). The score card for this solution is demand A early, demand B early, demand C late by 2 days, and demand D on time. Table 14.3 summarizes this solution.

Table 14.3 Results of solution 1

		Commitment		Actual delivery		
Demand ID	Type	Date	Quantity	Date	Quantity	Delta schedule
A	Module_2	05	8	04	8	1
B	Module_2	06	2	04	2	2
C	Module_1	10	10	12	10	−2
D	Module_1	12	15	12	15	0

Table 14.4 Results of solution 2

		Commitment		Actual delivery		
Demand ID	Type	Date	Quantity	Date	Quantity	Delta schedule
A	Module_2	05	8	06	8	−1
B	Module_2	06	2	06	2	0
C	Module_1	10	10	10	10	0
D	Module_1	12	15	12	15	0

A second option could be: (1) 10 units of Device_12 are allocated to Module_1 on day 0 to cover demand C; (2) 15 units of Device_12 are allocated to Module_1 on day 2; and (3) 10 units of Device_12 are allocated to Module_2 on day 2. The score card for this solution is: demand A is met 1 day late, demand B is met on time, demand C is met on time, and demand D is also met on time. Table 14.4 summarizes this solution.

Which is better? If the four module demands are exit demands, the answer depends only on the relative importance of each demand and the business policy on "sharing the pain" if a demand cannot be met on time. If demand A is demand class 1 (the lower the value, the more important the demand) and demands C and D are demand class 3, the first solution is the logical choice. If the demand classes are reversed, the second solution is the logical choice. What if all of the demands have the same demand class? Do we go with solution two since demand A is just one day late? Do we meet part of demands C and A on time and the other part is late (can we split the order)? Or do we meet all of demand C and part of demand A on time? This typically depends on the business policy of the enterprise.

If the four module demands are not exit demands, then in addition to the demand class, we need to also assess whether meeting the module demand on time ensures meeting the exit demand on time. For example, if demand A goes into the exit demand for CARD01 for the XYZ customer and the board (modules go on boards to make a card) required is 2 weeks late, there is no point in worrying about meeting module demand A on time.

In Sect. 14.1.4, we introduced a slightly more complicated example of allocating Device_12 to the production of Module_1 and Module_2 with additional supply and demand. Here, we identified three possible allocation schemes. The anticipated demand and supply are displayed in Fig. 14.12. One allocation solution is displayed in Fig. 14.13 with an OTD score of −6.50 and an inventory score of 78.

Anticipated Supply Device_12				
Day	Supply	Cum sup	Allocate	Remain
00	20	20	00	20
02	30	50	00	30
06	10	60	00	10
10	20	80	00	20

Allocate supply of Device_12 to demands for Module_1 & Module_2

Demands					
ID	Part	Cycle time	Priority	Commit Date	Amt
G	Module_2	04	2	05	15
A	Module_2	04	3	05	8
B	Module_2	04	3	06	2
H	Module_2	04	3	06	2
W	Module_2	04	2	07	4
C	Module_1	10	3	10	10
D	Module_1	10	1	12	15
N	Module_1	10	3	14	5
M	Module_1	10	5	14	15
L	Module_2	04	2	15	4
total					80

Fig. 14.12 Example of anticipated demand and supply

Method 1: order dmd by priority, date, quantity, allocate by priority, commit date, do not split a delivery												
Demands ordered by priority, date, qty						allocations , production, delivery						
order ID	Part	Cycle time	Priority	commit date	amt	alloc date	all amt	stock date	stock amt	delta	otd score	inv days
D	Module 1	10	1	12	15	00	15	10	15	02	0.00	30
G	Module 2	04	2	05	15	02	15	06	15	-01	−0.50	0
W	Module 2	04	2	07	4	00	4	04	4	03	0.00	12
L	Module 2	04	2	15	4	02	4	06	4	09	0.00	36
A	Module 2	04	3	05	8	02	8	06	8	-01	−0.33	0
B	Module 2	04	3	06	2	02	2	06	2	00	0.00	0
H	Module 2	04	3	06	2	06	2	10	2	-04	−1.33	0
C	Module 1	10	3	10	10	06	10	16	10	06	−1.67	0
N	Module 1	10	3	14	5	10	5	20	5	-06	−1.67	0
M	Module 1	10	5	14	15	10	15	20	15	-06	−1.00	0
								on time & inv score			−6.50	78

allocation history							
Day	Supply	cum sup	allocate	remain	cum rem	total	cumtot
00	20	20	19	01	01	19	19
02	30	50	29	01	02	29	48
06	10	60	12	−02	00	12	60
10	20	80	20	00	00	20	80

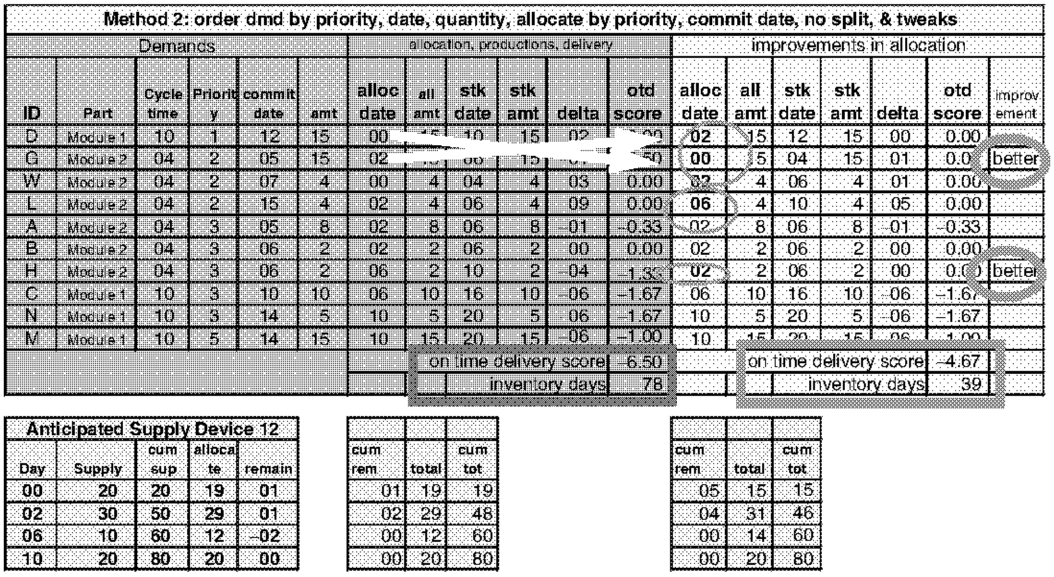
Fig. 14.13 Possible allocation solution to the example in Fig. 14.12

The row for demand D for Module_1 can be read as follows: 15 units from the supply of Device_12 available on day 00 are allocated to meet this demand. Fifteen units of Module_1 comes to stock (is completed) on day 10 and is allocated to demand D. Since the commit date for demand D is day 12 and the supply is available on day 10, the supply is 2 days earlier (this is the delta column). The OTD score is 0. The algorithm for OTD scoring is: (a) if the demand is met on time or early the OTD score is 0; (b) if the demand is late the smaller of the number of days late and -5 is divided by the demand priority (this caps the "days late" at 5 and weighs it inversely to the demand priority). The inventory score is 30. The algorithm for inventory scoring is: (a) 0 if the supply is just in time or late to meet demand, (b) if the supply is early, then the score is the number of days early times the number units of supply.

The bottom table (allocation of supply) of Fig. 14.13 summarizes when each supply is used. For example, the day 00 row tells us 20 units of Device_12 are available on day 00, out of these 20, 19 are immediately allocated to build modules and 1 unit is held for later use. If we examine the "alloc date" (allocation date) column of the top table, there are two rows with 00 in them – rows D and W. If we now examine the column "all amt" (allocation amount), the number of units of Device_12 allocated for demand D and W (with day 00 allocation) are 15 and 4, respectively. This totals 19 matching what is in the bottom table.

A second allocation solution is displayed in Fig. 14.14 with an OTD score of −4.67 and an inventory score of 39. This "smarter" solution improves both inventory and OTD and moves the efficiency frontier (Fig. 14.5) out. This improvement comes from:

1. Swapping the allocation of Device_12 between demand D (allocation moves from day 00 to day 02) and demand G (allocation moves from day 02 to day 00) – which improves the OTD of demand G and reduces the inventory days accumulated with demand G
2. Delaying the allocation of Device_12 for demand L from day 02 to day 06 – which maintains OTD, but reduces inventory days
3. Moving the allocation of Device_12 from demand C from day 06 to day 02 – which generates an OTD



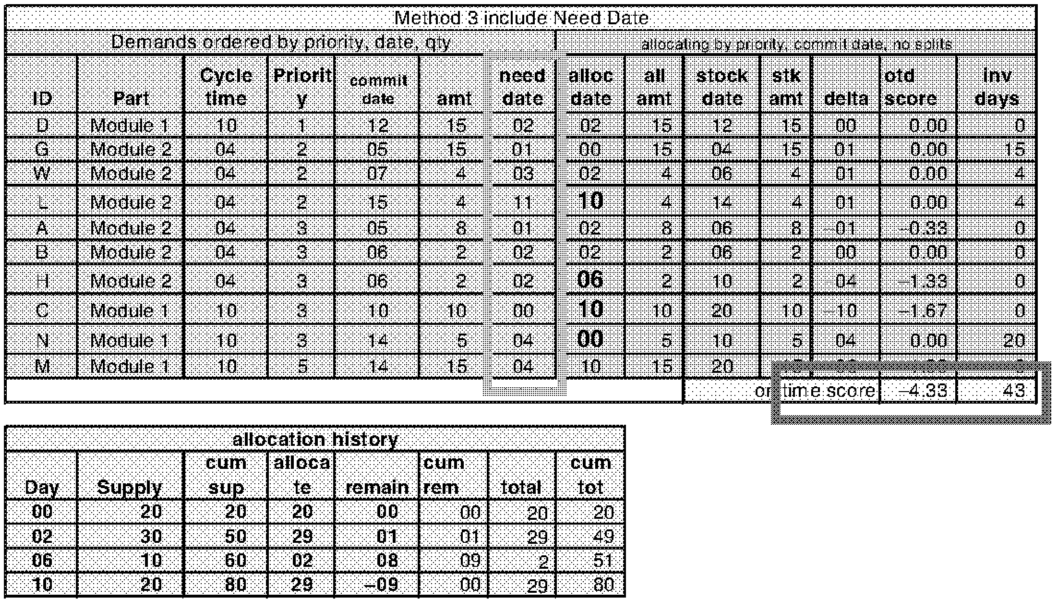
Method 2: order dmd by priority, date, quantity, allocate by priority, commit date, no split, & tweaks																		
Demands						allocation, productions, delivery						improvements in allocation						
ID	Part	Cycle time	Priority	commit date	amt	alloc date	all amt	stk date	stk amt	delta	otd score	alloc date	all amt	stk date	stk amt	delta	otd score	improvement
D	Module 1	10	1	12	15	00	15	10	15	02	0.00	02	15	12	15	00	0.00	
G	Module 2	04	2	05	15	02	15	06	15	−01	−0.50	00	15	04	15	01	0.00	better
W	Module 2	04	2	07	4	00	4	04	4	03	0.00	02	4	06	4	01	0.00	
L	Module 2	04	2	15	4	02	4	06	4	09	0.00	06	4	10	4	05	0.00	
A	Module 2	04	3	05	8	02	8	06	8	−01	−0.33	02	8	06	8	−01	−0.33	
B	Module 2	04	3	06	2	02	2	06	2	00	0.00	02	2	06	2	00	0.00	
H	Module 2	04	3	06	2	06	2	10	2	−04	−1.33	02	2	06	2	00	0.00	better
C	Module 1	10	3	10	10	06	10	16	10	−06	−1.67	06	10	16	10	−06	−1.67	
N	Module 1	10	3	14	5	10	5	20	5	−06	−1.67	10	5	20	5	−06	−1.67	
M	Module 1	10	5	14	15	10	15	20	15	−06	−1.00	10	15	20	15	−06	−1.00	
							on time delivery score				−6.50		on time delivery score				−4.67	
								inventory days			78			inventory days			39	

Anticipated Supply Device 12				
Day	Supply	cum sup	allocate	remain
00	20	20	19	01
02	30	50	29	01
06	10	60	12	−02
10	20	80	20	00

cum rem	total	cum tot
01	19	19
02	29	48
00	12	60
00	20	80

cum rem	total	cum tot
05	15	15
04	31	46
00	14	60
00	20	80

Fig. 14.14 Second possible allocation solution to the example in Fig. 14.12



Method 3 include Need Date													
Demands ordered by priority, date, qty							allocating by priority, commit date, no splits						
ID	Part	Cycle time	Priority	commit date	amt	need date	alloc date	all amt	stock date	stk amt	delta	otd score	inv days
D	Module 1	10	1	12	15	02	02	15	12	15	00	0.00	0
G	Module 2	04	2	05	15	01	00	15	04	15	01	0.00	15
W	Module 2	04	2	07	4	03	02	4	06	4	01	0.00	4
L	Module 2	04	2	15	4	11	**10**	4	14	4	01	0.00	4
A	Module 2	04	3	05	8	01	02	8	06	8	−01	−0.33	0
B	Module 2	04	3	06	2	02	02	2	06	2	00	0.00	0
H	Module 2	04	3	06	2	02	**06**	2	10	2	04	−1.33	0
C	Module 1	10	3	10	10	00	**10**	10	20	10	−10	−1.67	0
N	Module 1	10	3	14	5	04	**00**	5	10	5	04	0.00	20
M	Module 1	10	5	14	15	04	10	15	20	15	[illegible]	[illegible]	0
											on time score	−4.33	43

allocation history							
Day	Supply	cum sup	allocate	remain	cum rem	total	cum tot
00	20	20	20	00	00	20	20
02	30	50	29	01	01	29	49
06	10	60	02	08	09	2	51
10	20	80	29	−09	00	29	80

Fig. 14.15 Third possible allocation solution to the example in Fig. 14.12

A third solution is displayed in Fig. 14.15 with an OTD score of -4.33 and an inventory score of 43. This shifts the firm along the efficiency frontier by trading better OTD for worse inventory, presumably reflecting the firm's current business strategy.

Observe the CPE needs three core functions to handle this type of decision: (1) being able to "peg" or pass down the priority of the exit demand to all intermediate demands, (2) a method to pass in and represent the business policy, and (3) mechanisms to search for options and assess the trade-offs relative to overall efficiency and current business policy.

14.4.2 Simple Binning (or Sorting) with Downgrade Substitution

Simple binning with downgrade substitution refers to the classification of a part into one of a set of mutually exclusive and exhaustive categories based on some key performance factor using a series of testing procedures conducted after completing a set of manufacturing processes (see Leachman et al. 1996 for additional details on binning). In semiconductor manufacturing, the most common (but not the only) location for binning is the completion of wafer fabrication, where hundreds of devices are cut from a single finished silicon wafer. Because of the random variation in the wafer fabrication process, working conditions for the devices on the same wafer vary. Therefore, before proceeding any further in the manufacturing process, devices have to be classified into different categories (each having a unique PN) after being cut from the finished wafer (typically today, the actual testing is done on the wafer). Clock speed, for example, is usually among a number of key performance

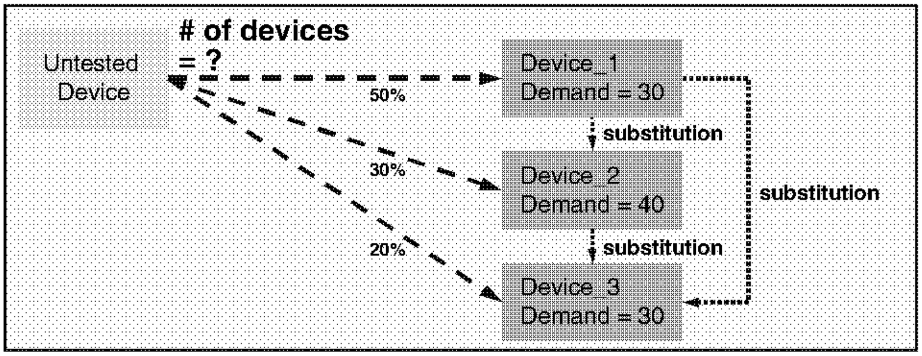


Fig. 14.16 Simple binning with downgrade substitution after wafer fabrication

factors to be tested for each device. Besides electronics, downgrade substitution can also be seen in such industries as consumer goods (bicycles) and building materials (grades of wood).

Figure 14.16 illustrates a typical binning scenario after the water fabrication process, where 50% of the time, the device is tested to have a "grade A" or top performance; 30% of the time, it has a "grade B" or medium performance; and 20% of the time, it has a "grade C" or low performance. These percentages are referred as binning percentages and can generally be observed in semiconductor manufacturing as a result of binning testing. Another phenomenon commonly associated with binning is substitution between materials of the same type. Quite often, parts with a higher performance can substitute for parts with a lower performance if necessary. This form of material substitution is generally called downgrade substitution, and typically, it occurs when there is a shortage of the lower performance part accompanied by an overage of the higher performance part. In Fig. 14.16, the dotted arrows indicate that "grade A" devices (Device_1) can substitute for both "grade B" devices (Device_2) and "grade C" devices (Device_3), and "grade B" devices can substitute for "grade C" devices. The binning percentages can and do change over time.

The challenge is to make an optimal use of coproducts and substitution to avoid overstating the number of wafer starts required to meet demand. If the demand is 30 for Device_1, 40 for Device_2, and 30 for Device_3 (Fig. 14.16), the challenge is to determine the minimum number of wafers/devices that must be produced to meet all three demands. One simple rule is the maximum of the quantity required for each device divided by its binning percentage. Continuing the same example, we would need 150 devices, which equals maximum (30/0.50, 40/0.30, 30/0.20) = maximum (60, 133, 150). As shown in Fig. 14.17, such a rule will leave an excess inventory of 50 devices, with 45 contributed by Device_1 and 5 contributed by Device_2. If we optimally account for coproducts and substitutions, the minimum number of devices required to meet all three demands is 100 – testing 100 devices creates 50 of Device_1, 30 of Device_2, and 20 of Device_3 (Fig. 14.18). The extra 20 of Device_1 are used to cover the shortfall of 10 of both Device_2 and Device_3.

