Available online at www.sciencedirect.com

SCIENCE DIRECT®

European Journal of Operational Research 156 (2004) 183–193

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Production, Manufacturing and Logistics

Dynamic programming for delayed product differentiation

Hsi-Mei Hsu *, Wen-Pai Wang

Department of Industrial Engineering and Management, National Chaio Tung University, 1001 Ta Hsueh Rd., Hsinchu 30050, Taiwan

Received 1 September 1999; accepted 24 August 2000

Abstract

Product proliferation and demand uncertainty result in material requirement planning difficulties, which give rise to high inventories and low customer service. A design concept, delayed product differentiation, is advocated for reducing the impact due to inaccurate forecasts and shortening the order response time. In this paper, a dynamic programming model using an AND/OR graph is constructed to determine the product differentiation points. Taking into account the costs and benefits associated with delayed product differentiation points, we proposed an approach to suggest whether the designer ought to differentiate specific products from the common part set at each design stage. Finally, we illustrate an example to characterize the optimal product differentiation points.
© 2003 Elsevier B.V. All rights reserved.

Keywords: Dynamic programming; Differentiation point; Demand uncertainty; Product proliferation

1. Introduction

The expanding high customer service provision, rapidly changing technologies, and globalization all give rise to major challenges for manufacturers of high-technology products to compete in the world market [3,4,10,12]. Such an environment increases product proliferation and demand uncertainty, which engender forecasting quite difficult. Inaccurate forecasts leads to high inventory investment and poor customer service. Most enterprises strive for strategies that can respond quickly to the diverse product needs and overcome the related operational challenges. Redesigning processes or products so as to defer the point of product differentiation positively address these formidable tasks. Deferring product differentiation points is thus a practicable design concept to decrease the negative influences of product proliferation and demand uncertainty [6,7,10].

Normally, a manufacturing process involves multiple stages, each requiring different parts or subassemblies. Increasing the level of part commonality at an early stage of manufacturing process may delay the differentiation of products. Commonality here is defined as the use of a component by several different products. When used properly, part commonality may decrease the inventory cost, manufacturing cost, and so on. In this paper, a part-base implementation approach

* Corresponding author. Tel.: +886-3-5731761; fax: +886-3-5722392.
E-mail address: hsimei@cc.nctu.edu.tw (H.-M. Hsu).

of the product differentiation deferment is developed for reacting to uncertainty, reducing inventory levels, and strengthening the flexibility for quick response to demand changes.

The principal design considerations for increasing part commonality level as much as possible are associated with traditional concerns which take into account manufacturability and performance, part relationship and sequence, geometric shape and features, and so on [7,9]. The deferral differentiation point concept reveals to redesign a production process so that the point of differentiation into multiple products in the process is delayed as much as possible. In addition, enhancing the level of part commonality exist a major problem, which the product differentiation points should be determined under considerations of the costs and benefits in the design phase. Deferment with product design changes may require extra investment. For example, a keyboard manufacturer in Taiwan invests a lot of capital in module design for supplying variable demands of the final labeling of characters and symbols. Thus, a comprehensive analysis of the right product and process design for deferment should be carried out by taking into account all of the relevant cost and benefit factors involved in the design process. The corresponding arguments include the capital investment, the inventory holding costs and the processing costs. On the other hand, such a deferment design provides several benefits. Significant advantages involve increased flexibility in the process to cope with market uncertainties and lower inventory for the same target service level. Namely, the benefits in the form of inventory reduction or service improvement are actually quite similar in many aspects to the risk-pooling effect of multi-echelon inventory systems [5,14].

Although the issue of product differentiation deferment has provoked a great deal of discussion, little attention has been focused on the corresponding approaches. Lee and Tang [11] described a model that illustrates the costs and benefits associated with delaying product differentiation. This model was however intended for strategic planning rather than tactical planning. As mentioned above, proceeding with the cost and benefit analysis of deferment alternatives at each design stage is crucial. Dynamic programming approach is a useful mathematical technique for making a sequence of interrelated decisions, which substantially reflect the specific features of product design problems. Such an approach has been successfully applied to many fields of production research [1,8,13,15].

Consequently, the main objective of this study was to construct a dynamic programming model to capturing the cost analysis of deferment design. The approach is accompanied with an AND/OR graph for representing the correspondent relationship of decision against state, which may present an aggregation of multiple states. In the next section, we briefly describe the design concepts for deferring product differentiation. A product differentiation dynamic programming model (PDDP) is constructed and discussed in Section 3. In Section 4, an example is used to illustrate the suitability of the proposed model. Finally, we present some concluding remarks.

2. Concepts for deferring the product differentiation points

The major issues in the production system were efficiency in the 80s and quality in the 90s. Quick response associated with lead-time shortening advances the principal topic for the current production research. However, today's market environment is characterized by diverse customer tastes and preferences, rapid technology development and globalization management. These factors result in the need for variety of products, which presents major challenges to production managers. In order to overcome the curse of product proliferation, an increasing trend towards redesigning the product and process so that the negative impacts of product variety can be ameliorated has been emphasized.