Other factors complicating the determination of the minimum number of starts required to meet demand in simple binning production structures include: demands

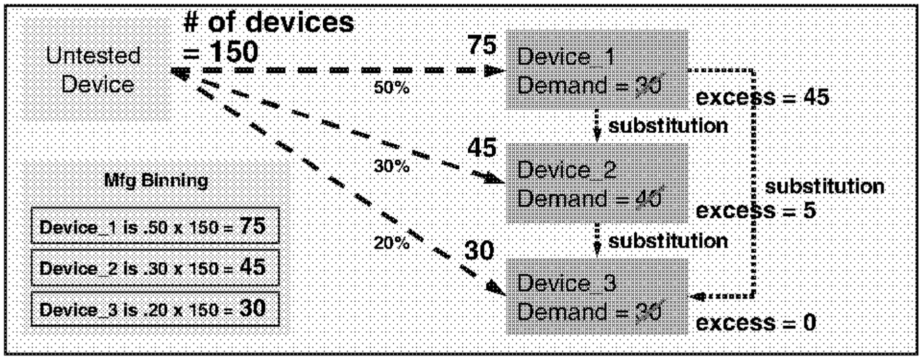


Fig. 14.17 Maximum quantity of starts leaves excess inventory of 50 devices

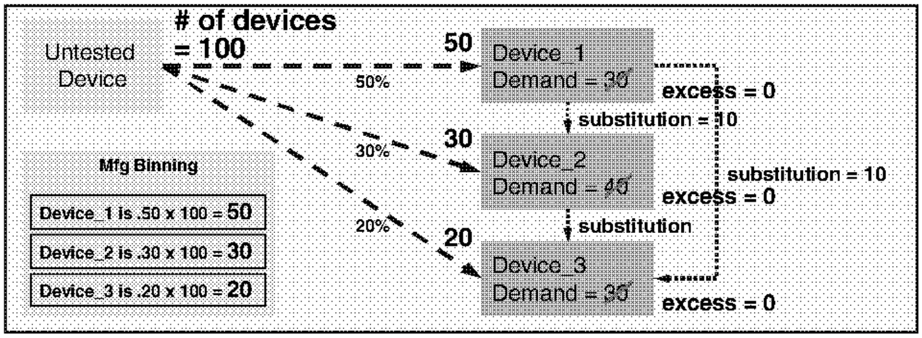


Fig. 14.18 Optimal number of devices meets all demands and leaves no inventory

for devices that spread throughout the planning horizon, existing inventory, projected WIP completion, and binning percentages and allowable substitutions that change during the planning horizon (date effectivity). In addition, the CPE must locate and isolate these binning situations in a large, complex demand-supply network, as well as maintain full traceability and handle demand priorities (Fig. 14.19).

14.4.3 BSA: Complex Binning, General Substitution, and Alternative BOMs

Within the production of modules, an increasingly common manufacturing characteristic is alternative BOM structures, general substitution, and complex binning. In its simplest form, alternative BOM means two or more manufacturing processes are available to produce the same PN. For example, Fig. 14.20 illustrates a scenario in which process P1 and P2 can both be used to produce Module_9. If P1 is selected, the process will consume Device_8A; if P2 is selected, the process will consume Device_8B. Complex binning refers to a situation, where one binning activity immediately invokes another, or substitutions are permitted across binning activities. This is illustrated in Figs. 14.9 and 14.10.

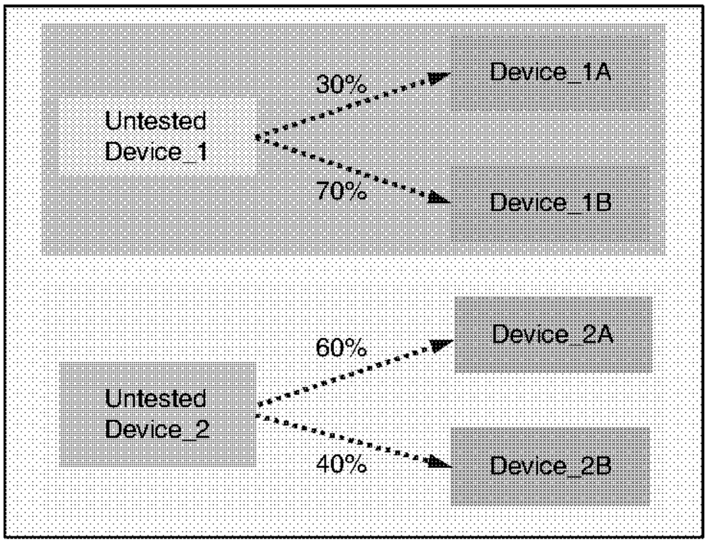


Fig. 14.19 Two binning situations that would be solved by separate LPs

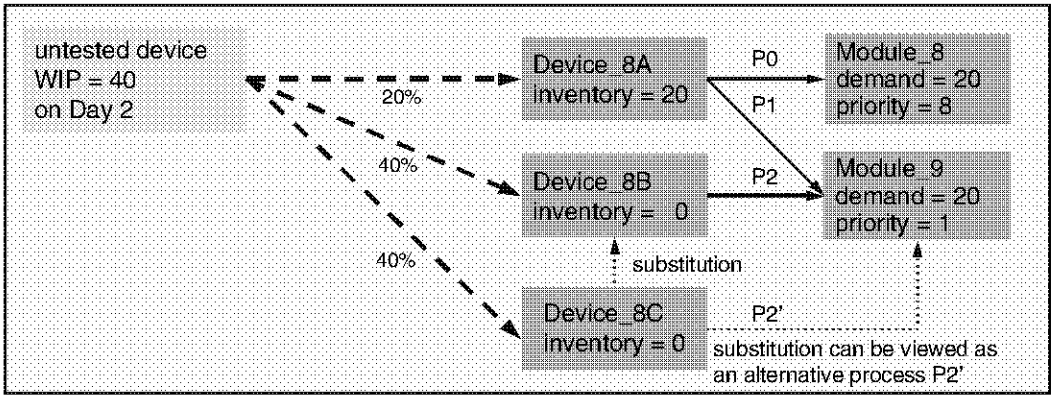


Fig. 14.20 Illustration of alternative BOM structure with substitution

To get a sense for the decision challenges created by alternative BOMs, we will use Fig. 14.20 to look at just the "explosion" question within the CPE. Two processes (P1 and P2) can be used to build Module_9, where P1 consumes Device_8A and P2 consumes Device_8B. In addition, Device_8C can generally be substituted for Device_8B, which means if Device_8B is not available and Device_8C is available, process P2 can use Device_8C to make Module_9. Conceptually, this general substitution can be viewed as the third alternative BOM option (call it P2-prime or P2′).

The explosion engine must determine how to explode demand for Module_9 and Module_8 back to the device level. Should it be half to P1 and half to P2, 2/3 to P1 and 1/3 to P2, or all to P1? Should P2′ be considered? The objective is to divide the demand for Module_9 across P1 and P2 (and perhaps P2′) to best use the existing

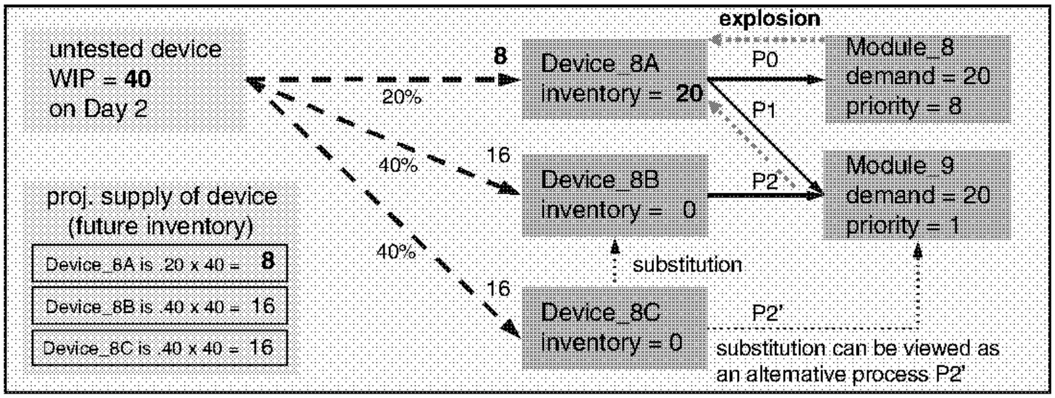


Fig. 14.21 Awareness of asset availability affects new starts

inventory and WIP, minimize new starts, and meet other relevant guidelines such as sharing percentages and capacity. Determining the best result requires an extensive search through the entire BOM structure.

In Fig. 14.20, assume the priority for demand for Module_9 (which is 1) is higher than that for Module_8, and 20 units are ordered for both modules. A quick search reveals that 20 units of Device_8A are in inventory, which can be used to make Module_9 with process P1 or Module_8 with process P0. There is no current inventory for Device_8B or Device_8C to make Module_9 using process P2 or P2′ (the substitution). Most heuristic-based search engines that guide the explosion through alternative BOM structures would explode the 20 units of demand for Module_9 down the P1 process (or leg). This would consume all of the 20 units in inventory for Device_8A and leave the demand for Module_8 totally uncovered; assuming building every unit of module would consume 1 unit of device (Fig. 14.21). To meet the demand for Module_8, a planning engine would require the demand-supply network to produce 100 new untested devices if the engine was not aware of the expected projection of the 40 untested devices (which could be tested into 8 units of Device_8A (= 40 × 0.20), 16 units of Device_8B (= 40 × 0.40), and 16 units of Device_8C (= 40 × 0.40)). On the contrary, if a planning engine is aware of this information, it would only require 60 new untested devices since there is a projection of 8 Device_8A ($60 = (20 - 8)/0.20 = 12/0.20$).

However, a broader search would uncover all available options at untested device and avoid the conflict for Device_8A between Module_8 and Module_9. There are 20 units of demand for Module_8 and it can only be made from Device_8A. So the question is: are there other options to meet the demand for Module_9? There are 40 units of projected WIP at the untested device and after binning, 8 will become Device_8A, 16 will become Device_8B, and 16 will become Device_8C. Since Device_8C can be substituted for Device_8B, there are actually 32 (16 + 16) future devices that can be used to produce Module_9 (but not Module_8), which is more than enough to meet the 20 units of demand for Module_9. So it is probably not optimal to explode the demand for Module_9 down the P1 leg. As illustrated in Fig. 14.22, both demands can be met without using any new untested devices:

1. Assign the 20 units of inventory for Device_8A to be used to produce 20 units of Module_8 (blue dotted arrow).
2. Explode the 20 units of demand for Module_9 into a need for 20 units of Device_8B (red dotted arrow). Net this need against the 16 projected Device_8B, resulting in a need for four new Device_8B.
3. Use 4 out of the 16 projected Device_8C to meet the need for four new Device_8B.

Appendix A contains an additional example of the limitations of heuristics to handle alternative BOM paths. Figure 14.23 illustrates how the complexity can grow quickly and Fig. 14.24 illustrates potential opportunities for parallelization through dynamic partitioning (indicated by different colors).

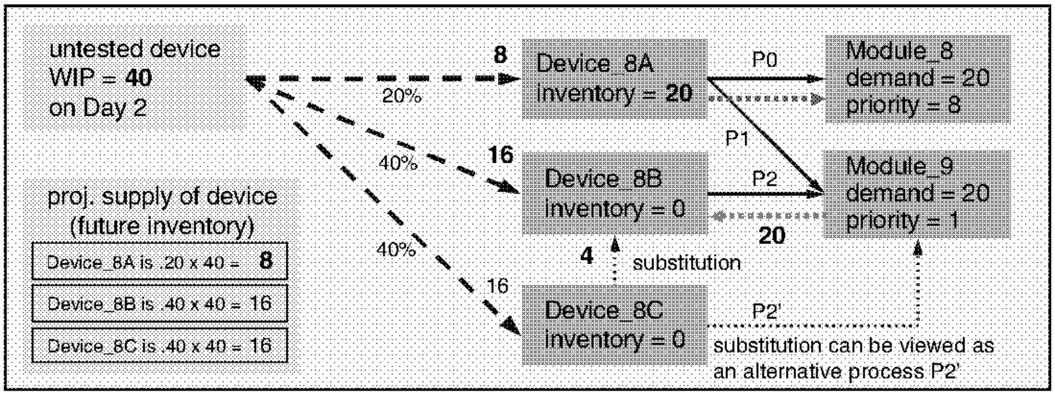


Fig. 14.22 A smarter option – meeting both demands without using new devices

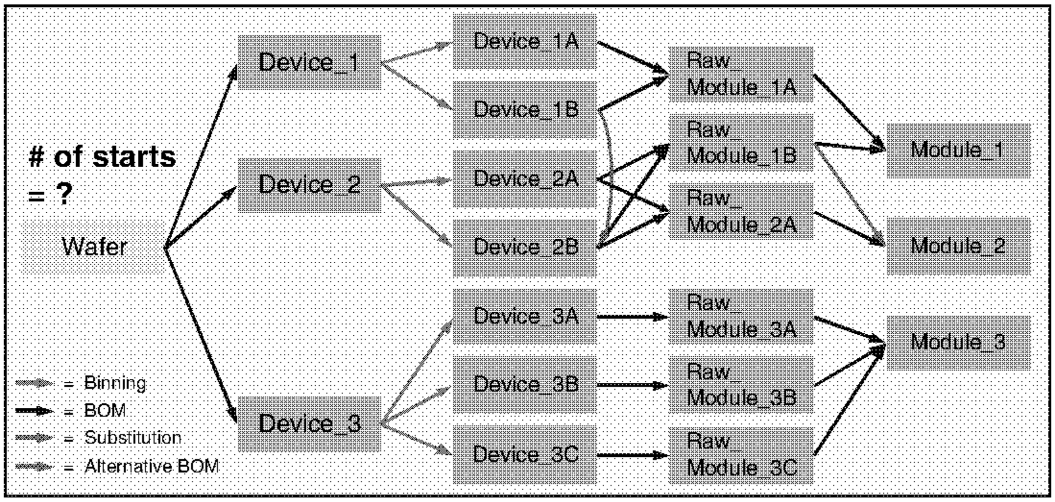


Fig. 14.23 Complexities across the bill of material structure

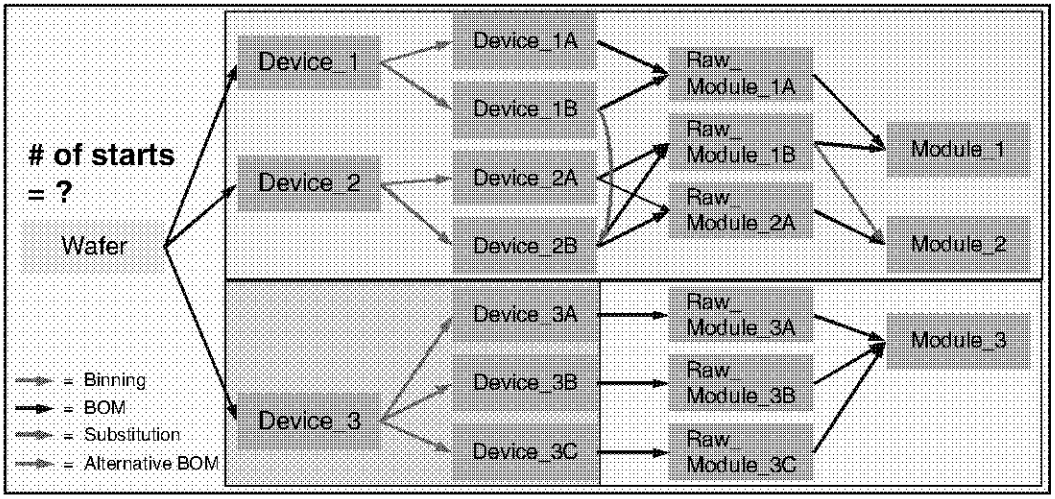


Fig. 14.24 Opportunities for parallelization through natural partitions

Increasingly, the CPE and the planner have to handle all three complexities: alternative BOM, general substitution, and complex binning. Figure 14.10 illustrates this "bundling." The following is an example scenario provided by one of the IBM's most experienced planners working with IBM and customer products.

The performance (speed) is specified at the module level when the module process including stress is complete. The module test spec is defined with safety tolerance that guarantees the module will perform to its specification in the application. This performance "spec" is then translated back to device test. The device test spec is also defined with tolerance, "Guard Band" to insure that a "fast" device at device test remains a "fast" device once it has been packaged at the module level. This translation of the module test spec back to device is not 100% perfect. Often times, a different test platform is used for device and module test, further introducing a margin of error.

You might ask the question of why do device test at all if it is not accurate. There are two primary reasons for this.

First, there can be a lot of variation in the speed distribution from one wafer lot to another. The speed limitations, at a high level, are defined by the design. But depending on where in the process window the lot was processed, you can end up with a "fast" lot, a "medium" lot, or a "slow" lot. Even though there is some margin of error from device test to module test, speed sorting at device allows you to be able to select the right device to package to best meet the demand. If you waited until the module is packaged, it may be too late.

Example: Let us say that all of your demand for the next few weeks is on the medium speed module. You would want to release medium devices as opposed to fast devices, thus avoiding down binning of fast modules to medium modules. Also if you did not do device test and you encounter a "slow" lot, you might not end up with enough "medium" modules to meet your demand and miss committed orders.

Second, the attached BOM assumes one package type. In many cases, devices can be packaged in different module types. These packaging options can serve entirely different markets with different speed requirements.

Example: You have two package types with different overall speed requirements as follows:

Package 1	Fast	Medium	Slow
Market Speed requirements Gaming Industry	0%	80%	20%
Package 2	Fast	Medium	Slow
Market Speed rqmt's High End Servers	40%	60%	0%

Obviously, you would never release "fast" devices to Package 1 and have to down bin all the fast module and loose margin as "fast" always commands premium dollars. Also you would never release "slow" devices to Package 2 as there is no market for "slow" modules in this package and you would suffer a lot of excess scrap.

The need for reasonably sophisticated decision technology to navigate BSA and find a true picture of a firm's potential supply is self evident – it can make or break the profitability of a firm. Remember all of the other complexities carry over: demand priority, WIP, inventory, date effectivity, capacity availability, etc.

14.4.4 Lot Sizing

Lot sizing refers to a core manufacturing characteristic that the number of units in each lot of activity can have a significant impact on productivity. This preference/ requirement is typically described by minimum lot size, maximum lot size, and multiples. Probably, the easiest and most common example is minimum lot sizing.

Within a manufacturing facility, the same tool set is used to process a variety of products. For example, the same testing equipment is used to investigate and sort a wide variety of modules. However, there is a substantial "setup cost" when the tool set switches from one part to another. Therefore, the manufacturing unit wants to get a "return" on this setup cost by processing a minimum quantity of each part. It does not want to do 3 of part A, then 5 of part B, then 2 of Part C, etc.; rather, it wants to do 30,000 of part A, then 5,000 of part B, then 20,000 of part C, etc.

This requirement creates a set of challenges for the CPE. First, the engine has to accommodate that in some cases the manufacturing release number can only come in discrete quantities (challenge for LP). Second, the engine needs a mechanism to keep track that the same lot will be accommodating different demand priorities. Third, not all lot sizing requirements are firm: the manufacturing facility may prefer to process 30,000 of part A instead of 3, but it can do just 3 if it is important enough.

14.4.5 Sourcing

If an enterprise has more than one supply location which provides certain parts, it typically wants to maintain some type of balance between the workload on each facility. This "balance" is typically described with a reasonably complex set of business rules. Often, these rules arise from complex contractual obligations to suppliers that vary over time and business conditions.

14.4.6 Fair Share

Fair share refers to sharing limited supply or capacity among a set of equally important customers, as opposed to filling their orders in some random fashion. For example, A and B are the firm's most important customers, and they both have just requested 100,000 units of the same part to be delivered on the same day (200,000 units in total). Suppose the available supply is only 140,000 units, 60,000 short of the combined demand quantity. If no concern is given to the final delivery quantity, the CPE may generate a supply chain plan, which ships 100,000 units to A (i.e., A's order 100% filled) and 40,000 units to B (i.e., B's order only 40% filled) despite the fact that they are equally important. With fair share, on the contrary, both customers will have 70% of their order delivered (and share the same degree of "pain"). The importance that all customers with the same level of importance to the company receive the same level of service is quite obvious. This requirement creates an additional challenge for the CPE.