Design for deferring product differentiation is a strategy whereby the final configuration of a product is postponed as much as possible, usually until a customer order is received. There are two types of postponement, which are time phase and form phase, respectively [9]. For most high-technology and complex products, time postponement strategy is commonly adopted that some of the differentiation tasks traditionally implemented at the central plant have moved to regional distribution centers for reacting to localization needs. Since it involves the entire global supply chain structure, there exist many influential factors that significantly enhance the analysis complexity. Aware of the distinct characteristics, volatile and difficult-to-predict demand, of high-technology products, a corresponding strategy of build-to-order (BTO) initiated by Dell and Gateway has continuously been adapted to capture the demand variability in the personal computer industry. A successful BTO achievement is Fujitsu that established a configuration center to proceed the final assembly in Tennessee.

Form postponement aims at standardizing the upstream stages as much as possible. Often, this is also accompanied by redesigning the product with increased modular structure and part commonality, which presents the risk-pooling effect and extra capital investment. In general, multiple end-products may share some common processes and/or parts in the initial stages of the production process. At some point in the process, specialized processes or components are then used to customize the work-in-process (WIP), which was a generic product up to that point in the process, into the different end-products. Such a point is usually known as the point of product differentiation. Form postponement refers to redesigning the process to delay the point of product differentiation. In this way, Compaq simplifies the product structure, and then the complexity of product mix is reduced to enhance the effect of BTO implementation. In addition, a 98/3 strategy, which indicates 98% of customer service level and 3 days of order fulfillment, is accomplished. In this paper, we concentrate our attention on form postponement to explore the differentiation point decisions in the manufacturing process.

A correct redesign analysis of product and process for deferring the product differentiation points should be achieved under prudent concerns of all the relevant cost factors impacted by deferment. As a general rule, risk-pooling effects involved in deferment strategy has significant impacts on inventory investment, which is a central driver of deferment evaluation. In other words, product differentiation deferment should lead to lower inventory investment for the same service target level. The magnitude of inventory reduction generally depends on the timing of product differentiation points, the variability and interdependency of the different product demands, and the specific service targets. A formal model developed by Lee [11] illustrated that the inventory can be reduced when the product differentiation point is deferred. Especially, the amount of inventory reduction is greater when the different product demands are more negatively correlated.

A practical analysis for deferment design may start with an evaluation of the amount of inventory investment needed by a new design that aimed at form postponement, while the same customer service level can be satisfied. In this study, we kept the customer service target as a constant, say as same as 98% accomplished by Compaq, and evaluated different deferment design alternatives in terms of their associated inventory investment. The corresponding advantage is that we can avoid the difficulty of comparing alternatives, which have different combinations of inventory and service.

On the other hand, capital investment is required for product and/or process redesign. Implementation costs such as retraining, retesting, retooling, common parts for multiple product versions, installation of special equipment at downstream process stage, are quantifiable and need to be considered in the redesign analysis. Since technology advances rapidly and product life cycle shortens gradually, the capital investment risks become increasingly higher. Hence, redesigning the product and process for postponement strategy should take heed to above-mentioned cost factors. At this point, only few research attempts have been made at concrete analysis models or approaches. Since the solution procedure of dynamic programming is designed to find an optimal policy for the overall problem, a prescription of the optimal policy decision is determined at each design stage. This approach provides a policy prescription of what to do under every possible circumstance. Therefore, in the following section we propose a dynamic programming model to provide a system-wide analysis concerning the postponement design associated with the cost drivers mentioned above for the managers.

3. The product differentiation dynamic programming (PDDP) model

The production system considered in this paper is a divergent network. It represents that a product differentiated at some design stage will proceed to its own assembly operations in the remaining stages and will not be considered with other products for the possibility of a design commonality (Fig. 1). It refers to part commonality, which is the application of form postponement. Hence, the common pool set contains all types of products at the initial stage of the process. Following the progress of the process, the designer has to decide what product types should be differentiated from the common pool set at each design stage.

We considered an existing divergent production system that produces J types of products, where each product requires processes performed in N stages. The process stages of the manufacturing system are numbered in ascending order. A buffer is held to store the WIP inventory right after each stage. Each stockpoint controls its inventory level using a periodic review policy [2]. This policy means that the inventory position is reviewed each unit time period. At each review, a replenishment order is issued so that the inventory position immediately after placing the order equals the order-up-to level S. Each stockpoint uses its own control parameter S. The levels are set to meet the target service levels. Moreover, we assumed that different stockpoints have the same review period (e.g. 1 week), and a replenishment order is issued on the same day (e.g. Monday).