14.4.7 Customer Request Versus Customer Commit

Customer request versus customer commit (date, quantity, priority) is emerging as a key complexity directly affecting customer satisfaction. Typically, a customer requests to receive a certain number of part(s) on a certain date, and this is the customer request date and request quantity. Next, the enterprise reviews the request and responds with a date and quantity to which it will commit which becomes the commit date and commit quantity. The customer and supplier may iterate a few times on date and quantity until a firm commit date and quantity are established. The supplier often associates some priority with the firm demand. Historically, this is the only date, quantity, and priority the CPE has for each demand, that is, without using the original request date and quantity information during its solution process. In current practice, at best some postprocessing activity occurs to identify "low hanging" opportunities to meet a few key request dates.

To start, let us consider the simple example illustrated in Fig. 14.25, where there is a demand for 100 units of part A with a request date of 06/01 and a commit date of 07/01. Assume WIP exists projected to come to stock on 06/01. A two-pass solution

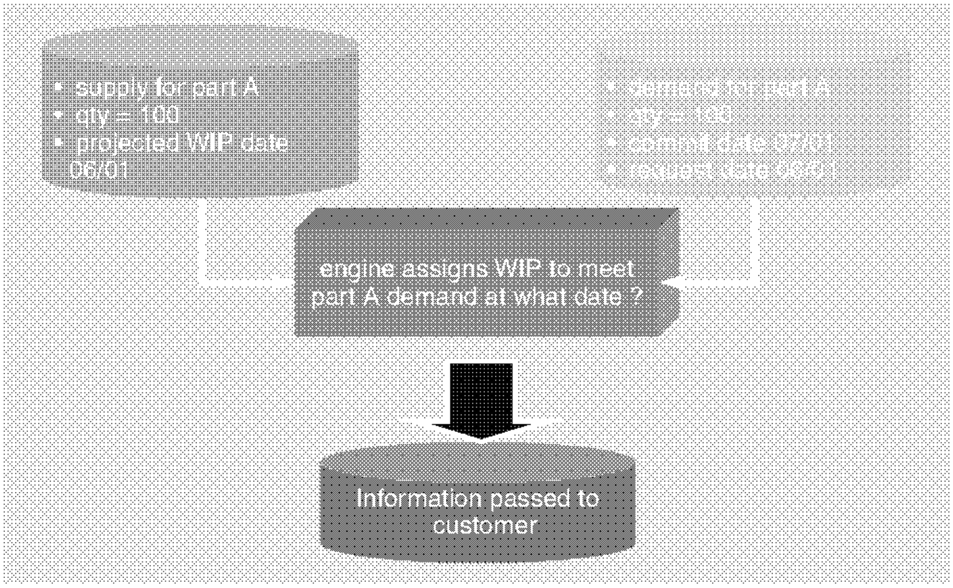
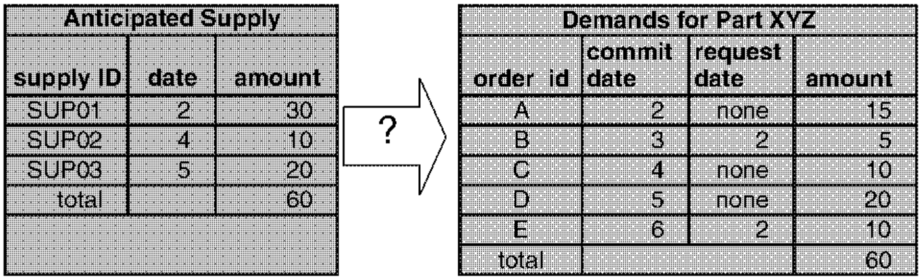

• supply for part A
• qty = 100
• projected WIP date 06/01
• demand for part A
• qty = 100
• commit date 07/01
• request date 06/01
engine assigns WIP to meet part A demand at what date ?
Information passed to customer

Fig. 14.25 Simple example of handling commit and request dates



Anticipated Supply		
supply ID	date	amount
SUP01	2	30
SUP02	4	10
SUP03	5	20
total		60

Demands for Part XYZ			
order id	commit date	request date	amount
A	2	none	15
B	3	2	5
C	4	none	10
D	5	none	20
E	6	2	10
total			60

Fig. 14.26 Demand and supply information for Part XYZ

could do the following: in the first pass, WIP is used to cover the 07/01 commit date and then netted out and became unavailable for the second pass. In the second pass, no unnetted asset remains to cover the 06/01 request date. The two-pass process did not pick up the possibility that WIP could cover the request date. A solution that carried both the commit and request date would recognize the WIP could be used to make the request date with the primary search process.

A more complicated example is shown in Fig. 14.26. Here, we have five demands for part XYZ and anticipated supplies arriving on days 2, 4, and 5. All demands have a commit date but only two (B and E) have a request date. To keep the example simple, we assume all demands have the same priority. The key business question is what date can we deliver the product to the customer! Can we meet the commit dates? Can we meet any of the request dates?

In Fig. 14.27, we show a typical solution focused only on the commit date – all the commit dates but none of the request dates are met. The OTD score for the commit date is 0, but the OTD score for the request date is −5.

Demands for Part XYZ				Allocated Supply to meet Demand				on time delivery	
order id	commit date	request date	amount	supply ID	date	amount	date to customer	for commit	for request
A	2	none	15	SUP01	2	15	2	0	NA
B	3	2	5	SUP01	2	5	3	0	–1
C	4	none	10	SUP01	2	10	4	0	NA
D	5	none	20	SUP03	5	20	5	0	NA
E	6	2	10	SUP02	4	10	6	0	–4
total			60	total		60		0	–5

Fig. 14.27 Solution focused only on commit date

Demands for Part XYZ				Allocated Supply to meet Demand				on time delivery	
order id	commit date	request date	amount	supply ID	date	amount	date to customer	for commit	for request
A	2	none	15	SUP01	2	15	2	0	NA
B	3	2	5	SUP01	2	5	**2**	1	0
C	4	none	10	SUP01	2	10	4	0	NA
D	5	none	20	SUP03	5	20	5	0	NA
E	6	2	10	SUP02	4	10	6	0	–4
total			60	total		60		1	–4

Fig. 14.28 Solution obtained with a simple postprocessing routines

Demands for Part XYZ				Allocated Supply to meet Demand				on time delivery	
order id	commit date	request date	amount	supply ID	date	amount	date to customer	for commit	for request
A	2	none	15	SUP01	2	15	2	0	NA
B	3	2	5	SUP01	2	5	**2**	1	0
C	4	none	10	**SUP02**	4	10	4	0	NA
D	5	none	20	SUP03	5	20	5	0	NA
E	6	2	10	**SUP01**	2	10	2	4	0
total			60	total		60		5	0

Fig. 14.29 Solution in which both request dates are met

Sometimes, a simple postprocessing routine is executed to identify "easy" opportunities to meet request date. In our example, the supply used to meet demand B has an anticipated supply date of 2, therefore the customer can have the supply on day 2. The results of a postprocessing routine are shown in Fig. 14.28. With this routine, we are able to meet the request date for order B but not the request date for order E.

Figure 14.29 shows the results of carrying both commit date and request date within the solution process. Both request dates (demand B and E) are met. This is accomplished by swapping SUP01 and SUP02 between demand C and E.

The challenge is to carry both commit and request information (date, quantity, priority, etc.) intrinsically within the CPE and identify opportunities to come closer to customer requests (respecting their relative importance) without sacrificing any commitments which have been made. This requires carrying multiple dates and quantities throughout the explosion component of the CPE, identifying opportunities to assign assets to exploded demand to improve the posture of the exit demand, and being robust enough to handle such complexities as binning, substitution, and lot sizing.

14.4.8 Minimum Starts

Sometimes, a planner will want the CPE to ensure that a minimum number of starts occur at a certain point in the production process. For example, in Fig. 14.7, the planner may want a minimum of 100 Module_2 per week over the next 20 weeks to smooth the manufacturing flow or to meet contractual agreements (e.g., a vendor may be willing to promise a shorter cycle time if it has a "smoothed" start plan). In these cases, the CPE must explode the exit demand back to the identified minimum start manufacturing activity, compare the minimum start value with the required value, then continue the explosion process. On the return trip (implosion), the CPE must also adapt for the relative priority of the minimum start.

14.4.9 Date Effective Parameters, Substitution Rules, BOM, Capacity, Etc.

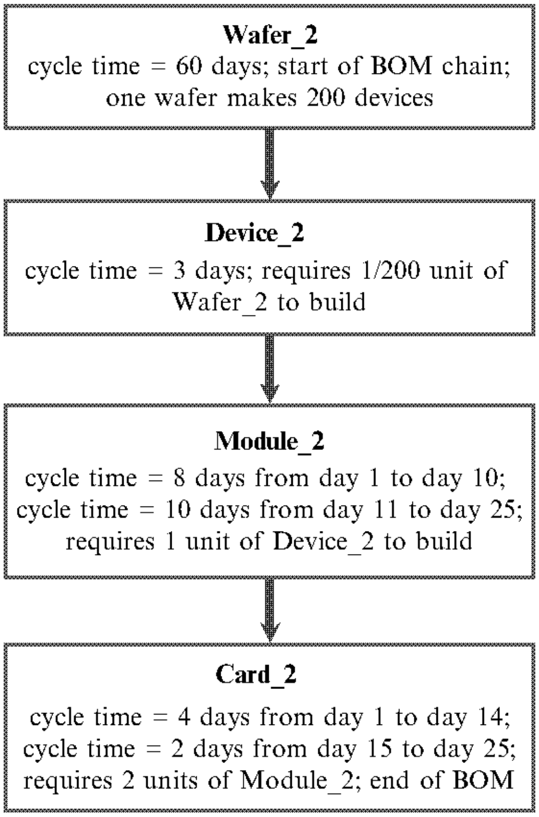
"Date effective parameters" are a challenge that is often underestimated and underserved – the solution offered often does not do an efficient job at meeting this challenge, leaving a firm with last minute patching and manual intervention.

Simply, this challenge is that most "descriptive" elements of the demand-supply network, such as yields, CTs, capacity available, capacity consumed, allowable substitutions, BOM, and so on, have a start date and an end date (date effective), and the CPE must recognize that these elements will change over time. For our discussion, we will use CT.

Previously, we used Fig. 14.7 as a simple production flow to explain explosion and implosion. In that example, the CT was fixed over time. Let us make the CT for Module_2 and Card_2 date effective (Fig. 14.30). The CT to produce Module_2 is 8 days in duration from day 1 to day 10 of the planning horizon and 10 days from day 11 to day 25. This means that the production of Module_2 started on day 5 has a cycle time of 8 days and will complete on day 13 (5 + 8). But if the production starts on day 12, the cycle time becomes 10 days and the completion time will be day 22 (12 + 10). The cycle time for Card_2 is 4 days from day 1 to day 14 and then reduces to 2 days from day 15 to day 25. Therefore, the search engine in both explosion and implosion must account for these changing cycle times.

Now let us revisit our implosion example. Manufacturing estimates 4 units of Device_2 will be available on day 10. If manufacturing immediately uses these 4 units to produce Module_2, the cycle time will be 8 days and on day 18 (10 + 8) 4 units of Module_2 will be completed. Continuing the projection process, the 4 units

Fig. 14.30 Flow for production of semiconductor parts with date effective cycle times


Wafer_2
cycle time = 60 days; start of BOM chain; one wafer makes 200 devices
Device_2
cycle time = 3 days; requires 1/200 unit of Wafer_2 to build
Module_2
cycle time = 8 days from day 1 to day 10; cycle time = 10 days from day 11 to day 25; requires 1 unit of Device_2 to build
Card_2
cycle time = 4 days from day 1 to day 14; cycle time = 2 days from day 15 to day 25; requires 2 units of Module_2; end of BOM

of Module_2 are immediately used to create 2 units of Card_2 beginning on day 18 (therefore the cycle time is 2 days), which will be available on day 20 (18 + 2).

What if capacity was not available on day 10 to start the production of Module_2? If capacity became available on day 11, the CT for Module_2 would increase from 8 days to 10 days. Therefore, a 1-day delay due to a capacity restriction would result in the completion of Card_2 being delayed 3 days to day 23 (11 + 10 + 2).

Let us turn our attention to explosion. To meet demand for 1 unit of Card_2 on day 20, the plant must have two completed units of Module_2 available on day 18 (20minustheCTforCard_2 on day18 = 20 – 2 = 18). This generates an exploded demand of 2 units of Module_2 with a need date or due date of day 18. When does the manufacturing facility need to start the production of 2 units of Module_2? When we had only one cycle time for the module, we simply subtracted that value (8) from the module's need date. This would drive a Module_2 start on day 10 (18 – 8). Life is not that simple with date effectivity. As a starting point, we first find the cycle time for Module_2 on day 18, which is 10 days. Remember 10 is the CT for the production of Module_2 that starts on day 18. But we do not want to start production on day 18; rather, we want production to complete (called to stock) on day 18. So we subtract 10 from 18 and get an initial starting date of 8. Does this work? We check the CT for the production of Module_2 on day 8 and find it is 8 days, meaning that if we start the production on day 8 the module will be completed on day 16, 2 days earlier than the time we need them. We might just stop here, since we have found a feasible solution. However, another demand may need a start on day 16. We extend our search to determine whether we can start later than day 8. Logically, we might try day 10. The CT for the production of Module_2 on day 10 is still 8 days, so the modules will be completed on day 18 (10 + 8). Just in time!

This is not the only complexity associated with date effectivity. Two other common ones are aggregating cycle times and converting to time buckets. In CTs, we might lump the production of Module_2 and Card_2 into one production activity (call it Mod_Crd_2). Before introducing date effectivity, this was simple – just add the two CTs and it becomes the cycle time for the new activity (8 + 4 = 12). Now we need to adjust for cycle time changes. In this case, the cycle time for Mod_Crd_2 is 12 days from day 1 to day 6, 10 days from day 7 to day 10, and 12 days again from day 11 to day 25. In time buckets, we need to convert the cycle time from daily information into number of buckets. For example, if the time bucket was 3 days in duration, we have 8+ buckets (25/3), and we need to restate each cycle time from days to buckets. Some of the buckets are located on a date effective boundary.

14.4.10 Other Technical Challenges

Other challenges include, but are not limited to, demand perishability, squaring sets, soft capacity constraints, alternative capacity, preemptive versus weighted priorities, splitting demand to match partial delays in supply, stability, express lots, delay assembly to test, dispatch lots, foundry contracts (Fig. 14.31: service model; multiple exit demands along the same BOM supply chain), risk-based inventory policy, multisupplier sourcing using inventory, WIP projection, rules governing purchase order change recommendations (allow for date change, quantity change, both or neither).

14.4.11 Challenge of Uncertainty: The Stochastic Nature of the Demand-Supply Network

Uncertainty (in parameters, estimated supplies, projected demands, etc.) is no doubt another critical challenge, but we will only briefly touch on this topic in this section.

In the near term, uncertainty may force planning applications to take a conservative approach to risk. For example, if (a) the average binning percentage for fast parts is 30%, (b) it has a uniform variability of plus/minus 10%, and (c) there are 100 parts currently being tested, then the actual number of fast parts that we will get from this specific manufacturing activity ranges from 20 to 40. Although on average we will get 30 fast parts from the sorting operation, 50% of the time the actual number of fast parts will be less than 30. Working with the average value is fine over a moderate or long timeframe, but it is impractical if your timeframe is only 1 day. One proposed method called cycle variation looks at taking a more conservative

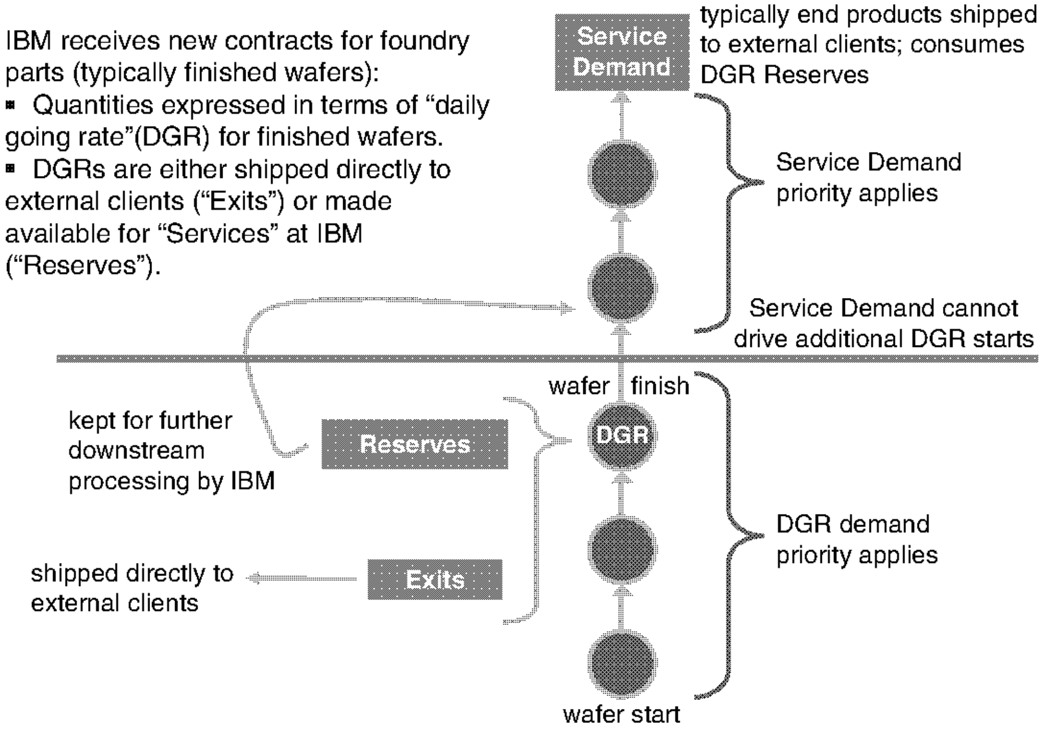

IBM receives new contracts for foundry parts (typically finished wafers):
▪ Quantities expressed in terms of "daily going rate"(DGR) for finished wafers.
▪ DGRs are either shipped directly to external clients ("Exits") or made available for "Services" at IBM ("Reserves").
Service Demand
typically end products shipped to external clients; consumes DGR Reserves
Service Demand priority applies
Service Demand cannot drive additional DGR starts
wafer finish
kept for further downstream processing by IBM
Reserves
DGR
shipped directly to external clients
Exits
DGR demand priority applies
wafer start

Fig. 14.31 Service and DGR demands for foundry contracts

approach with high priority demands than with low priority ones. The downside of this method is that "lower priority demands" had a higher probability of the CPE determining they could be met on time – a small problem!

For long range planning, executives would prefer to understand the range of possible outcomes and their likelihood, instead of being given a point estimate. There is a range of work going on (e.g., Kempf 2004) with some involving stochastic optimization while others pulling from DS of wafer fabricators (Burda et al. 2007) and wafer fabrication capacity planning (Zisgen 2005) and still others exploring inventory policy (IBM 2005).

14.4.12 Human Challenge: The Pull Between Simple and Complex Models

Little (1992) identified the human challenge as follows.

"Since, as humans, we have finite intellectual capacity or "bounded rationality" (Simon), we tend to break complex systems down into small, manageable pieces for analysis, design and control. Once we have decomposed a system into parts, we then have a desire to resynthesize small entities into big ones and work with the large entities as new units.

"Such hierarchical modeling is a useful approach, but not without pitfalls. Forrester (1961) points out that the parts of the system sometimes interact in unexpected ways and offers system dynamics as an approach for treating this."