All feasible designs for the manufacturing routing are initially established as the possible states in each stage n, $s^{(n)}$. The stage means that designers ought to determine whether to continue with common processing or with some specific processing. The state $s^{(n)}$ denotes a part, which is the WIP for some products. An AND/OR graph applied here may generate all feasible designs for the product differentiation decisions. The branches of an AND/OR graph which are linked by arcs represent AND alternatives which must be achieved simultaneously. When the branches are not linked by arcs, only one branch may be selected which implies OR alternatives. In an AND/OR graph (Fig. 2), the design decision at stage n is denoted as $x^{(n)}$ where $x^{(n)} = d_k^{(n)}$, $k = 1, 2, 3, \ldots, K_n$, and K_n is total number of decisions for stage n. For example as shown in Fig. 2, numbers in the rectangle of state $s^{(n)}$ stand for the WIP of final product 1, 2, and 3. Selecting decision $x^{(n)} = d_1^{(n)}$

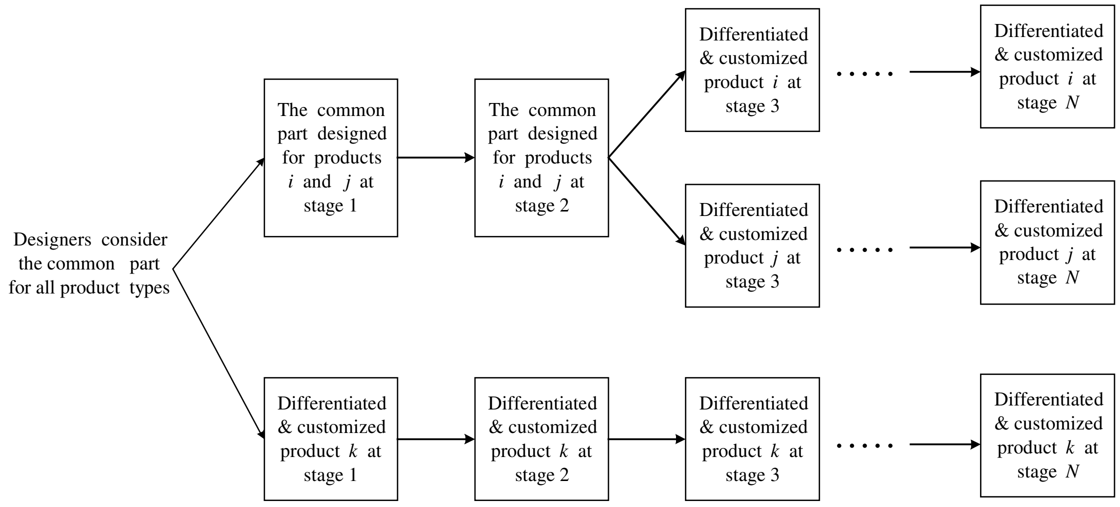
Designers consider the common part for all product types
The common part designed for products i and j at stage 1
The common part designed for products i and j at stage 2
Differentiated & customized product i at stage 3
Differentiated & customized product i at stage N
Differentiated & customized product j at stage 3
Differentiated & customized product j at stage N
Differentiated & customized product k at stage 1
Differentiated & customized product k at stage 2
Differentiated & customized product k at stage 3
Differentiated & customized product k at stage N
.....

Fig. 1. An example of the concerned divergent for three product types.

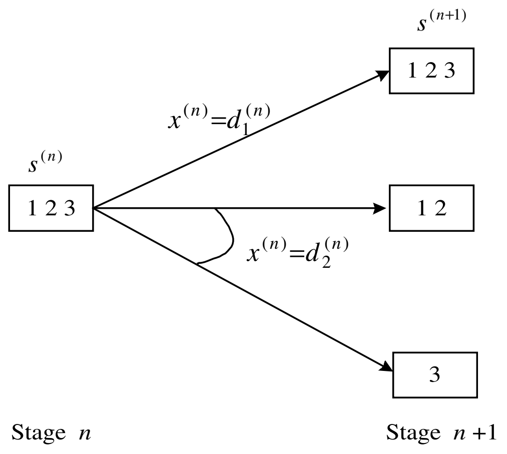

Fig. 2. An example of basic structure for design decisions among states of two successive stages in an AND/OR graph.

then moves the process from current state $s^{(n)}$ to some state $s^{(n+1)}$ at next stage, which denotes the WIP of final product 1, 2, and 3. Meanwhile, making decision $x^{(n)} = d_2^{(n)}$ indicates that one specific processing is required for the WIP of final product 1 and 2, and another for the WIP of final product 3 only. In the case of $x^{(n)} = d_2^{(n)}$, state $s^{(n)}$ (1 2 3) moves to two states (1 2) and (3) simultaneously at stage $n+1$. These two states are represented as $S(x^{(n)})$, i.e., $S(d_2^{(n)}) = \{(1\,2), (3)\}$.