14.5 Core of the CPE: Dynamically Making the Best Use of LP and Advanced Heuristics

This section describes the following core aspects of the CPE solution mechanism, with a focus on its capability to dynamically mix and match linear (mathematical) programming (LP) and advanced heuristic resource allocation technology. These core elements are:

1. IMEX (implode–explode) heuristic and its ability to dynamically invoke the Binning LP model to handle simple binning with downgrade substitutions (but not BSA, that is, complex binning, general substitution, and alternative BOM)
2. Supply Chain Optimization Planning Engine (SCOPE), an advanced LP-based engine, which handles the BSA, lot sizing, and preemptive priorities
3. Multidimensional partitioning strategy
4. Structures and algorithms to dynamically mix and match both resource allocation decision technologies (IMEX and SCOPE) to balance scope and scale

The focus of this section will be on the two LP formulations, the IMEX heuristic's ability to dynamically invoke the Binning LP, and the divide-and-conquer strategy, which involves a partitioning process to enable the mixing and matching of IMEX and SCOPE. It is outside the scope of this chapter to describe IMEX in detail, the partition algorithm, fair share, lot sizing, customer request, and date effectivity.

It is well known that without the right data structures, all attempts to build successful solutions are doomed to failure. Although we will not spend much time describing the data structures, they are the foundation of IBM's success. There are about ten key input tables that are tightly coordinated but independent. These tables provide such key information as build options, BOM, yields, cycle times, substitutions, binning, sourcing, receipts, inventory, etc. They also provide a solid base for the solutions to handle the few-to-many relationships and date effectivity with ease (data driven). Figure 14.32 has a list of the input tables and Fig. 14.33 the output tables.

14.5.1 IMEX: Implode/Explode BCD Heuristic

IMEX executes three major steps (Fig. 14.34). Step 1 is an MRP type of explosion of demand across the entire BOM supply chain, where hints are established and the binning points are optimized with the "Binning LP" (BINLP, see Figure set 7

Master Data

- CPE Alternate Source Parts
- **Binning**
- **Build Options**
- **BOM**
- CPE Bottom Level Parts
- Build to Forecast Percentage
- Process Cost
- Part Type
- New Plan Time Periods
- Period Used for Cap Calculations
- New Plan Effective Date
- **Part Number Data**
- Revenue Table
- Shutdown Calendar
- Sourcing
- **Substitution**
- **Cumulative Yield/Cycle Time**

Parameter & Rules

- Cap Priorities
- Macro Cap Reqmts Profile for Use at Bottom of BOM
- Cap Reqmts Profile
- Cycle Time Variability, and Loc Mult
- Degrade Option 1, and 2
- Override Option
- Forward Allocation rules
- Frozen Zone Parameter
- Inventory Policy, Qualifier
- Lot Size Rules
- Penalty Backorder
- Penalty Inventory
- Reorder Point
- Split Indicator
- Stability Push
- Starts Override Info
- Substitution Rules, by Customers, Parents
- Priority Option
- User Set

Transactions

- Mfg Cap Avail
- Balance of Mfg Cap Avail at WC
- Mfg Cap Avail WC
- **Demands**
- **Inventory Data**
- **Receipts**
- Scheduled Demand
- Scheduled Part Number
- **Shipping**
- Starts

CPE Partitioning Rules

- Card Partition SCOPE Extension
- Card Partition Description
- Device/Substrate Partition
- Module Partition LP Extension
- Module Partition

Fig. 14.32 Input tables for the CPE

Supply Side

- Byproducts
- Cap Detail
- Cap Detail by Demand Priority
- Cap Summarized
- **Manufacturing Releases**
- Manufacturing Releases by Demand Priority
- Planned Stocks
- Planned Substitutions
- Planned Supplier Shipments
- Projected Capacity Available
- Projected WIP
- Purchase Order Priorities
- **Vendor Shipments**
- WIP Priorities

Demand Side

- **Customer Shipments**
- **Dependent Demand**
- Early Warning
- Substitution to Customers
- Variability Demand

Fig. 14.33 Output tables for the CPE

for an optimized use of downgrade substitutions at a binning point). Hints refer to required manufacturing starts, tentative assignment of WIP or inventory to support a specific start, and demand priority associated with each level of exploded demand (Dangat et al. 1999). The (optional) second step allows the files generated during the first step to be modified to influence the final creation of anticipated supply. For example, the wafer start file could get modified. The third step executes a BOM implosion to create a final BCD solution that meets planning requirements such as demand class, fair share, etc.

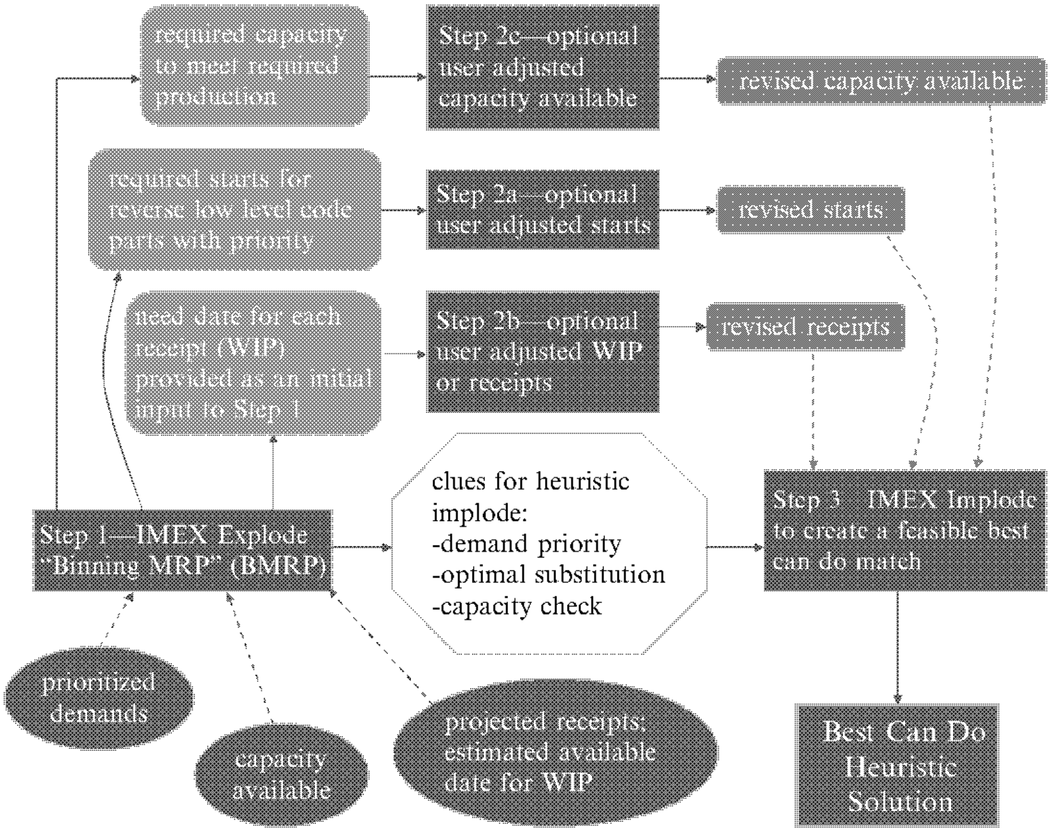

required capacity to meet required production
Step 2c—optional user adjusted capacity available
revised capacity available
required starts for reverse low level code parts with priority
Step 2a—optional user adjusted starts
revised starts
need date for each receipt (WIP) provided as an initial input to Step 1
Step 2b—optional user adjusted WIP or receipts
revised receipts
Step 1—IMEX Explode "Binning MRP" (BMRP)
clues for heuristic implode:
-demand priority
-optimal substitution
-capacity check
Step 3—IMEX Implode to create a feasible best can do match
prioritized demands
capacity available
projected receipts; estimated available date for WIP
Best Can Do Heuristic Solution

Fig. 14.34 Core flow for IMEX (Heuristic best-can-do)

14.5.1.1 Brief Explanation of Low Level Codes and Traditional MRP Explosion

To understand IMEX in detail, we must first review the classical MRP explosion and low level codes (LLCs). The reader already familiar with these topics can skip this section.

Figure 14.35 shows a simplified part flow from wafer to card. Common Wafer_1 can follow one of the three manufacturing processes to become Device_1, Device_2, or Device_3. The CTs for the manufacturing activities to create Device_1, Device_2 and Device_3 are 15, 10 and 12 days, respectively. Module_1 requires 1 unit of Device_1 and 1 unit of Device_2. Module_2 requires just 1 unit of Device_2. Module_3 requires 1 unit of Device_3. The cycle times required to complete Modules 1, 2 and 3 are 2, 3 and 2 days, respectively. Card_1 requires 1 unit of Module_1 and 1 unit of Module_2, and Card_2 requires 2 units of Module_2 and 1 unit of Module_3. The cycle times required to complete Cards 1 and 2 are 5 and 4 days, respectively. Independent customer (or finished goods) demand can exist for Card_1, Card_2, and Module_3 (marked with an asterisk in the figure). Module_3 is both a finished good and an intermediary part.

The first step in an explosion process is to determine the LLC for each manufacturing activity. Generally, activities that are only independent demand or finished

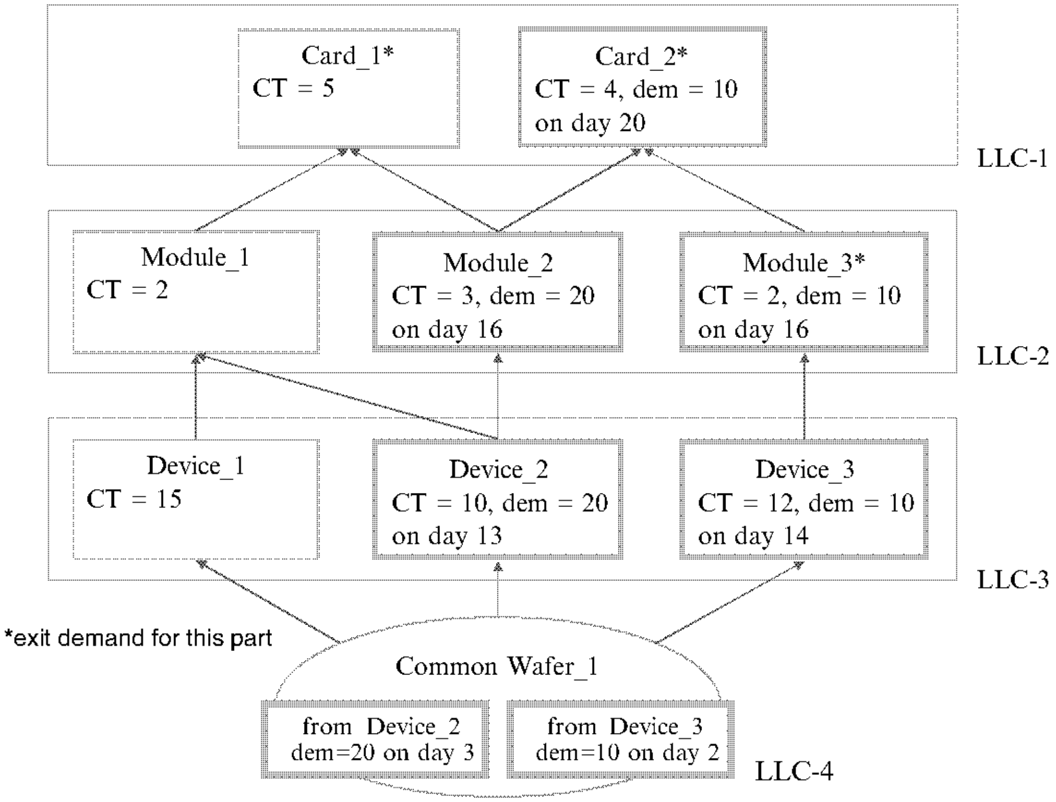


Fig. 14.35 Example of BOM explosion process

goods are put into the group of low level code 1 (LLC-1). In Fig. 14.35, parts Card_1 and Card_2 are in LLC-1. As to Module_3, although it has independent demand, it can also have exploded demand placed on it by Card_2. So Module_3 does not qualify to be in LLC-1. Next, LLC-2 contains all parts that have independent demand and/or dependent demand as a result of exploding demand on parts in LLC-1. In our example, parts Module_1, Module_2, and Module_3 are in LLC-2. Following the same logic, all devices are in LLC-3 and the Common Wafer_1 is in LLC-4.

The second step is exploding each demand backward through the BOM structure, one LLC at a time, to determine which parts in what quantity on what date have to be available to meet independent or dependent demand. In our example, this step begins by gathering and aggregating all of the demand on cards and then exploding the card demand to the module level. Turn to Fig. 14.35 again but this time, let us focus only on those orange boxes. Wetcountere see whether 10 units of Card_2 are required on day 20, then 20 (2 × 10) units of Module_2 and 10 units of Module_3 are required on day 16 (20 – 4). Next is to aggregate all exploded demands on modules with any independent demand and then explode this demand back to devices. To ensure 20 units of Module_2 are available on day 16, 20 units of Device_2 are required on day 13 (16 – 3), and to ensure 10 units of Module_3 are available on day 16, 10 units of Device_3 must be available on day 14 (16 – 2). This process continues until the BOM chain is exhausted. Continuing the example, to ensure 20 units of Device_2

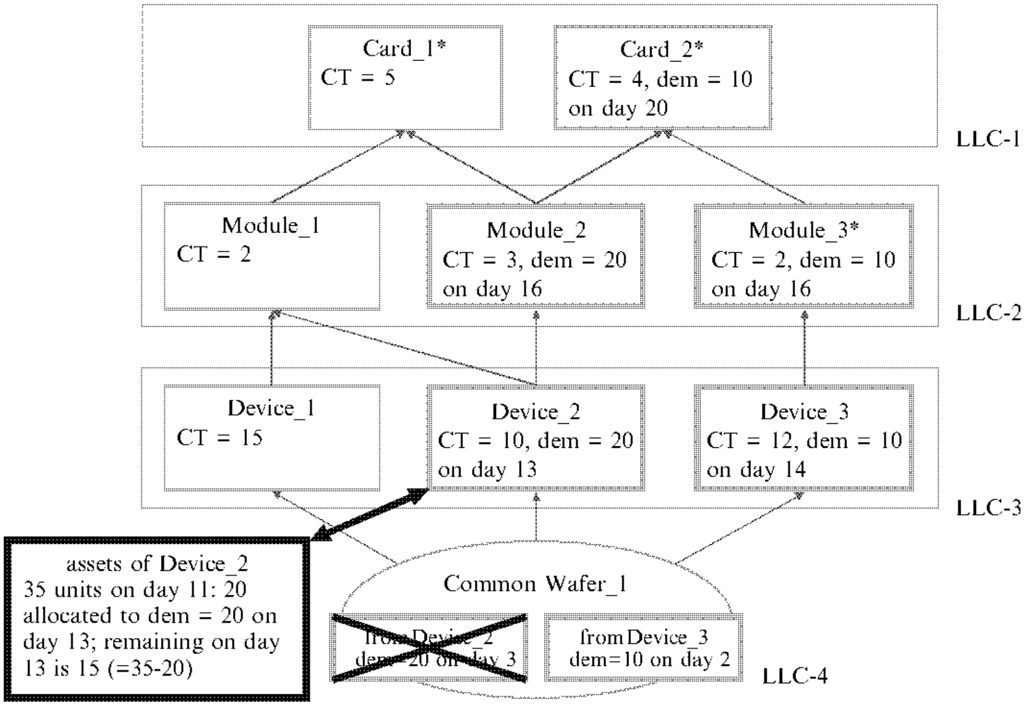


Fig. 14.36 Example of BOM explosion process, continued

are available on day 13, 20 units of Common Wafer_1 must be available on day 3 (13 − 10), and to ensure 10 units of Device_3 are available on day 14, 10 units of Common Wafer_1 must be available on day 2 (14 − 12).

Typically, during an explosion, the backwards engine will check whether existing inventory or projected WIP is available to meet the exploded demand. If it finds such assets in the pipeline, it truncates the explosion. Let's relate this to our example in Fig. 14.36. If 35 units of Device_2 are forecasted to come to stock on day 11 (the black box in the figure), these units will be allocated to meet the exploded demand created by Card_2 (20 units of Device_2 on day 13 to build Module_2, which is needed to build Card_2). As a result, no exploded demand will be sent to Common Wafer_1 to support the build of 20 units of Device_2.

14.5.1.2 IMEX Step 1: Explosion with Binning MRP

The first step of the IMEX is sending the demands to a special variation of MRP called "Binning MRP" (BMRP) to create exploded demand for each part on the BOM structure.

During the explosion step, the IMEX first examines all parts located at the top level of the BOM (i.e., those with LLC of one, or LLC-1) and separates the binned parts from the nonbinned ones. After the separation, the IMEX creates dependent demand on parts located at the lower levels (i.e., with a larger LLC) by exploding the nonbinned parts with the traditional MRP logic and the binned parts with a "binning LP" (BINLP) algorithm. The information considered in explosion includes, but is

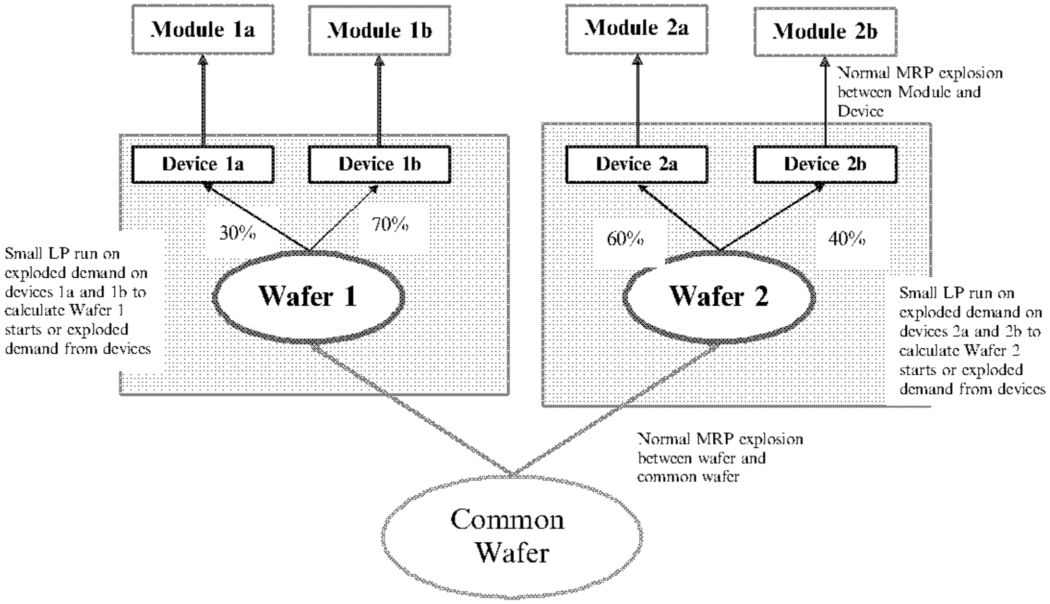

Module 1a
Module 1b
Module 2a
Module 2b
Normal MRP explosion between Module and Device
Device 1a
Device 1b
Device 2a
Device 2b
30%
70%
60%
40%
Small LP run on exploded demand on devices 1a and 1b to calculate Wafer 1 starts or exploded demand from devices
Wafer 1
Wafer 2
Small LP run on exploded demand on devices 2a and 2b to calculate Wafer 2 starts or exploded demand from devices
Normal MRP explosion between wafer and common wafer
Common Wafer

Fig. 14.37 Binning LP within MRP explosion for IMEX

not limited to, manufacturing starts, receipt due date, and required capacity. The same process repeats itself at every level in the BOM until reaching the lowest one (parts with the largest LLC). This completes the first major step of the IMEX heuristic. The key decision variable in the BINLP algorithm is the required starts in each time period for the binned parts. The objective function ensures that demand is met as much as possible (by imposing a huge penalty on every unit of unsatisfied demand) while lowering the period-end inventory at the same time. The constraints ensure a material balance between inventory, starts, receipts, and demand. Please refer to Appendix B for the detailed formulation.

The process flow for explosion is illustrated in Fig. 14.37. The BMRP first uses the traditional MRP logic to explode the demand for Module_1a, Module_1b, Module_2a, and Module_2b to create dependent demand for Device_1a, Device_1b, Device_2a, and Device_2b, respectively. Next, the BMRP explodes those (dependent) demands to create dependent demand for Wafer_1 and Wafer_2. Two small LPs are dynamically created (instantiated) for that purpose: one includes Device_1a, Device_1b, and Wafer_1, and the other includes Device_2a, Device_2b, and Wafer_2. The BMRP would solve both LPs to obtain exploded demand for Wafer_1 and Wafer_2. After that, the BMRP again uses the traditional MRP logic to calculate dependent demand for the Common Wafer. Note that the LPs are generally solved very fast in production runs, often in just a few seconds.

In addition to invoking LP runs at simple binning points, please also note these two features exhibited during the explosion process of the IMEX:

- Initial capacity checking, which means capacity is checked for new starts. And if capacity is exceeded, the starts will be moved to a different time and/or split based on available capacity.

- Demand data propagation, which means information such as demand class and lot-size quantity is carried along throughout the explosion process.

  Other than dependent demand, the IMEX explosion also creates the below information, which is critically important to the heuristic's implosion step.
- Required starts associated with such data as part ID, demand class, start quantity, start date, components, etc. (this includes those that have been moved forward or backward in time as a result of capacity checking).
- Required receipts or units of WIP (each with a need date).
- Required capacity for meeting all demand.
- Optimal substitutions as suggested by the BINLP algorithm.
- Reverse low level codes (RLLCs).

The required starts help the subsequent implosion step in identifying opportunities for earlier starts (in the microelectronics industry, starts often, but not always, mean wafers). Note that only one demand class is associated with each start. RLLCs establish the order in which parts will be processed during implosion – it is essentially the reverse of that followed by the traditional MRP logic. For purpose of further discussion, we define a PN to have an RLLC-1 if it does not have any components.

14.5.1.3 IMEX Step 2: Option to Modify Three Inputs for the Implosion

In this step, three types of output created by the explosion can be modified if necessary: required starts, projected receipts, and required capacity. For example, starts created in the past (i.e., those with a negative time) are brought forward to the current time, or a separate model that handles "cascading capacity" (Bermon and Hood 1999; Zisgen 2005) can be invoked to perform a detailed capacity analysis for the requested starts.

14.5.1.4 IMEX Step 3: Implosion

The implosion process begins by adjusting the starts located at the top of the reverse BOM (i.e., have an RLLC-1) to be time and capacity feasible. The starts are sorted by demand class, start date, and PN. Further, those with a start date in the past are changed to have a user-specified date or the first day in the planning horizon as the new start date. The stock date (i.e., when manufacturing is expected to complete and parts become available) is also adjusted accordingly. These revised starts are then examined to determine whether capacity is sufficient. If capacity is not enough, it will be allocated on a first-come-first-serve basis. Sorted by demand class and date, starts with a more important demand class have preemptive priorities over those with a less important demand class. If a start date has to be adjusted to accommodate capacity issues, the IMEX heuristic would attempt to move that start earlier in time, or delay it if moving earlier is not possible. When this is all done, the output is an adjusted starts file that is both capacity and time feasible for use in the subsequent implosion steps.

Next, parts are imploded beginning from those with an RLLC-1. At each reverse LLC, a proper processing order is identified such that substituting PNs (i.e., they substitute for others in short supply) are processed before those being substituted, and shipping locations (i.e., they send out shipments) are processed before those receiving shipments. Supplies such as starts, inventories, purchases, or WIP are collected. Then, the IMEX uses the demands generated by the BMRP run to determine supply allocation. Demands with the same demand class are covered first-come-first-serve. When supply is short, the more important demands may preempt supplies, i.e., taking away supplies that have already been allocated to the less important demands.

For lot-sizing, it will be preserved whenever possible during the implosion step, but will be relaxed when supplies are short to better meet customer demand. For example, if a lot of 25 wafers is needed but only nineteen are available on the need date, those nineteen wafers will be taken to continue the implosion process without delay (as opposed to waiting until later in time, six more wafers become available to make up a full lot).

14.5.1.5 IMEX Summary

The IMEX heuristic produces a projected supply schedule and commit date estimates satisfying a variety of constraints (temporal, asset-based, and business policy). When capacity is insufficient, starts with a more important demand class are given preference. Similarly, when supplies for a particular component are short, they would be allocated based on the demand class of competing parts.

The IMEX uses a binning LP algorithm to handle much of the complexity found in semiconductor-based manufacturing. Generally speaking, it cannot find high-quality solutions when encountering BSA: complex binning, general substitution, or alternative BOM paths (Sect. 14.4.3, Appendix A). For IBM CPE, these complexities are specifically handled by another LP-based solution called SCOPE (Sect. 14.5.2).

14.5.2 Supply Chain Optimization Planning Engine

SCOPE is an LP-based supply chain solution developed and deployed by IBM in the 1990s that has been continuously enhanced since. This solution is primarily used to handle the "triple crown of complexity" (i.e., the BSA described earlier sections) for which the IMEX and other heuristic-based solutions are in general unable to identify optimal solutions. The SCOPE solves a supply chain LP model with a cost minimization formulation. So a minimum cost solution is obtained when the model is solved to optimality. As defined in Appendix C, the SCOPE considers such costs as backorder, processing, inventory holding, material substitution, part shipment (ship to customers or ship to other manufacturing locations within the enterprise), and a few others. Whenever a feasible solution is identified, these costs will be multiplied by the associated decision variables in the objective function to obtain the (total) cost for the corresponding supply chain plan. In addition to costs, the SCOPE requires the usual input such as customer demand, scheduled vendor shipment, yield, capacity, cycle time, and so on. The input data structure is very much the same as that for the IMEX. Also as in the IMEX, most input data can be provided as time effective that the value would change over time to reflect the real situation.

The decision variables in the SCOPE LP model are chosen to correspond to actual supply chain activities: customer shipment (F_{makqj} in Appendix C), backorder (B_{mkqj}), substitution (L_{amnj}), manufacturing start (P_{maej}), interplant shipment (T_{mavj}), inventory level (I_{maj}), and sourcing (S_{auzj} and G_{auzj}). In any feasible solution, these variables must hold values to satisfy the following five types of constraints:

- Material balance equations maintain a flow balance between the creation and consumption (arrival and departure) of any PN at any stocking point, in any period, and at any manufacturing location. These equations handle all the complexities associated with binning, general substitution, and alternative BOM structures.
- Backorder conservation constraints keep track of unsatisfied demand throughout the planning horizon. That is, in any period they capture the total unsatisfied quantity contributed by all demands with the same PN, customer location, and demand class so it can be backordered and met in a future period.
- Capacity control equations safeguard capacity utilization, ensuring no resource is overutilized by starts requiring the same resource.
- Sourcing constraints enable the SCOPE users to control the deviation from sourcing targets, each of which is specified by two numbers, $MAXPCT_{auzj}$ and $MINPCT_{auzj}$. As defined in Appendix C, they indicate the maximum and minimum percentage of all shipments destined to consumption location(s) u that are preferred to originate from a particular supply location a. If the actual percentage falls above or below the targeted range, penalties will be incurred for the excess or shortage amount and added to the total supply chain cost.
- Nonnegativity constraints require all decision variables to be greater than or equal to zero.

14.5.2.1 Planning Periods with the SCOPE

Like other LP-based supply chain solutions, the planning horizon for the SCOPE is divided into a number of periods (or buckets), which is quite different from the IMEX heuristic. This characteristic is seen clearly in Appendix C, where, for example, the manufacturing start variable P_{maej} is defined for every PN m, plant location a, manufacturing process e, and period j. Furthermore, the number of (consecutive) days covered by one period may differ from that covered by other periods. It is obvious that for the same planning horizon, say, 2 years, more periods will increase the model size, which means it will take longer for the SCOPE to complete its solution process. On the contrary, if there are not enough periods, or if the period lengths are not set up properly, the SCOPE may not generate supply chain plans with sufficient granularity to meet business requirements. Therefore, it is crucial that the right period configuration (number–length combination) be identified.

Different period configurations are generally required for different planning purposes. For an enterprise strategic SCM plan, the planning horizon may span 2 years comprising 28 daily periods, 21 four-day periods, 10 weekly periods, 6 monthly periods, and 1 yearly period. In contrast, the planning horizon for an operational SCM plan may be 1 year, consisting of 30 two-day periods, 8 weekly periods, 2 monthly periods, and 1 six-month period. It takes experience and lots of experiments and fine-tunings to identify the right period configuration. Planning purposes, memory size, CPU speed, and the time window allowed for a supply chain plan to be generated are all factors in the determination of the right period configuration.

14.5.2.2 Cost Setup for the SCOPE

SCOPE creates supply chain plans with the lowest cost. Therefore, how costs are set up in the model has a direct impact to what solutions are found. Backorder costs (BOC_{mkqj}) are probably the most important, which are incurred whenever all or part of a demand cannot be satisfied by the commit date. Normally, a number of customer "tiers" are defined for planning purposes, representing how important a group of customers is to the enterprise relative to the others. Each customer is strategically assigned to a tier, which is associated with a carefully chosen backorder cost. Normally, any enterprise would try to meet the more important demands, that is, those received from the more important customer tiers, on time as much as possible. As such, the backorder cost for those customer tiers would need to be set as much higher than those for the less important customer tiers. Setting up the right backorder costs may be a challenge when the number of tiers is in the dozens, as there is a safe range for the maximum and minimum backorder costs to fall in without risking a numeric runtime error, while the cost difference between some neighboring tiers, e.g., tier one and tier two, also needs to be significant enough in order to meet planning goals.

14.5.2.3 Solution Process

The SCOPE begins its solution process by checking, filtering, and processing a variety of input data. In particular, relationships are established between dates and periods in terms of what the dates are for each period, and they serve as the base for transforming raw input data (which is provided in days) to such forms as needed by the subsequent LP processing steps. Data with the same attributes is further aggregated. One such example is customer demand that it is aggregated over the same PN, customer location, demand class, and planning period. Data aggregation reduces the size of the model, therefore reducing the time needed by the LP solver. New data such as the relationship between manufacturing input (i.e., starts) and output (i.e., stocks) is also obtained from the BOM, build options, yields, and part attributes. When all data processing steps are finished, various "TAB" files are generated with data displayed in a tabular format ready to be used for model generation.

The MPS format is used to represent an LP in the SCOPE solution process. Compared to the input data processing, model generation is quite straightforward that data in the TAB files is written to become one of the following in the MPS file: a decision variable's subscript, a constraint coefficient, or an objective function coefficient. The data in the TAB files is generated in such a way that it only requires a little more processing for model generation. The SCOPE then calls a subroutine library to solve the LP model.

The postprocessing step follows the return of an optimal solution by the optimization solver. A mirrored step of the input data processing, this step processes the LP solution so it can be used by supply chain planners. A comprehensive set of reports are produced directly from the LP solution, including customer shipments, manufacturing release schedules, dependent demand, purchase orders, and vendor shipment schedules. The LP solution can also be disaggregated and linked with input data to produce reports with more granularity.

14.5.2.4 Implementation

The SCOPE is implemented in $C++$ and shell scripts, and can dynamically invoke different optimization solvers including the open-sourced COIN-OR (COmputational INfrastructure for Operations Research, http://www.coin-or.org/) and the IBM Optimization Subroutine Library (OSL, developed at IBM T.J. Watson Research Center). The LPs generated for the module portion of IBM semiconductor manufacturing typically have two to three million variables and one to two million rows.

14.5.3 "Divide & Conquer": Decomposing the Problem (CPE Partitioning)

Despite the constant improvements in hardware performance, determining an optimal central plan for a large enterprise with just the SCOPE (or LP-based solutions in general) along is not even remotely close to being realistic. If the IMEX heuristic is used by itself, the quality of the decisions would most likely be less than desirable for complex product flows. Besides performance and ability to handle complex product flows, there are business reasons to partition and sequence the solution flow. For example, executing an explosion step identifies a prioritized set of required starts/releases needed to satisfy all demand in time. This is essentially a prioritized "wish list." This information would not be available with a single, monolithic solution process, since the only starts ever calculated are feasible with respect to capacity and component availability. The information of prioritized starts is particularly helpful when a planner is attempting to determine what actions to take to meet certain demands that are currently behind schedule or to establish minimum starts. Therefore, developing a fully automated, dynamically partitioning-and-sequencing algorithm which makes the best use of both heuristics and LP technologies was the logical course of action.

There are four fundamentals to the "divide-and-conquer" solution strategy:

1. The BOM or production flow can be divided into stages that occur naturally and every manufacturing activity is assigned to one stage.
2. Every part can be classified as complex or simple.
3. There is an explicit explosion action set followed by an explicit implosion action set.
4. Humpty Dumpty can be put back together again.

The following sections will explain each of these in more detail.

14.5.3.1 Assigning a Part to a Stage in the Production of a Finished Good

Within the production flow of semiconductor-based manufacturing, historically there have been part groupings or manufacturing stages that emerge naturally from the organizational structure. At IBM, there are three major part groupings or stages: Card, Module, and Device/Wafer. The core manufacturing processes at these stages are different and are typically performed at different manufacturing facilities. Therefore, from manufacturing point of view, starts are occurring at each stage and the exact timing and conditions of these starts are handled with some special "local logic." In addition, a consolidation process often occurs at each stage.

To assign a part to a stage, typically, the part is first associated with a specific part type; then, that part type is assigned to a stage. We will use Fig. 14.38 to illustrate this process. First, PNs Module_1, Module_2, Module_3 and Module_5 would be assigned to part type "Exit Module;" then, part type Exit Module would be assigned to stage "Module." This linkage puts all four modules into the Module stage. Similar structures would be used for other PNs, resulting in all the devices and wafers assigned to the "Device/Wafer" stage, and all the modules and raw modules assigned to the Module stage. There are a variety of checks to ensure a logical consistency across stage assignments. For example, if a user places device Dev_1A1 into the Module stage but wafer Wafer_1A into the Device/Wafer stage, this would be flagged since parts that are linked by binning must be in the same stage. As a second example, assume a planner places Module_3, Raw_ Mod_3A, and Raw_Mod_3B into the Module stage but Raw_Mod_3C into the Device/Wafer stage. This would also be flagged since all the component parts for Module_3 must be in the same stage. Figure 14.41 shows the BOM structure in Fig. 14.38 grouped by stage.

The algorithms to navigate the hierarchy, traverse linkages among manufacturing activities (resource consumed, part consumed, etc.) make linking associations, and do the consistency checking or flagging are set of network flow "membership"

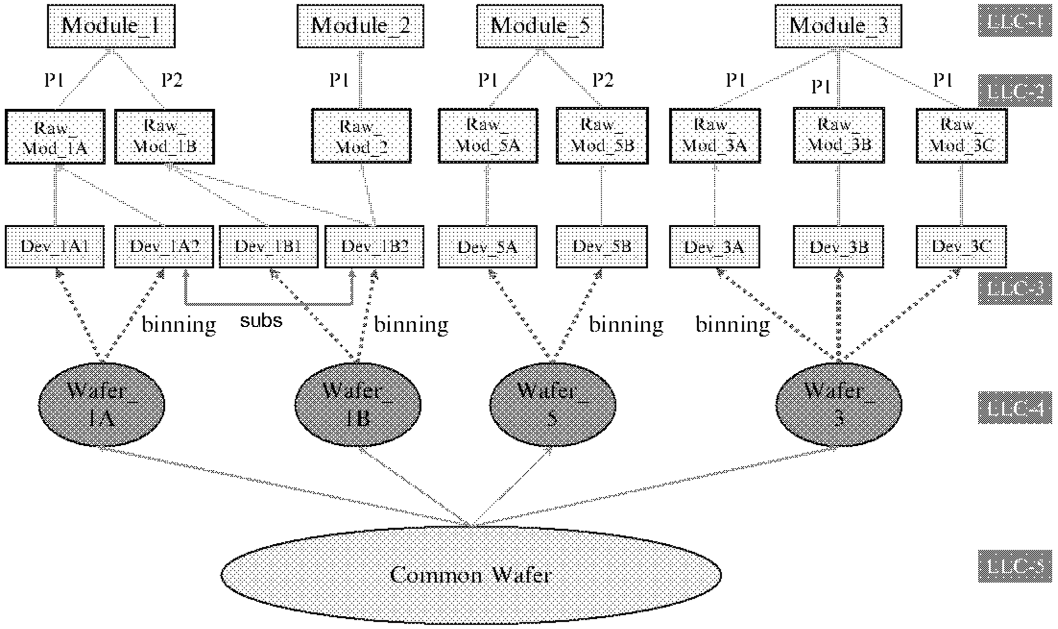


Fig. 14.38 Simple BOM structure from common wafer to module

algorithms (Sullivan et al. 1991) familiar these days to most computer science students. They were standard practice among the IBM team members who were "living history" (can remember programming before color and spreadsheets).

While the Card-Module-Device/Wafer stage decomposition is appropriate for IBM (and other firms that produce similar products), different decompositions may be necessary for different manufacturers. In general, the CPE assumes no specific ways to decompose the supply chain problem.

14.5.3.2 Assigning a Part as Simple or Complex

The second dimension of the decomposition is based on "complexity." The goal of this classification is to assign each part to be processed by either the SCOPE (i.e., a complex part) or the IMEX (a simple part). A part is typically classified as a complex part if one of the following conditions holds:

1. The part can be produced via alternative manufacturing processes at a given location.
2. The part can be produced via alternative general substitutions, and these substitutions are viewed by the business as the equivalent of alternative manufacturing processes at a given location (Figure set 8).
3. The part is involved in a complex binning situation.
4. The user of the CPE prefers the SCOPE to solve the part for some reason.
5. The part is connected to a complex part through product flows or resource sharing.

The first three conditions are determined directly by the CPE through examining the core input files. The fourth one is a planner decision. For example, if the alternative sourcing possibilities are particularly complex or resource availability is tight, the planner may want the solution to be created by the SCOPE. Any parts that are classified as complex via conditions one through four are called "primary complex parts." In Fig. 14.38, Module_1 and Module_5 are primary complex parts as both can be made using two different processes P1 and P2. Dev_1A2 and Dev_1B2 are also primary complex as there is a general substitution relationship between them that Dev_1A2 may substitute for Dev_1B2 in any condition and vice versa.