The following notation was used to model the dynamic program for the design problem of product differentiation deferment:

N	total stages in the system
n	index of the stage, $n = 1, 2, \ldots, N$
J	total product types
j	index of the product, $j = 1, 2, \ldots, J$
D_j	demands for product j per time period is denoted by an i.i.d. random variable, where it is normally distributed with (μ_j, σ_j)
β	service level required for satisfying the needs of back-end stage of the process
$\mathrm{lt}(s^{(n)})$	processing lead-time required for executing operations of the common part represented by state $s^{(n)}$
$\mathrm{IC}(s^{(n)})$	extra investment costs for executing operations of the common part represented by state $s^{(n)}$. This term is determined by the allocation of the process and the extra design expenditure for the equipment or materials needed at stage n
$\mathrm{PC}(s^{(n)})$	unit processing cost for executing operations of the common part represented by state $s^{(n)}$
$\mathrm{HC}(s^{(n)})$	unit inventory holding cost of the specific components for executing operations of the common part represented by state $s^{(n)}$. The buffer average inventory level of the specific part for state $s^{(n)}$ is

$$I_{s^{(n)}} = \frac{\sum_{j\in s^{(n)}} \mu_j}{2} + z_{s^{(n)}}\sigma_{s^{(n)}}\sqrt{\mathrm{lt}(s^{(n)}) + 1}$$

where $\sigma_{s^{(n)}}$ denotes the aggregation standard deviation of products represented by $s^{(n)}$ and can be computed as

$$\sigma_{s^{(n)}} = \sqrt{\mathrm{Var}\left(\sum_{j\in s^{(n)}} D_j\right)} = \sqrt{\sum_{j\in s^{(n)}} \sigma_j^2 + 2\sum_{j,k\in s^{(n)},} \sum_{j<k} \mathrm{Cov}(D_j, D_k)}.$$

$z_{s^{(n)}}$ is the safety factor that satisfies β and can be computed as

$$z_{s^{(n)}} = [1/2\sqrt{(2/\pi)}]\ln[\beta/(1-\beta)].$$

Table 1
product patterns and relative demand data

Pattern $j (j = 1, 2, 3, 4)$	Correlation	Standard deviation
(1) Japan		
(2) USA	$\rho_{12} = -0.4$; $\rho_{13} = -0.4$;	$\sigma_{12} = 1096$; $\sigma_{34} = 1342$;
(3) England	$\rho_{14} = -0.4$; $\rho_{23} = -0.3$;	$\sigma_{123} = 894$; $\sigma_{234} = 1265$;
(4) Germany	$\rho_{24} = -0.3$; $\rho_{34} = -0.1$	$\sigma_{1234} = 447$
$D_j \sim N(10{,}000, 1000)$		

Table 2
Relative cost data for recursive function

$s^{(n)}$	$x^{(n)}$	$s^{(n+1)}$	$\mathrm{IC}(s^{(n)})$	$\mathrm{PC}(s^{(n)})$	$\mathrm{HC}(s^{(n)})$
$s_1^{(1)}$	$d_1^{(1)}$	$s_1^{(2)}$	3000	2	2
$s_1^{(1)}$	$d_2^{(1)}$	$s_2^{(2)}$	2000	4	4
$s_1^{(1)}$	$d_2^{(1)}$	$s_3^{(2)}$	2000	4	4
$s_1^{(1)}$	$d_3^{(1)}$	$s_4^{(2)}$	2500	5	4
$s_1^{(1)}$	$d_3^{(1)}$	$s_5^{(2)}$	1500	3	6
$s_1^{(2)}$	$d_1^{(2)}$	$s_1^{(3)}$	2000	6	4
$s_1^{(2)}$	$d_1^{(2)}$	$s_2^{(3)}$	1000	4	8
$s_1^{(2)}$	$d_2^{(2)}$	$s_3^{(3)}$	1500	5	6
$s_1^{(2)}$	$d_2^{(2)}$	$s_4^{(3)}$	1500	5	6
$s_1^{(2)}$	$d_3^{(2)}$	$s_5^{(3)}$	2000	6	4
$s_1^{(2)}$	$d_3^{(2)}$	$s_7^{(3)}$	1000	4	8
$s_2^{(2)}$	$d_4^{(2)}$	$s_3^{(3)}$	0	3	6
$s_3^{(2)}$	$d_5^{(2)}$	$s_4^{(3)}$	0	3	6
$s_3^{(2)}$	$d_6^{(2)}$	$s_6^{(3)}$	1000	4	8
$s_3^{(2)}$	$d_6^{(2)}$	$s_7^{(3)}$	1000	4	8
$s_4^{(2)}$	$d_7^{(2)}$	$s_3^{(3)}$	1500	5	6
$s_4^{(2)}$	$d_7^{(2)}$	$s_6^{(3)}$	1000	4	8
$s_4^{(2)}$	$d_8^{(2)}$	$s_5^{(3)}$	0	3	4
$s_5^{(2)}$	$d_9^{(2)}$	$s_7^{(3)}$	0	3	8
$s_1^{(3)}$	$d_1^{(3)}$	$s_1^{(4)}$	0	4	6
$s_1^{(3)}$	$d_2^{(3)}$	$s_3^{(4)}$	1500	7	10
$s_1^{(3)}$	$d_2^{(3)}$	$s_7^{(4)}$	1500	7	10
$s_1^{(3)}$	$d_2^{(3)}$	$s_8^{(4)}$	1500	7	10
$s_1^{(3)}$	$d_3^{(3)}$	$s_3^{(4)}$	500	5	10
$s_1^{(3)}$	$d_3^{(3)}$	$s_5^{(4)}$	1000	6	8
$s_2^{(3)}$	$d_4^{(3)}$	$s_2^{(4)}$	0	4	10
$s_3^{(3)}$	$d_5^{(3)}$	$s_2^{(4)}$	500	5	10
$s_3^{(3)}$	$d_5^{(3)}$	$s_3^{(4)}$	500	5	10
$s_3^{(3)}$	$d_6^{(3)}$	$s_4^{(4)}$	0	4	8
$s_4^{(3)}$	$d_7^{(3)}$	$s_5^{(4)}$	0	4	8
$s_4^{(3)}$	$d_8^{(3)}$	$s_7^{(4)}$	500	5	10
$s_4^{(3)}$	$d_8^{(3)}$	$s_8^{(4)}$	500	5	10
$s_5^{(3)}$	$d_9^{(3)}$	$s_4^{(4)}$	1000	6	8
$s_5^{(3)}$	$d_9^{(3)}$	$s_7^{(4)}$	500	5	10
$s_5^{(3)}$	$d_{10}^{(3)}$	$s_6^{(4)}$	0	4	6
$s_5^{(3)}$	$d_{11}^{(3)}$	$s_2^{(4)}$	500	5	10
$s_5^{(3)}$	$d_{11}^{(3)}$	$s_3^{(4)}$	500	5	10
$s_5^{(3)}$	$d_{11}^{(3)}$	$s_7^{(4)}$	500	5	10
$s_6^{(3)}$	$d_{12}^{(3)}$	$s_7^{(4)}$	0	4	10
$s_7^{(3)}$	$d_{13}^{(3)}$	$s_8^{(4)}$	0	4	10
$s_1^{(4)}$	$d_1^{(4)}$	$s_2^{(5)}$	1000	8	12
$s_1^{(4)}$	$d_1^{(4)}$	$s_3^{(5)}$	1000	8	12
$s_1^{(4)}$	$d_1^{(4)}$	$s_4^{(5)}$	1000	8	12
$s_2^{(4)}$	$d_2^{(4)}$	$s_1^{(5)}$	0	5	12
$s_3^{(4)}$	$d_3^{(4)}$	$s_2^{(5)}$	0	5	12
$s_4^{(4)}$	$d_4^{(4)}$	$s_1^{(5)}$	500	7	12
$s_4^{(4)}$	$d_4^{(4)}$	$s_2^{(5)}$	500	7	12
$s_5^{(4)}$	$d_5^{(4)}$	$s_3^{(5)}$	500	7	12
$s_5^{(4)}$	$d_5^{(4)}$	$s_4^{(5)}$	500	7	12
$s_6^{(4)}$	$d_6^{(4)}$	$s_1^{(5)}$	1000	8	12
$s_6^{(4)}$	$d_6^{(4)}$	$s_2^{(5)}$	1000	8	12
$s_6^{(4)}$	$d_6^{(4)}$	$s_3^{(5)}$	1000	8	12
$s_7^{(4)}$	$d_7^{(4)}$	$s_3^{(5)}$	0	5	12
$s_8^{(4)}$	$d_8^{(4)}$	$s_4^{(5)}$	0	5	12

The cost function and recursive relationship of the proposed PDDP model are displayed as follows:

$f_n(s^{(n)}, x^{(n)})$ the cost function for stage n as $s^{(n)}$ and $x^{(n)}$ are its current state and the corresponding design decision, respectively.

$$f_n(s^{(n)}, x^{(n)}) = \sum_{s^{(n+1)} \in S(x^{(n)})} C(s^{(n+1)}, x^{(n)}) + \sum_{s^{(n+1)} \in S(x^{(n)})} f_{n+1}^{*}(s^{(n+1)}),$$

where $f_{n+1}^{*}(s^{(n+1)})$ indicates the corresponding minimum costs of state $s^{(n+1)}$ at next stage.

Accordingly, recursive relationship is

$$f_n^{*}(s^{(n)}) = \mathrm{Min}\left\{ \sum_{s^{(n+1)} \in S(x^{(n)})} C(s^{(n+1)}, x^{(n)}) + \sum_{s^{(n+1)} \in S(x^{(n)})} f_{n+1}^{*}(s^{(n+1)}) \right\}.$$

$\sum_{s^{(n+1)} \in S(x^{(n)})} C(s^{(n+1)}, x^{(n)})$ the total operation cost when design decision $x^{(n)}$ is adopted and the differentiation process is moving to some state $s^{(n+1)}$, $s^{(n+1)} \in S(x^{(n)})$.

In addition,

$$C(s^{(n+1)}, x^{(n)}) = \mathrm{IC}(s^{(n+1)}) + \mathrm{PC}(s^{(n+1)}) \sum_{j \in s^{(n+1)}} D_j + \mathrm{HC}(s^{(n+1)}) I_{s^{(n+1)}},$$

which includes the extra investment cost, processing cost and inventory holding cost.