A network membership algorithm is used to classify parts as complex via the above fifth condition. Starting from the primary complex parts, this algorithm will conduct an up-and-down search along the BOM supply chain. Any part that is reached by this search is labeled "secondary complex" (it is essentially "guilt by association"). We will first use Fig. 14.20 to illustrate primary and secondary complex parts. Module_9 is a primary complex part as it can be built in three methods: method one uses process P1 and consumes Device_8A; method two uses process P2 and consumes Device_8B; method three still uses process P2 but consumes Device_8C (as a substitution for Device_8B). All other parts in the figure, i.e., Module_8 and untested device, are secondary complex. As another example, Module_1 in Fig. 14.38 is a primary complex part (it can be produced by process P1 or P2), so all of the parts below it (upstream in the BOM) are classified as secondary complex. Figure 14.39 shows all of the parts in Fig. 14.38 classified as simple and complex.

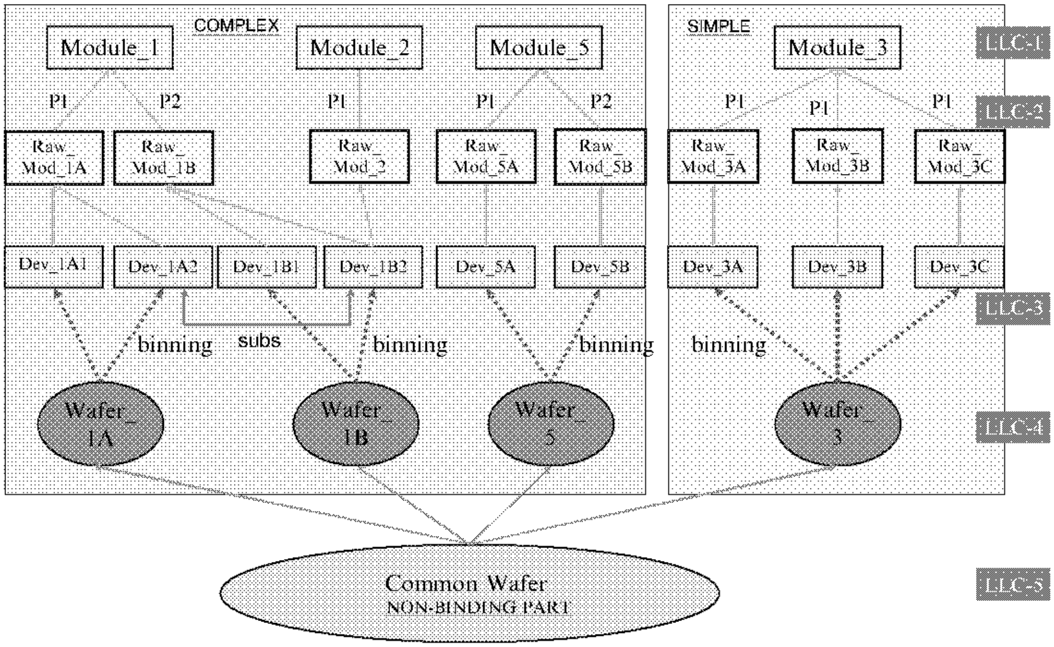


Fig. 14.39 BOM structure with parts grouped by simple and complex

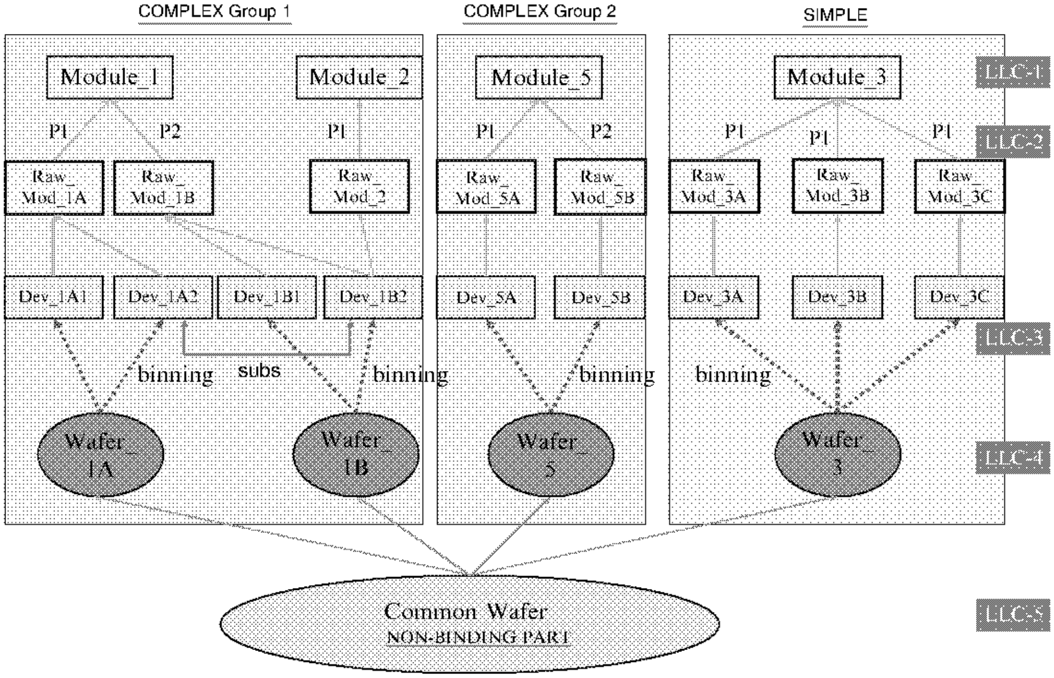


Fig. 14.40 BOM structure showing two independent complex groups of parts

Note that the part called Common Wafer is not classified as complex; rather, it is called a "nonbinding part." A part is classified as nonbinding if the supply is sufficient to meet all demand at any time – that is there is never a need to allocate its supply between competing demands. It is called nonbinding, since it cannot be used to "connect" one part of the BOM structure with another in terms of determining whether a part is complex or not – or in terms of independent paths. In this example, the part Common Wafer is nonbinding. The CPE will assume any exploded demand for this part is met immediately. Without the "nonbinding" part, then all parts would be complex parts and all parts would belong to one single group. Although a simple concept, it is critical to the success of classify parts a complex or simple and being able to identify two mutually independent complex groups of parts shown in Fig. 14.40.

It is important to note that the search for secondary complex parts may or may not cross stage boundaries. Figure 14.41 shows how the BOM structure is grouped by stage: LLCs 1 and 2 belong to the Module stage and LLCs 3, 4, and 5 belong to the Device/Wafer stage (for simplicity, the Card stage is not shown). In complex group 1 identified in Fig. 14.40, the complex status of the parts flows from the Module stage into the Device/Wafer stage. Therefore, parts in this group are complex in both stages. This is, however, not the case for complex group 2: the parts are complex in the Module stage but simple in the Device/Wafer stage.

Decisions on shared capacity may also drive parts into the complex group, even if they are not connected to primary complex parts through a BOM linkage.

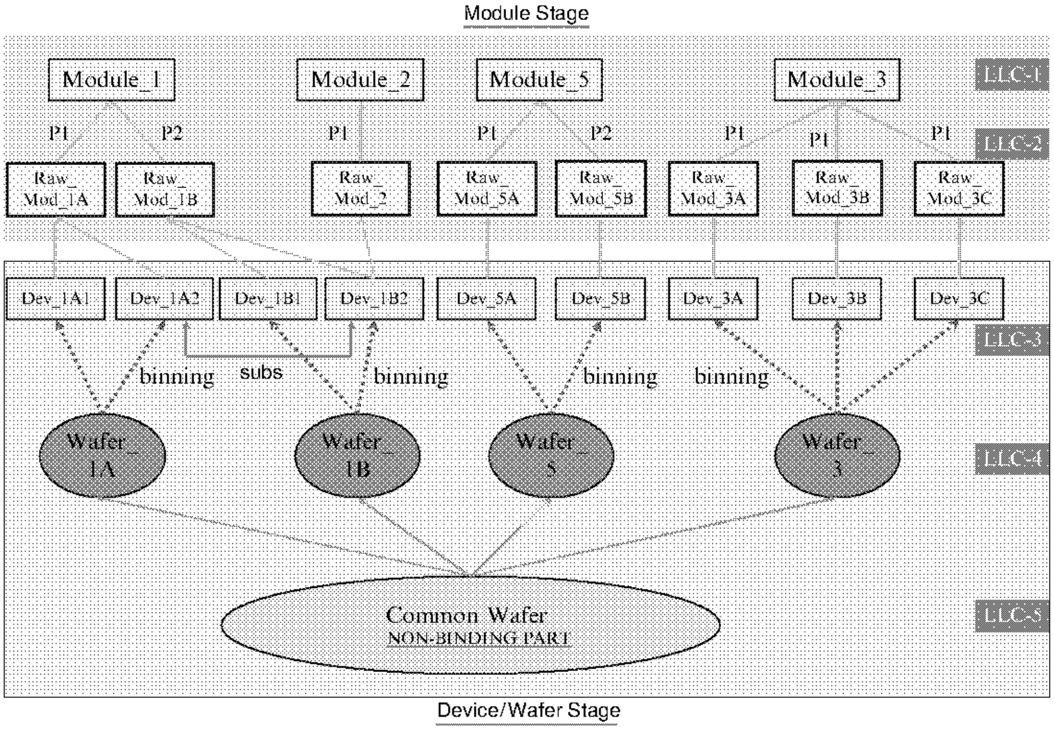


Fig. 14.41 BOM structure with the module and device/wafer stages

14.5.3.3 Decomposing the Problem across Two Dimensions: How Do We Get a Solution?

The methods described in Sects. 14.5.3.1 and 14.5.3.2 form the partitioning algorithm shown in Fig. 14.42. Apply it and we can decompose a semiconductor & package supply chain problem into six subproblems or "partitions" (in the case of IBM): Card-Complex, Card-Simple, Module-Complex, Module-Simple, Device-Complex, and Device-Simple (Fig. 14.43). Note these partitions are obtained across two dimensions, and the Complex ones will be solved by the SCOPE and the Simple ones will be solved the IMEX heuristic.

So now the question becomes: how do we connect all the pieces in the right sequence to create an enterprise-wide, detailed supply chain plan? The key is to mimic the old manual pattern that existed at IBM until the mid 1990s (and is still being done today at many firms). We will adapt and enhance this pattern with the use of a partitioning algorithm (Fig. 14.42), the IMEX heuristic (Sect. 14.5.1), and the SCOPE (Sect. 14.5.2).

At a high level, the algorithm for the CPE consists of four major steps: preprocessing, explosion, implosion, and postprocessing (Fig. 14.44). Preprocessing divides the original SCM problem into six partitions, which allows the entire problem to be solved in a controllable, divide-and-conquer manner (one partition at a time). The next step carries out a traditional, MRP-type explosion of the BOM to determine capacity, material, and other requirements. This step is executed in the

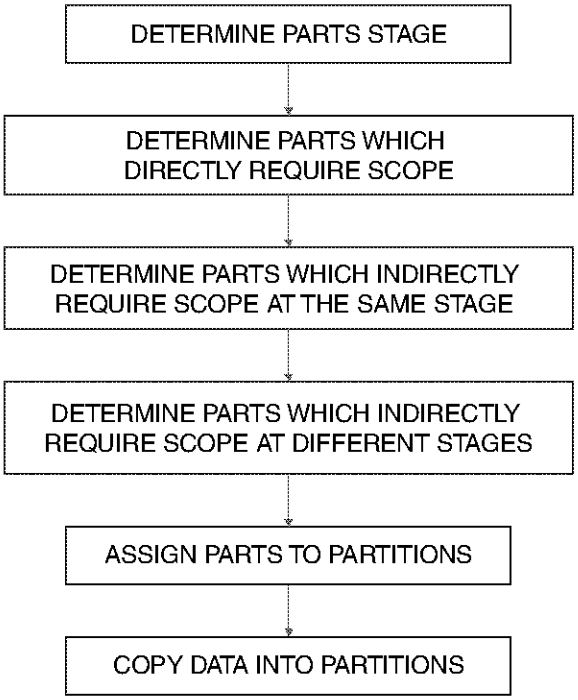

DETERMINE PARTS STAGE
DETERMINE PARTS WHICH DIRECTLY REQUIRE SCOPE
DETERMINE PARTS WHICH INDIRECTLY REQUIRE SCOPE AT THE SAME STAGE
DETERMINE PARTS WHICH INDIRECTLY REQUIRE SCOPE AT DIFFERENT STAGES
ASSIGN PARTS TO PARTITIONS
COPY DATA INTO PARTITIONS

Fig. 14.42 Steps in establishing partitions

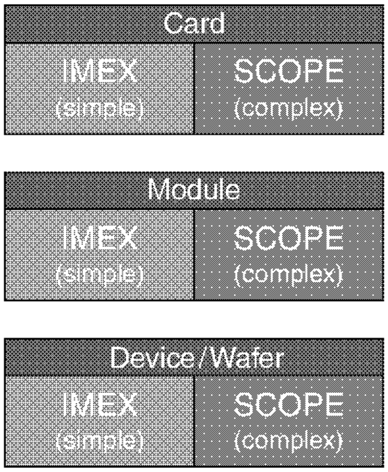


Fig. 14.43 Partitions of the CPE

sequence of the Card stage, the Module stage, and the Device/Wafer stage, and both the IMEX and the SCOPE are involved in the explosion within each stage in a co-operative manner. A method called "one level explosion" is executed to connect the explosion of one stage to the next, which will be explained in more detail later. After the completion of the BOM explosion, an implosion step begins to match supplies with the requirements created by the explosion. In essence, this is a mirror step of the previous one: it is executed in the sequence of the Device/Wafer stage, the Module stage, and the Card stage; within each stage, it uses both the IMEX and the SCOPE to consider capacity availability and consumption and create a feasible

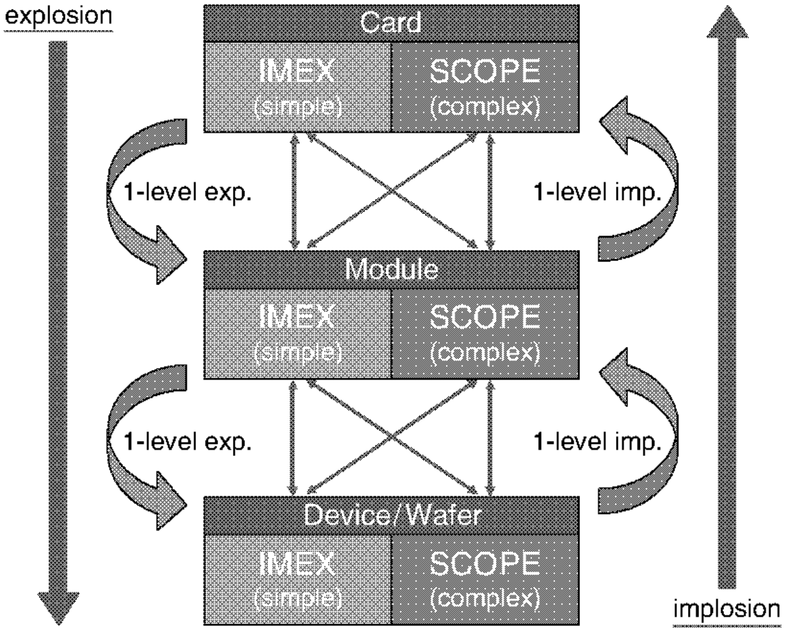


Fig. 14.44 Algorithmic flow of the CPE

supply chain plan; and a method called "one level implosion" is executed to connect the implosion of one stage to the next. Finally, the postprocessing step consolidates outputs from all the partitions to create a single, coherent solution, which is further processed through a series of formatting procedures to produce usable reports for the supply chain planners.

Starting Point for the Solution: Explosion at Cards

The CPE begins the explosion for the Card stage by gathering all the pertinent information: BOM, substitutions, inventory, WIP, yields, etc. It then divides this pool of information into two categories: complex (i.e., the Card-Complex partition) and simple (i.e., the Card-Simple partition). For the Card-Complex partition, the SCOPE is used in an MRP instantiation to determine the exploded demand for the Module stage (remember that MRP is concerned with determining the minimum starts needed on the latest possible date to meet demand without any concern for capacity). For the Card-Simple partition, the IMEX which contains the BINLP algorithm (Sect. 14.5.1, Appendix B) is used. Details of each individual demand are maintained, and exploded demands are consolidated into a single picture. It turns out that this is also a good time to handle some aspects of lot sizing and a few other planning requirements. This process is repeated two more times, moving backwards by stages from Card to Module and then to Device/Wafer.

One Level Explosion Algorithm Between Stages

One level explosion is a critical component of the CPE, as decomposing the manufacturing process into stages relies on this algorithm running efficiently to link two consecutive stages. One level explosion passes crucial information such as demand type and demand class from one stage to the next. For instance, when the CPE has finished the explosion of, say, the Card stage, it needs to know how many units of what (dependent) demand to create at the Module stage to support the starts recommended for the Card stage (in turn, these starts would support exit demand).

The type of demand created during the solution process is called dependent demand. To allow one level explosion to create dependent demand accurately, an association (or "pegging") between customer shipments and manufacturing starts is required, where these starts are made for the parts located at the bottom of the BOM for the current stage. This association must have such granularity that a correct portion of each start is pegged to a particular customer shipment, and thus to a particular customer demand (because customer shipment contains the original demand information). Such information is made available by the pegging method.

Continuing the Explosion Process

After the one level explosion between cards and modules is complete and pegged dependent demand for the modules is established, the explosion process is repeated at the Module stage creating device demand. The same process is repeated for a third time at the Device/Wafer stage creating wafer demand. These wafer demands are the bottom of the food chain.

Starting Implosion to Create a Supply Statement

After the completion of the (last) explosion process for the Device/Wafer stage, the CPE reverses the stage sequence and starts an implosion process for each of the stages. Especially, the implosion step considers the requirements generated during the explosion step and solves the problem within each partition with limited capacity. For the SCOPE solver, typically the capacity allocated to each time period is the maximum of three amounts:

1. The prorated share of the total capacity required for the given time period (determined during the explosion step).
2. The prorated share of the total capacity required for the given time period, the time period before the given one, and the time period after the given one.
3. The prorated share of the total capacity required throughout the planning horizon.

The use of the maximum ensures the capacity available to the SCOPE is at least equal to the prorated share of the complex parts during the current, intermediate, and long-term time periods. There are a number of alternatives that are more complex and can account for demand priority; these alternatives may also involve an iterative exchange between the two solvers. But generally, the above maximum rules work reasonably well in practice.

The establishment of capacity availability allows the SCOPE to solve the complex partition and create a feasible, intelligent solution. The exact capacity consumed by the complex parts can be calculated, and it is subtracted from the available capacity. Then, the IMEX is invoked to solve the simple partition at the same stage, and again, the available capacity is adjusted by subtracting the amount consumed by the simple parts. The remaining capacity will be used later by appropriate parts belonged to a subsequent stage. The implosion step also needs pegging and other necessary methods to link two consecutive stages (e.g., from Module to Card), which is achieved by passing information such as demand class on to the next stage and executing some required (nonlinear) business rules.

First Implosion Step

The first stage to be considered for implosion is Device/Wafer. Parts are again classified into complex and simple and then solved by the SCOPE and the IMEX, respectively. The goal of the implosion step is to best meet prioritized demand created during the previous step without violating temporal or capacity constraints. The algorithm used to allocate capacity between the complex and the simple partition has been described in the previous section, which can be tweaked by the planner if necessary. It turns out estimating capacity availability/consumption is a quantum more elusive than knowing such things as product flows, inventory, and WIP. These sharing algorithms work well in practice.