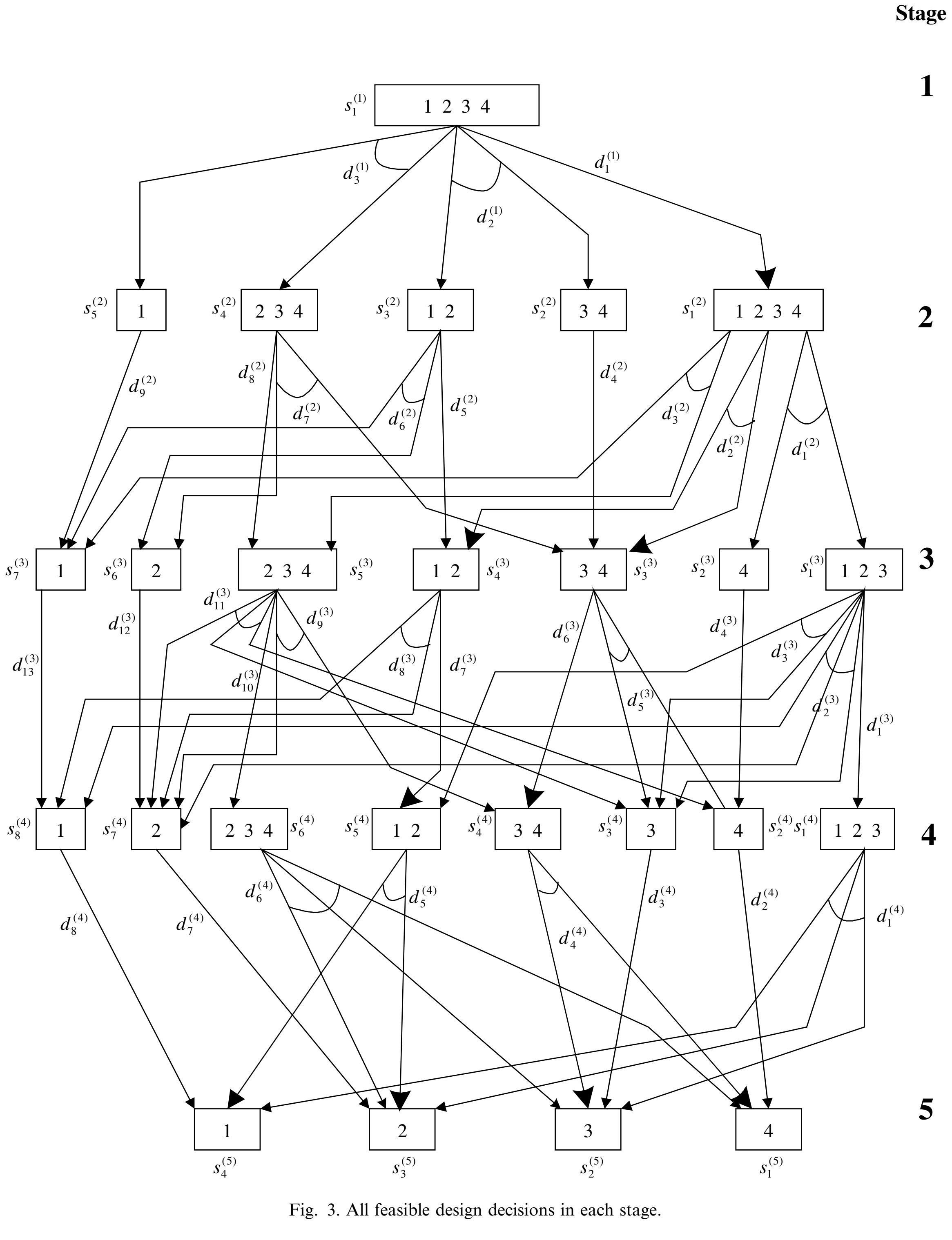

Fig. 3. All feasible design decisions in each stage.

4. Illustrative example

A manufacturing system involves five major stages ($N = 5$) through which each of five end-products ($J = 4$) must proceed. Redesigning the products or processes for the postponement strategy perhaps requires extra investment to arrange the common parts for use in some of these products. This increases capital introduction, but there is a great advantage to risk-pooling effects. At the initial stage of the process, the common pool set contains all components required for 4 product types. Hence, which kind of components should be differentiated from this set for the next stage must be decided by product designers.

The first step in designing differentiation processes is the establishment of the feasibility of the common parts for the specific product patterns at each stage. Within this step, the engineers and designers analyze and determine the proper structure and functionality for the common parts. After prudent analysis and screening test by management and design teams, feasible design alternatives are therefore determined at each stage.

Assume that the service level required for each stockpoint is 98% and the lead-time for each stage takes 1 week. Demands for each product per week have an identical normal distribution with mean $\mu_j = 10{,}000$ and standard deviation $\sigma_j = 1000$ ($j = 1, 2, 3, 4$). Moreover, we assume that the demands of the end-products represent negatively correlated to take advantage of the increase in the inventory savings. Table 1 presents the relevant data of product demands for the dynamic programming analysis. Table 2 shows the required cost parameter values corresponding to each feasible state.

At each design stage, a decision should be made whether to continue with the common part assemblies or to proceed with specific component differentiation. The recursive relationship keeps recurring as we start at the end stage and move *backward* stage by stage. That is, a backward search should be executed to determine the most meaningful differentiation plan. Fig. 3 depicts all feasible design alternatives at each design stage. The computational results at each stage are shown in Tables 3–6 . An optimal solution for the entire product differentiation problem can now be identified from the four tables. These results also are summarized in Fig. 4. Following the arrows in Fig. 4 from stages 1 to 5 gives the prescriptive optimal solutions. Namely, the results show that all components required for different product patterns are differentiated at the final stage. It is to design a common part used for patterns 1 and 2, and another common part used for patterns 3 and 4 at stages 3 and 4, respectively. And so forth it is to design a common part used for all patterns in the initial two stages.

This example clarifies how the differentiation plan should be determined and executed. It shows the potency of the proposed PDDP approach and its application for solving a practical design

Table 3
Computational result for $n = 4$

$s^{(4)}$	$x^{(4)}$								$x^{(4)*}$	$f_4^*(s^{(4)})$
	$f_4(s^{(4)}, x^{(4)}) = \sum_{s^{(5)} \in S(x^{(4)})} C(s^{(5)}, x^{(4)})$									
	$d_1^{(4)}$	$d_2^{(4)}$	$d_3^{(4)}$	$d_4^{(4)}$	$d_5^{(4)}$	$d_6^{(4)}$	$d_7^{(4)}$	$d_8^{(4)}$		
$s_1^{(4)}$	547164								$d_1^{(4)}$	547164
$s_2^{(4)}$		151388							$d_2^{(4)}$	151388
$s_3^{(4)}$			151388						$d_3^{(4)}$	151388
$s_4^{(4)}$				343776					$d_4^{(4)}$	343776
$s_5^{(4)}$					343776				$d_5^{(4)}$	343776
$s_6^{(4)}$						547164			$d_6^{(4)}$	547164
$s_7^{(4)}$							151388		$d_7^{(4)}$	151388
$s_8^{(4)}$								151388	$d_8^{(4)}$	151388

Table 4
Computational result for $n = 3$

$s^{(3)}$	$x^{(3)}$													$x^{(3)*}$	$f_3^*(s^{(3)})$
	$f_3(s^{(3)}, x^{(3)}) = \sum_{s^{(4)} \in S(x^{(3)})} C(s^{(4)}, x^{(3)}) + \sum_{s^{(4)} \in S(x^{(3)})} f_4^*(s^{(4)})$														
	$d_1^{(3)}$	$d_2^{(3)}$	$d_3^{(3)}$	$d_4^{(3)}$	$d_5^{(3)}$	$d_6^{(3)}$	$d_7^{(3)}$	$d_8^{(3)}$	$d_9^{(3)}$	$d_{10}^{(3)}$	$d_{11}^{(3)}$	$d_{12}^{(3)}$	$d_{13}^{(3)}$		
$s_1^{(3)}$	775662	922134	801394											$d_1^{(3)}$	775662
$s_2^{(3)}$				275878										$d_4^{(3)}$	275878
$s_3^{(3)}$					572756	540808								$d_6^{(3)}$	540808
$s_4^{(3)}$							534016	572756						$d_7^{(3)}$	534016
$s_5^{(3)}$									868186	783342	631134			$d_{11}^{(3)}$	631134
$s_6^{(3)}$												275878		$d_{12}^{(3)}$	275878
$s_7^{(3)}$													275878	$d_{13}^{(3)}$	275878

Table 5
Computational result for $n = 2$

$s^{(2)}$	$x^{(2)}$									$x^{(2)*}$	$f_2^*(s^{(2)})$
	$f_2(s^{(2)}, x^{(2)}) = \sum_{s^{(3)} \in S(x^{(2)})} C(s^{(3)}, x^{(2)}) + \sum_{s^{(3)} \in S(x^{(2)})} f_3^*(s^{(3)})$										
	$d_1^{(2)}$	$d_2^{(2)}$	$d_3^{(2)}$	$d_4^{(2)}$	$d_5^{(2)}$	$d_6^{(2)}$	$d_7^{(2)}$	$d_8^{(2)}$	$d_9^{(2)}$		
$s_1^{(2)}$	3137992	2593790	10016584							$d_2^{(2)}$	2593790
$s_2^{(2)}$				1261338						$d_4^{(2)}$	1261338
$s_3^{(2)}$					1261338	768940				$d_6^{(2)}$	768940
$s_4^{(2)}$							1715712	2450114		$d_7^{(2)}$	1715712
$s_5^{(2)}$									373470	$d_9^{(2)}$	373470