One Level Implosion Algorithm between the Stages

One level implosion is another critical component of the CPE. It runs between two consecutive stage implosion processes, e.g., after the completion of implosion for the Device/Wafer stage and before the start for the Module stage. This algorithm gathers up supplies generated during the implosion process which just finished, merges them with inventory, WIP and other existing supplies, and allocates these supplies to the next implosion process. One level implosion handles all of the "tracking" issues that associate supply with demand priority, some aspects of lot sizing, and capacity allocation as appropriate.

In practice, there is a considerable amount of intelligence required to optimally allocate supply to demand at this point of the solution process. The implosion step of the CPE mixes and matches a simple, but powerful and exceptionally fast heuristic (not the IMEX, but one specifically designed for this task) and a scaled-down version of the SCOPE to perform this task. In addition, there is a tremendous opportunity for large grain parallelization – for large problems that require the solution power of the SCOPE; dynamic parallelization is a key to robust performance.

This process continues until the last stage is completed.

Putting Humpty Dumpty Back Together Again

Once the implosion for the Card stage is completed, all individual solution pieces need to be put together into a single, consolidated solution. The reconfiguration of the supply chain into stages or partitions is only for the purpose of identifying an intelligent plan – apparently, users of the CPE do not need to know which partition a part was in while the CPE was running, so such information needs to be invisible to the users. The consolidated solution provides the planners with a cohesive, detailed plan or schedule of events in the enterprise with a clockwork precision, allowing them to view the plan from an aggregate level or monitor detailed lot movements and capacity consumption. Again, the pegging/tagging information is used by the appropriate algorithms to achieve this result. The output tables created by the CPE contain early warning reports, coverage analysis, detailed demand pegging (individual lot or customer order level of detail), detailed manufacturing starts, capacity consumption, substitutions, etc. All this information is available to the planners in their "tool of choice." Additional details about the core algorithms in this CPE can be found in Hedge et al. (2004), Milne et al. (1999), Orzell et al. (2004), Denton and Milne (2006), Denton et al. (2004), and Denton et al. (2005).

Beyond the Holy Grail

As the reader has noticed, the goal to create and deploy a centralized, end-to-end enterprise wide supply chain plan with a sufficient level of detail and a reasonable runtime performance has been the "holy grail" since the early 1990s. Today, a substantial portion of that goal has been achieved.

As with any science, the accomplishment of one goal brings not only a sense of pride but also a huge dose of reality in what is left to be finished. In SCM, achieving a reasonably strong level of central control can dramatically increase organizational performance – but it also clearly identifies gaps in a timely synchronized response that currently can only be handled with ad hoc manual interventions that operate without a global awareness. A simple example is an order change once the plan has been established; another example is a component part that can be finished earlier than the planned cycle time to meet customer demand but there is no connection between the two activities.

In both of these examples, the fundamental question is when to "update the plan." In the current standard approach, the plan is regenerated based on some amount of elapsed time. As Harpal Singh (2007) observes, this is often not best practice:

> "We are finding the need of a monitoring mechanism that measures the changes in supply and demand so that the "current" plan can be tested constantly for relevancy. So the issue becomes – when should the plan be re-run? The traditional approach is for a fixed, time based cycle like a week or a month. We are finding that this is not necessarily a good approach because the frequency is in direct proportion to the volatility in the environment."

The flip side of a gap is an opportunity window and we will look at these opportunities in the next section.

14.6 Future Direction 1: Extending the Big Bang

Over the past 10 years, much of the effort in enterprise-wide central planning has been to move organizations from a decentralized, loosely coupled approach that created a new plan once a month to a reasonably tightly integrated process using more intelligent models to create new plans from once a week to once a day. There is still considerable work to do across business organizations in an enterprise to achieve this level of central planning. Even for firms who have achieved this level, effort is still required to keep it from slipping back into bad habits.

We refer to today's SCM central planning process as the "big bang" approach. An enterprise creates a centralized process and data representation of the firm at some point in time. Then, through some combination of automated models and manual processes, the enterprise creates a "global plan." Some firms execute a global planning process a few times each day; others do it once a month and take a week to create the plan. In both cases, the new plan arrives after a reasonable amount of effort and replaces the old plan in its entirety. A new "universe" is created via the "big bang!"

There are two primary opportunities/challenges for this big bang approach: (1) expanding the use of advanced global planning from leading firms to all firms, and (2) improving the quality of the global plan by capturing more of the complexity in the supply chain solver. The first challenge belongs to business consultants and operations management researchers. The key question to ask for this challenge is this: what are the obstacles at average firms that keep them from bringing the same level of sophistication and supply chain performance as observed at leading firms? Our observation on this topic is that most business consultants and operations management researchers focus mostly on processes and data and fail to understand the importance of the decision models. The second challenge includes reducing solution times with parallel processing and faster algorithms and machines, improving accuracy with smaller time buckets or units of granularity, with more robust representations of the demand-supply network, or with latest techniques such as constraint programming, handling the stochastic nature of information, and automating some aspects of plan review.

14.7 Future Direction 2: Beyond the Big Bang: Supply Chain Physics of the Twentieth Century Meets Its Quantum Revolution

At the start of the twentieth century, physicists learned that we do not live in a clockwork universe (Wolfson 2000). The same can be said for supply chain modelers at the start of the twenty-first century. Although centralized processes and models have increased organizational effectiveness, there are clear limits and we are rapidly reaching them.

Effective centralization refers to the ability to take into consideration all aspects of the decision situation simultaneously and generate an optimal or at least a very good solution. To be effective, a central solution requires a synchronized current view of the entire decision landscape, the ability to handle complex trade-offs, and a reasonably fast runtime performance. Gaps exist and are often created by time lags, summarization, performance, triggers, and formulation. By triggers, we refer to the event that "wakes up" at a specific time during the day and runs the central solution. Once a week, once a day, or once every 3 days, the central solver executes, but the decision to execute is made without any knowledge or monitoring of events since the last execution. Formulation gap refers to the inability to formulate key decision questions in such a manner that lends itself to a central solution as opposed to a sequence of negotiations or collaborations. The reality is, even if we could get the big bang models to finish execution in 5 min, an organization would run it once a day at best to match demand with supply and synchronize the enterprise. As the time between runs decreases (for example, from once a week to once a day), the following limitations become apparent:

1. Understanding and repairing the plan.
    - 1.1 Why does the plan give these results?
    - 1.2 What demands are not being met (alerts) and why?
    - 1.3 What actions can I take to improve the plan?
        - 1.3.1 Identify and book actions to improve the supply posture as it relates to demand.
        - 1.3.2 Less significant items such as inventory picture.
    - 1.4 Monitoring the repair actions as they are being executed.
2. The nature (quality) of the plan – no plan to plan continuity.
    - 2.1 No checks, filters, or alerts (CFAs) on demand information/signals.
    - 2.2 No CFAs on supply information/signals.
    - 2.3 No CFAs on changes in production specification or business policy.
    - 2.4 Each plan is built from scratch.
    - 2.5 No built-in dialog with other key providers of input, such as
        - 2.5.1 Projected supply.
        - 2.5.2 Projected demand.
        - 2.5.3 Capacity estimation.
3. Observe both 1 and 2 require an incremental matching (planning) or net change engine.

The following example shows how a "simple" repair action can quickly become complex. In Table 14.5, we have two demands for the same part (P111): D001 with a due date of 4/4/2006, priority of 2, and quantity of 80 and D002 with a due date of 4/5/2006, priority of 1, and quantity of 100.

In Table 14.6, we have two anticipated supplies for part P111: S00A and S00B with an anticipated delivery date of 4/4/2006 and 4/6/2006; the quantity is 100 units each.

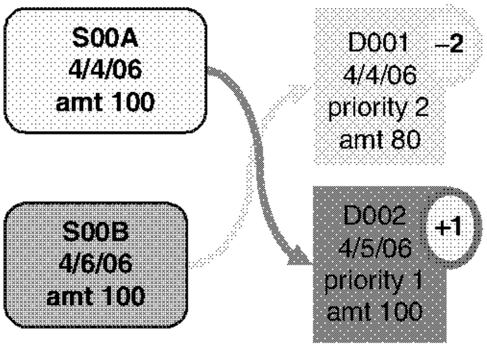
Table 14.5 Demands for Part P111

Demand for Part P111			
ID	Date	Priority	Quantity
D001	04/04/06	2	80
D002	04/05/06	1	100

Table 14.6 Supplies for Part P111

Supply for Part P111			
ID	Date	Priority	Quantity
S00A	04/04/06	na	100
S00B	04/06/06	na	100

Fig. 14.45 Initial assignment of supply to demand for Part P111



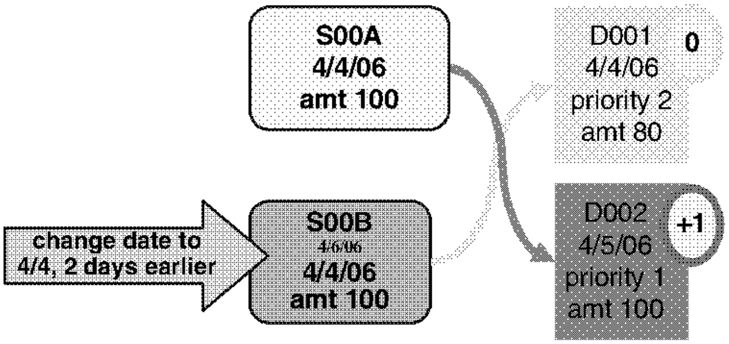


Fig. 14.46 Simple repair to meet demand D001 on time

Initially, we assign S00A to D002 (Fig. 14.45) since D002 has a higher priority and the other supply S00B would arrive 1 day after the due date for D002. This assignment resulted in D001 being met 2 days late.

Figure 14.46 demonstrates a simple repair action to meet demand D001 on time: the analyst requests that the S00B supply be completed 2 days earlier on 4/4/2006.

Figure 14.47 shows a smarter repair action. First, the analyst switches the assignment of supply to demand – supply S00A is assigned to D001 and supply S00B is assigned to D002. Then the analyst requests that supply S00B be expedited only 1 day to 4/5/2006 to meet D002 on time.

To take organizations to the next leaps in efficiency requires a substantive adjustment in approach. Just as nineteenth century physics had to adjust its equations and

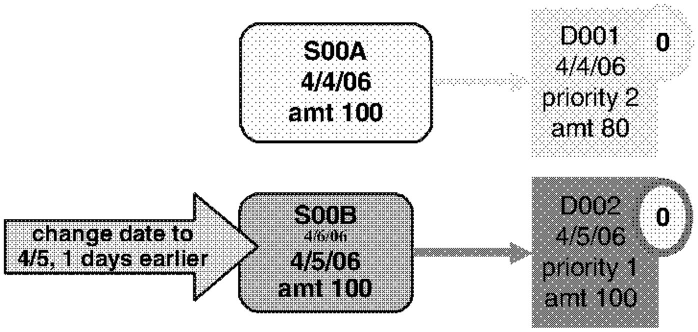


Fig. 14.47 Smarter repair action to meet both demands on time

formulations for special relativity, general relativity, and quantization, supply chain solvers must learn to accommodate techniques from intelligent agents and sense and respond and learn to incorporate the concept of collaboration into their solution processes. Collaboration refers to an iterative process that focuses on finding a satisfactory solution. The next search step depends on prior steps and may involve back tracking. Often, the step involves negotiating a temporary change in a subset of the rules governing the game for a limited period of time, and typically contingency occurs to handle uncertainty.

Two major areas in which the IBM supply chain team is currently working are understanding and repairing the plan and establishing plan-to-plan continuity with sense, response, and incremental matching (net change engine).

Typically when a planner reviews a plan, he or she would visit the following questions: what demands are not being met (alerts) and why; why the solver gave these results in the plan; what the options are to improve the plan; identify and book actions to improve the supply posture as it relates to demand or inventory; and monitoring that the repair actions are being executed. The IBM team is working on a tool called AIIRR (Assess, Identify, Improve, Respond, Repair).

Today, each plan is built from scratch, that is, without reference to the prior plans or changes in demand or supply occurred after the execution of the last plan. There are no checks, filters, or alerts (CFAs) on demand information, supply information, and changes in product specification or business policy. There is no built-in dialog with other key providers of input such as projected supply and projected demand. There is no ability to incrementally modify an existing plan. Currently, the IBM team is working on applications to close these gaps.

14.8 Conclusion: New Science Emerges and Extends the Borders of Bounded Rationality

When one of the authors, Ken Fordyce, joined IBM in 1977 as a junior programmer with an undergraduate degree in mathematics, a few courses in operations research and statistics, one of the first people he met was Herschel Smith. Herschel was in his 60s and Ken was 23. Herschel had built a small LP model to optimize taxes for IBM World Trade. His efforts were recognized and appreciated by IBM, but he was part of an Information Systems organization. At that time, modeling did not have enough traction to be its "own person," and decision modeling applied to business problems was at most 25 years old.

Today, IBM and some of its clients use a combination of LP and some pretty clever "heuristics" to establish a daily plan for their enterprise. These firms cannot think of "life without these decision models." This same level of success of decision models can be seen in a number of areas. We believe we are witnessing a growing awareness of the importance of decision models and already competing on analytics (Davenport 2006).

That said, there is still a long way to go. People are comfortable with their guesses and decision scientists often fail to deliver real value. Politely, life is much cleaner if your modeling work remains an academic exercise. What decision scientists offer is the potential to "Extend the Borders of Bounded Rationality."

Herbert Simon (Nobel Prize Winner in Economics) (1957) observed, "as humans, we have "bounded rationality" and break complex systems into small manageable pieces." The challenge for organizations is to integrate information and decision technology to push boundaries out and improve performance. Nick Donofrio (IBM Senior Vice President) observed, "access to computational capability will enable us to model things that would never have believed before." The challenge reaches beyond coding algorithms, linking with data, and turning it on. Each decision-science team must execute its role as "intelligent evolutionists" to ensure the organization adopts complex decision technology in a sustained incremental fashion.

This will enable organizations from semiconductor firms to hospitals to be more responsive.

> "The ability to simultaneously respond to customers' needs and emerging business opportunities in an intelligent orderly manner is a survival requirement for today's market place. Our customers continue to tell us that the quality of our responsiveness is as important as the quality of our products... The work done by the innovators on our PROFIT team, as well as their colleagues, has enabled us to meet the customer responsiveness challenge. And I'm convinced that the monetary savings associated with the PROFIT work [$80 million in 1999 alone] – though significant in their own right – understate the work's true value to IBM." (Nick Donofrio, IBM Senior Vice President in Lyon et al. (2001)

Appendix A: Example of Limitations of a Simple Heuristic to Handle Alternative BOM

Figure 14.48 contains a simple alternative flow example with demand and assets. The Red Module can be produced with process P1 or P2. The P1 path consumes Dev01 and the P2 path consumes Dev02. The Blue Module can be produced with two processes also, consuming Dev02 or Dev03. The on-hand inventories, demands,

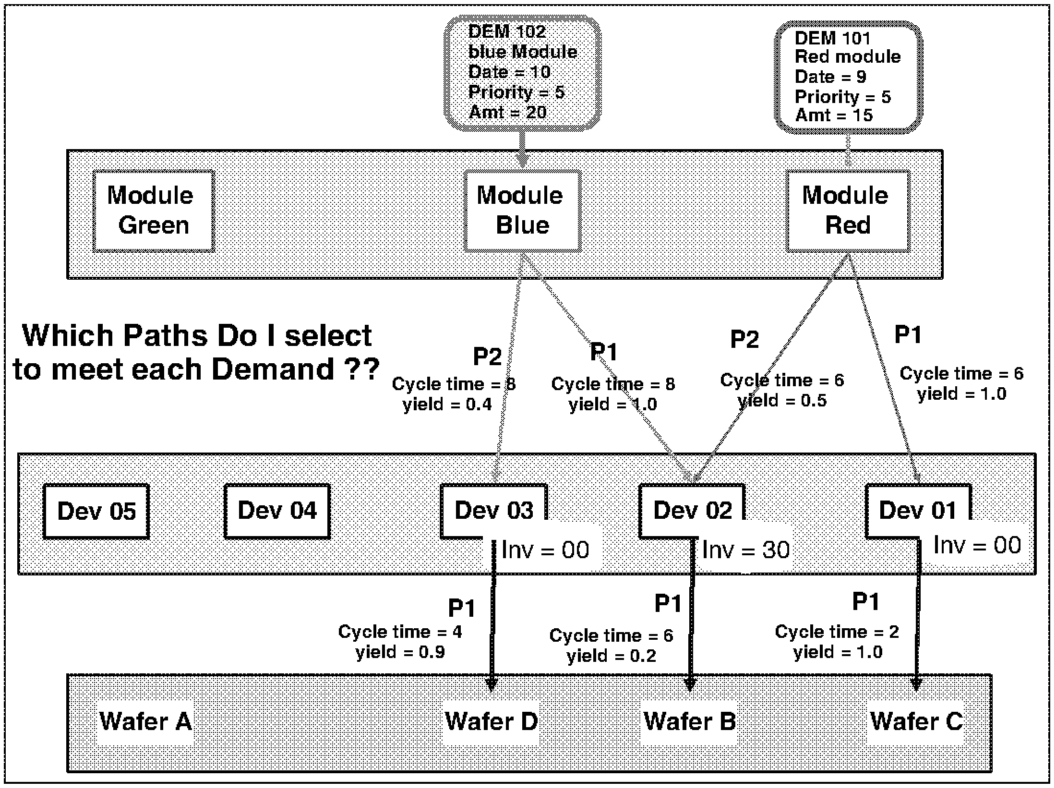


Fig. 14.48 Alternative paths to meet demands that share common assets

cycle times, and yields can be seen in the figure. The CPE must decide a series of activities to meet the demand for the Red and Blue Module.

One possible heuristic to handle alternative BOM might focus on consuming existing inventory first to "optimize" the use of existing assets. The heuristic would work as follows:

1. Process demand in the following order
   1.1 Start with the highest (most important) demand class.
   1.2 If there is a tie in demand class, start with the demand which has the earliest due date (within that demand class).
2. When alternative BOM paths exist
   2.1 Do a "one level explosion" down the first path (e.g., P1)
      2.1.1 If inventory exists to meet this demand, consume this inventory.
      2.1.2 If there is no or insufficient inventory, explore the next alternative path (e.g., P2).
   2.2 Repeat this process of looking to consume existing inventory through each alternative BOM.
   2.3 If there is no or insufficient inventory at each alternative BOM, return to the first (P1) process and apply the same heuristic rule to explode the next level

If we apply the above heuristic to our example in Fig. 14.48, we get the following solution.

1. Process demand Dem101 (Dem101 is processed first since it has the same priority as Dem102 and an earlier due date).
   1.1 Explode the P1 path first
      1.1.1 Need 15 units of Dev01 on day 3.
      1.1.2 Inventory of Dev01 is 0, so try the next path.
   1.2 Explode the P2 path
      1.2.1 Need 30 units of Dev02 on day 3.
      1.2.2 30 units of Dev02 are available, so assign them to Dem101.
2. Process Dem102
   2.1 Explode the P1 path.
      2.1.1 Need 20 units of Dev02 on day 2.
      2.1.2 The inventory of Dev02 is 0 (already allocated all to Dem101), so try the next path.
   2.2 Exploded the P2 path
      2.2.1 Need 50 units of Dev03 on day 2.
      2.2.2 But the inventory of Dev03 is 0.
   2.3 Return to the P1 path and explode to the next level
      2.3.1 Need 100 units of Wafer B on day -4.
      2.3.2 Generate a need of 100 units on Wafer B on day 0 (ignore lot sizing).
3. Results of this heuristic are
   3.1 Dem101 is met 3 days early.
   3.2 Dem102 is met 4 days late.
   3.3 remaining inventory is 0.
   3.4 wafer needs are 100 units of Wafer B.