Table 6
Computational result for $n = 1$

$s^{(1)}$	$x^{(1)}$			$x^{(1)*}$	$f_1^*(s^{(1)})$
	$f_1(s^{(1)}, x^{(1)}) = \sum_{s^{(2)} \in S(x^{(1)})} C(s^{(2)}, x^{(1)}) + \sum_{s^{(2)} \in S(x^{(1)})} f_2^*(s^{(2)})$				
	$d_1^{(1)}$	$d_2^{(1)}$	$d_3^{(1)}$		
$s_1^{(1)}$	3264076	3569252	4851442	$d_1^{(1)}$	3264076

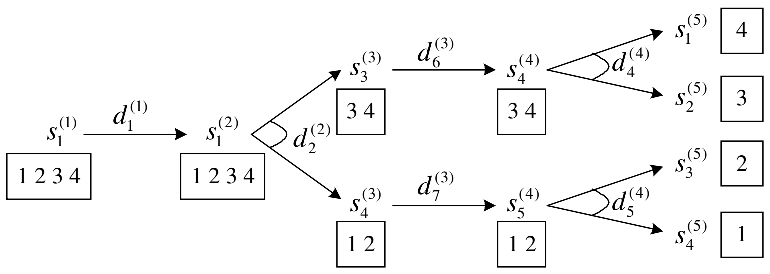

Fig. 4. Graphical display of the dynamic programming solution of the example.

problem of product differentiation deferment. Using the suggested dynamic approach, the differentiation problems can be solved in a systematic way, which provides satisfactory results. This approach is the basis for a logical formulation, search and selection of the most meaningful differentiation process. It should be considered when dealing with the issue of determining the deferral points of product differentiation.

5. Conclusions

To respond to the challenge of product proliferation and demand uncertainty, deferring product differentiation in the product design process is imperative. We propose a dynamic programming approach following an AND/OR graph to concretely analyzing the decisive design problem for differentiation deferment. The impact of deferment on capital investment and inventory risk-pooling effects were quantified and incorporated in the model. The represented PDDP model is intended for tactical planning rather than strategic planning under considerations of the investment costs, the inventory holding costs and the processing costs. This tactic defines the meaningful product differentiation process for execution. The PDDP model provides conditions to decide at every design stage and to designate which step must follow. In fact, the current study expects to play the pioneer role in solving the entire product/process redesign problem concerning the differentiation point deferment. Future research includes expanding the cost terms considered in the objective and devoting computational efforts in the solution procedure. In addition, the issues of the lead time and service levels integrated into the model are deserve for studying the quick response effectiveness in detail.

References

[1] J. Antonio, F. Chauvet, C. Chu, J.M. Proth, The cutting stock problem with mixed objectives: Two heuristics based on dynamic programming, European Journal of Operational Research 114 (1999) 395–402.
[2] A. Clark, H. Scarf, Optimal policies for a multi-echelon inventory problem, Management Science 6 (1960) 474–490.
[3] P. Child, R. Diederichs, F. Sanders, S. Wisniowski, The management of complexity, Sloan Management Review, Fall (1991) 73–80.
[4] M.A. Cohen, H.L. Lee, Strategic analysis of integrated production-distribution systems: Models and methods, Operations Research 36 (1988) 216–228.
[5] G.D. Eppen, L. Schrage, in: L.B. Schwarz (Ed.), Centralized Ordering Policies in a Multi-Warehouse System with Lead Times and Random Demand in Multi Level Produc-

tion/Inventory Systems: Theory and Practice, North-Holland, Amsterdam, 1981.

[6] D.W. He, A. Kusiak, Design of assembly systems for modular products, IEEE Transactions on Robotics and Automation 13 (5) (1997) 646–655.

[7] D.W. He, A. Kusiak, T.L. Tseng, Delayed product differentiation: A design and manufacturing perspective, Computer-Aided Design 30 (2) (1998) 105–113.

[8] C.Y. Lee, Two-machine flowshop scheduling with availability constraints, European Journal of Operational Research 114 (1999) 420–429.

[9] H.L. Lee, C. Billington, Designing Products and Processes for Postponement, Working Paper, 1994.

[10] H.L. Lee, Effective management of inventory and service through product and process redesign, Operations Research 44 (1996) 151–159.

[11] H.L. Lee, C.S. Tang, Modeling the costs and benefits of delayed product differentiation, Management Science 43 (1) (1997) 40–53.

[12] M.J. Maloni, W.C. Benton, Supply chain partnerships: Opportunities for operations research, European Journal of Operational Research 101 (1997) 419–429.

[13] K.D. Penev, A.J. Ron, Determination of a disassembly strategy, International Journal of Production Research 34 (2) (1996) 495–506.

[14] L.B. Schwarz, Models for assessing the value of warehouse risk-pooling: Risk-pooling over outside-supplier lead times, Management Science 35 (1989) 828–842.

[15] A.I. Sonmez, A. Baykasoglu, A new dynamic programming formulation of $(n * m)$ flowshop sequencing problems with due dates, International Journal of Production Research 36 (6) (1998) 2269–2283.