With just a little bit of study, we can identify two alternative answers that better meet demand and result in a better inventory position. The first alternative solution is as follows:

1. For Dem102
   1.1 Explode the P2 path
      1.1.1 Need 56 (=(20/0.4 = 50)/0.9) units of Wafer D on day −2.
2. The results become
   2.1 Dem101 is met 3 days early.
   2.2 Dem102 is met 2 days late.
   2.3 Remaining inventory is 0.
   2.4 Wafer needs are 56 units of Wafer D.

The following is the second alternative solution.

1. For the 30 units of Dev02 in inventory
   1.1 Allocate 20 units to Dem102.
   1.2 Allocate the remaining 10 units of Dev02 to Dem101.
   1.3 Explode the P1 path for the 10 units of unsatisfied Dem101.
      1.3.1 Need 10 units of Wafer C on day.

2. The results are
   2.1 For Dem101
      2.1.1 Units are met 3 days early (from 10 units of Dev02 in inventory).
      2.1.2 10 units are met on time or potentially 1 day early.
   2.2 Dem102 is met 2 days early.
   2.3 remaining inventory is 0.
   2.4 wafer needs are 10 units of Wafer C.

Appendix B: Detailed Formulation for the Binning LP

Bellow is the LP formulation for simple binning and downgrade substitution to obtain the minimum production required for binned part at the next level.

LP Formulation for Simple Binning and Substitution

Definition of Constants

J Set of all output parts, resulting when the binned part is produced
K Set of all the parts that have demand, obviously we have $K \subseteq J$
T Number of time periods

Definition of Subscripts

j Output PN that results from the binning process, $j \in J$
k PN that has demand, $k \in K$
t Time period, $t = 1, 2, \ldots, T$

Definition of Coefficients

I_{j0} Inventory of output part j at the beginning of the planning horizon, i.e., $t = 0$.
D_{kt} Demand for part k in period t.
R_{jt} Receipts of output part j in period t.
B_{jt} Binning percentage of output part j in period t.

Definition of Decision Variables

P_t Production of the binned part in period t
S_{jkt} Quantity of output part j that is used to satisfy the demand of output part k during period t.
I_{jt} Inventory of output part j at the end of period t.
Z_{kt} Unsatisfied demand of part k during period t.

Below is the LP formulation to solve the simple binning and substitution production situation.

$$\text{Minimize} \sum_t \left[P_t + \sum_j \left[10^9 Z_{jt} + 0.0001 I_{jt} + \sum_k 0.00001 S_{jkt} \right] \right] (0.9)^t$$

Subject to

$$I_{jt} = I_{j(t-1)} + B_{jt} P_t + R_{jt} - \sum_k S_{jkt}$$

$$D_{kt} = Z_{kt} + \sum_j S_{jkt}$$

$$I_{jt} >= 0,\ P_t >= 0,\ Z_{jt} >= 0,\ S_{jkt} >= 0,\ \forall \text{ all } j,\ t,\ k.$$

The objective function computes the minimum production required for the binned part. A huge penalty is charged for every piece of unsatisfied demand; naturally, the LP will always try to meet all the demand on time. The first constraint calculates the inventory of the output part j at the end of period t: it is equal to the part's inventory at the end of the previous period $(t - 1)$ increased by the new stock from the binned part's production (i.e., $B_{jt} P_t$) and the part's receipts in period t, then decreased by the total substitution quantity. The second constraint specifies the relationship between the demand and the unsatisfied demand for part k during period t. Finally, the last constraint requires all the decision variables to be nonnegative. This LP formulation can be solved very quickly to optimality.

Appendix C: Detailed Formulation for the Complex LP (SCOPE)

LP Formulation for Complex Product Structures. Below is a detailed description of the basic LP model adopted for SCOPE.

Definition of Subscripts

j Time period/bucket
m Part number.
n Part being substituted.
z Group (representing a family of parts).
e Process (can be a manufacturing or purchase process).
a Plant location within the enterprise.
v Receiving plant location.

k Customer location (note that a customer location can never be a plant location).

q Demand class (indicates relative demand priority).

w Resource capacity (a resource can be a machine, worker, etc.).

u Consuming location(s) (which can be a plant within the enterprise or an external demand location).

Definition of Objective Function Coefficients

PRC_{maej} Cost of releasing one piece of part m during period j at plant a using process e.

$SUBC_{amnj}$ Substitution cost per piece of part n which is being substituted by part m during period j at plant a.

TC_{mavj} Transportation cost per piece of part m leaving plant a for plant v during period j.

$INVC_{maj}$ Inventory cost of holding one piece of part m at plant a at the end of period j.

$DMAXC_{auzj}$ Cost per piece of exceeding the maximum amount of shipments specified for parts in group z from plant a to consuming location u during period j

$DMINC_{auzj}$ Cost per piece of falling short of the minimum amount of shipments specified for parts in group z from plant a to consuming location u during period j.

BOC_{mkqj} Backorder cost of one piece of part m at the end of period j for demand class q at customer location k.

Definition of Constants

$DEMAND_{mkqj}$ Demand requested during period j for part m at customer location k for demand class q.

$RECEIPT_{maj}$ Quantity of projected WIP and purchase order receipts for part m expected to be received at plant a during period j.

$CAPACITY_{waj}$ Capacity of resource w available at plant a during period j to support manufacturing starts.

$CAPREQ_{wmaej}$ Capacity of resource w required for part m at plant a for process e during period j needed for manufacturing starts.

$QTYPER_{maenj}$ Quantity of component part m needed per part n during period j at plant a using process e.

$YIELD_{maej}$ Output of part m per piece released (or started) at plant a during period j using process e.

$SUBQTY_{amnj}$ Quantity of part m required to substitute for one piece of part n at plant a during period j.

$MAXPCT_{auzj}$ Maximum percentage of total shipments of group z (a collection of parts) leaving supply plant a during period j to support consumption at location(s) u.

$MINPCT_{auzj}$ Minimum percentage of total shipments of group z leaving supply plant a during period j to support consumption at location(s) u.

CT_{maej} Cycle time (which is the number of periods between the release and completion of parts) for releases of part m using process e at plant a during period j.

TT_{mav} Transport time for part m from plant a to plant v.

Definition of Decision Variables

I_{maj} Inventory at the end of period j for part m at plant a

P_{maej} Manufacturing starts of part m during period j at plant a using process e.

L_{amnj} Quantity of part n which is being substituted by part m during period j at plant a.

T_{mavj} Internal shipments of part m leaving plant a for plant v during period j.

F_{makqj} Shipments of part m leaving plant a during period j to satisfy class q demand at customer location k.

B_{mkqj} Backorders of part m at the end of period j for class q demand at customer location k.

H_{uzj} Total shipments of group z leaving supply locations during period j to support consumption at location(s) u.

S_{auzj} Amount by which total shipments of parts in group z from plant a to consumption location(s) u during period j exceeds the maximum amount specified as desired in the sourcing rules.

G_{auzj} Amount by which total shipments of parts in group z from plant a to consumption location(s) u during period j falls short of the minimum amount specified as desired in the sourcing rules.

LP Model Formulation

Minimize

$$\sum_m \sum_a \sum_e \sum_j PRC_{maej} P_{maej} + \sum_a \sum_m \sum_n \sum_j SUBC_{amnj} L_{amnj}$$
$$+ \sum_m \sum_a \sum_v \sum_j TC_{mavj} T_{mavj} + \sum_m \sum_a \sum_j INVC_{maj} I_{maj}$$
$$+ \sum_a \sum_u \sum_z \sum_j DMAXC_{auzj} S_{auzj} + \sum_a \sum_u \sum_z \sum_j DMINC_{auzj} G_{auzj}$$
$$+ \sum_m \sum_k \sum_q \sum_j BOC_{mkqj} B_{mkqj}$$

Subject to

(material balance constraints)

$$
\begin{aligned}
I_{maj} = {} & I_{ma(j-1)} + RECEIPT_{maj} + \sum_{x \ni x + CT_{maex} = j} \sum_{e} YIELD_{maex} P_{maex} \\
& + \sum_{n} L_{anmj} + \sum_{x \ni x + TT_{mva} = j} \sum_{v} T_{mvax} - \sum_{n} SUBQTY_{amnj} L_{amnj} \\
& - \sum_{v} T_{mavj} - \sum_{k} \sum_{q} F_{makqj} - \sum_{\substack{n \ni m \\ is\_component\_of\ n}} \sum_{e} QTYPER_{maenj} P_{naej}
\end{aligned}
$$

(backorder conservation constraints)

$$B_{mkqj} = B_{mkq(j-1)} + DEMAND_{mkqj} - \sum_{a} F_{makqj}$$

(capacity constraints)

$$\sum_{m} \sum_{e} CAPREQ_{wmaej} P_{maej} \le CAPACITY_{waj}$$

(sourcing constraints)

$$H_{uzj} = \sum_{m \in z} \sum_{a} \left(T_{mauj} + \sum_{q} F_{mauqj} \right)$$

$$\sum_{m \in z} \left(T_{mauj} + \sum_{q} F_{mauqj} \right) - S_{auzj} \le MAXPCT_{auzj} H_{uzj}$$

$$\sum_{m \in z} \left(T_{mauj} + \sum_{q} F_{mauqj} \right) + G_{auzj} \ge MINPCT_{auzj} H_{uzj}$$

(nonnegativity constraints)

$$\text{all decision variables } X_{i,j,\ldots} \ge 0.$$

The objective function measures a production plan's total cost (or penalty) over the planning horizon chosen by the planner. The total cost is contributed by activities of manufacturing (P_{maej}), material substitutions (L_{amnj}), interplant logistics (T_{mavj}), inventory holding (I_{maj}), sourcing (S_{auzj} and G_{auzj}), and backorders (B_{mkqj}) occurred in all the time buckets. It is a common practice to formulate SCM models based on a number of time buckets with variable durations or lengths. Then, these buckets collectively form the planning horizon for the production plan. Different combinations of time buckets are usually used for different planning purposes. For example, planning horizons for an enterprise strategic SCM plan may span 2 years and consist of 28 daily buckets, 21 four-day buckets, 10 weekly buckets, 6 monthly buckets, and 1 yearly bucket. Because of this modeling practice, part of the LP solution process is to aggregate appropriate demands and other items for every time bucket so that the correct quantities are used for model generation.

Among the five types of constraints, material balance constraints ensure the inventory balance of material flows at every stocking point in the BOM supply chain as well as through time buckets in the planning horizon. In simple words, they make sure that what goes in equals what comes out. Next, backorder conservation constraints calculate the backorder amount that is needed at the end of every time bucket j, which is the balance between the total shipment made in that bucket and the demand for the same bucket plus the backorder carried forward from bucket $j-1$. Any (positive) backorders are accumulated over time, and they disappear only when sufficient shipments can be made to cover them and any new demand. Next, capacity constraints safeguard the usage of any resource capacity so that none goes beyond what is available. Obviously, if capacity has an unlimited supply, there will be no need to backorder because manufacturing starts can just be made to satisfy any demand. The sourcing constraints adopted in our model are "soft" in the sense that the ideal sourcing levels $MAXPCT_{auzj}$ and $MINPCT_{auzj}$ can be violated by paying a price, with S_{auzj} and G_{auzj} being slack variables to capture the quantity of sourcing overage and underage and then be penalized in the objective function. Finally, nonnegativity constraints ensure that all the decision variables in the model remain either positive or zero.

References

Arntzen B, Brown G, Harrison T, Trafton L (1995) Global supply chain management at digital equipment corporation. Interfaces 25(1):69–93

Bermon S, Hood S (1999) Capacity optimization planning system (CAPS). Interfaces 29(5):31–50

Burda R, Degbotse A, Dews B, Milne RJ, Sullivan G (2007) Who would have thought – optimization in fabricator dispatch and artificial intelligence in enterprise wide planning. In: (Working paper) IBM Strategic Systems, 1000 River Road, Essex Junction, VT 05452, USA

Dangat GS, Gokhale AR, Li S, Milne RJ, Orzell RA, Reid RL, Tang X, Yen C (1999) Best can do matching of assets with demand in microelectronics manufacturing. U.S. Patent 5,971,585, 26 Oct 1999

Davenport T (2006) Competing on analytics. Harvard Business Review, January 2006, pp 1–9

Denton B, Milne RJ (2006) Method for optimizing material substitutions within a supply chain. U.S. Patent 6,983,190, 3 Jan 2006

Denton B, Hedge S, Orzell RA (2004) Method of calculating low level codes for considering capacities. U.S. Patent 6,584,370, 18 May 2004

Denton B, Forrest J, Milne RJ (2005) A method for considering hierarchical preemptive demand priorities in a supply chain optimization model. U.S. Patent Application: 2005–0171828, also, IBM docket: BUR9–2003–0198US1

Denton B, Forrest J, Milne RJ (2006) Methods for solving a mixed integer program for semiconductor supply chain optimization at IBM. Interfaces 36(5):386–399

Duchessi P (1987) The conceptual design for a knowledge based system as applied to the production planning process. In: Silverman B (ed) Expert systems for business, pp 163–194

Fogarty D, Hoffman T (1983) Production and inventory management. South-West Publishing, Cincinnati, OH

Fordyce K (1998) Matching assets with demand engines for PROFIT and supply chain management. MicroNews (a publication of the IBM Microelectronics Division, 3rd Quarter, 1998) 4(3). www.chips.ibm.com/micronews/vol4_no3/profit.html

Fordyce K (2001) New supply chain management applications provide better customer service: serious gets exciting. MicroNews (a publication of the IBM Microelectronics Division, 2nd Quarter, 2001) 6(3). http://www.chips.ibm.com/micronews/vol7_no2/fordyce.html

Forrester J (1961) Industrial dynamics. M.I.T. Press, Cambridge, MA

Galbraith J (1973) Designing complex organizations. Addison-Wesley, Reading, MA

Glover F, Jones G, Karney D, Klingman D, Mote J (1979) An integrated production, distribution, and inventory planning system. Interfaces 9(5):21–35

Goldman S (2004) Science in the twentieth century. Great Courses on CD by the Teaching Company, Chantilly, VA

Graves RJ, Konopka JM, Milne RJ (1995) Literature review of material flow control mechanisms. Prod Plan Contr 6(5):395–403

Hackman ST, Leachman RC (1989) A general framework for modeling production. Manag Sci 35(4):478–495

Hegde SR, Milne RJ, Orzell RA, Pati MC, Patil SP (2004) Decomposition system and method for solving a large-scale semiconductor production planning problem. United States Patent No. 6,701,201 B2

IBM white paper G510–6402–00 (2005) DIOS – dynamic inventory optimization, IBM Corporation, 1133 Westchester Avenue, White Plains, NY 10604, USA

IBM white paper G299–0906–00 (2006) Collaboration with IBM E&TS Helps ADI Stay ahead of Customer Demand, IBM Corporation, 1133 Westchester Avenue, White Plains, NY 10604, USA

Kempf K (1994) Intelligently scheduling wafer fabrication. In: Intelligent scheduling. Morgan Kaufmann, San Francisco, CA, pp 517–544 (Chapter 18)

Kempf K (2004) Control-oriented approaches to supply chain management in semiconductor manufacturing. In: Proceedings of the 2004 American control conference, Boston, MA, pp 4563–4576

Leachman R, Benson R, Liu C, Raar D (1996) IMPReSS: an automated production planning and delivery-quotation system at Harris corporation – semiconductor sector. Interfaces 26(1):6–37

Lee HL, Padmanabhan V, Whang S (1997) Information distortion in a supply chain: the bullwhip effect. Manag Sci 43(4) (special issue on frontier research in manufacturing and logistics):546–558

Lin G, Ettl M, Buckley S, Yao D, Naccarato B, Allan R, Kim K, Koenig L (2000) Extended enterprise supply chain management at IBM personal systems group and other divisions. Interfaces 30(1):7–25

Little J (1992) Tautologies, models and theories: can we find “laws” of manufacturing? IIE Trans 24(3):7–13

Lyon P, Milne RJ, Orzell R, Rice R (2001) Matching assets with demand in supply-chain management at IBM microelectronics. Interfaces 31(1):108–124

Milne RJ, Orzell RA, Yen C (1999) Advanced material requirements planning in microelectronics manufacturing. U.S. Patent 5,943,484, 28 Aug 1999

Norden P (1993) Quantitative techniques in strategic alignment. IBM Syst J 32(1):180–197

Orlicky J (1975) Material requirements planning: the new way of life in production and inventory management. McGraw-Hill, New York

Orzell R, Patil S, Wang C (2004) Method for identifying product assets in a supply chain used to satisfy multiple customer demands. U.S. Patent 20050177465A1, 17 Oct 2004

Promoting O.R.: The Science of Better (2005) Matching assets to supply chain demand at IBM microelectronics. http://www.orchampions.org/prove/success_stories/mascdi.htm

Simon HA (1957) Administrative behavior, 2nd edn. The Free Press, New York

Shobrys D (2003) "History of APS," Supply Chain Consultants (www.supplychain.com), 460 Fairmont Drive, Wilmington, DE 19808, USA

Shobrys D, Fraser J (2003) Planning for the next generation (supply chain planning). Manuf Eng 82(6):10–13

Singh H (2007) Personal communication with Ken Fordyce supply chain consultants (www.supplychain.com), 460 Fairmont Drive, Wilmington, DE 19808, USA

Sullivan G (1990) IBM Burlington's logistics management system (LMS). Interfaces 20(1):43–61

Sullivan G (2005) PROFIT: decision technology for supply chain management at IBM microelectronics division. In: Applications of supply chain management and E-commerce research. Springer, New York, pp 411–452

Sullivan G (2007) Evaluating planning engines in 1994, working paper, chapter in memoirs

Sullivan G, Jantzen J, Morreale M (1991) Using Boolean matrices or integer vectors to analyze networks. In: APL91 Proceedings editor Jan Engel, APL Quote Quad, vol 21(4), pp 174–185

Swaminathan J, Smith S (1998) Modeling supply chain dynamics: a multi agent approach. Decis Sci 29(3):607–632

Tayur S, Ganeshan R, Magazine M (1998) Quantitative models for supply chain management. Kluwer Academic, Boston, MA

Uzsoy R, Lee C, Martin-Vega LA (1992) A review of production planning and scheduling modules in the semiconductor industry, Part 1: System characteristics, performance evaluation, and production planning. IIE Trans, Scheduling Logistics 24(4):47–60

Uzsoy R, Lee C, Martin-Vega LA (1994) A review of production planning and scheduling modules in the semiconductor industry, Part 2: Shop floor control. IIE Trans, Scheduling Logistics 26(5):44–55

Wolfson R (2000) Einstein's relativity and the quantum revolution. Great Courses on CD by the Teaching Company, Chantilly, VA

Woolsey G (1979) Ten ways to go down with your MRP. Interfaces 9(5):77–80

Zisgen H (2005) EPOS – stochastic capacity planning for wafer fabrication with continuous fluid models. IBM Global Engineering Services, Decision Technology Group Mainz, Germany