MANAGEMENT SCIENCE
Vol. 54, No. 5, May 2008, pp. 956–968
ISSN 0025-1909 | EISSN 1526-5501 | 08 | 5405 | 0956

informs

DOI 10.1287/mnsc.1070.0784

Sequential Testing of Product Designs: Implications for Learning

Sanjiv Erat
Rady School of Management, University of California at San Diego, La Jolla, California 92093,
serat@ucsd.edu

Stylianos Kavadias
College of Management, Georgia Institute of Technology, Atlanta, Georgia 30308,
stelios@gatech.edu

Past research in new product development (NPD) has conceptualized prototyping as a "design-build-test-analyze" cycle to emphasize the importance of the analysis of test results in guiding the decisions made during the experimentation process. New product designs often involve complex architectures and incorporate numerous components, and this makes the ex ante assessment of their performance difficult. Still, design teams often learn from test outcomes during iterative test cycles enabling them to infer valuable information about the performances of (as yet) untested designs. We conceptualize the extent of useful learning from analysis of a test outcome as depending on two key structural characteristics of the design space, namely whether the set of designs are "close" to each other (i.e., the designs are similar on an attribute level) and whether the design attributes exhibit nontrivial interactions (i.e., the performance function is complex).

This study explicitly considers the design space structure and the resulting correlations among design performances, and examines their implications for learning. We derive the optimal dynamic testing policy, and we analyze its qualitative properties. Our results suggest optimal continuation only when the previous test outcomes lie between two thresholds. Outcomes below the lower threshold indicate an overall low performing design space and, consequently, continued testing is suboptimal. Test outcomes above the upper threshold, on the other hand, merit termination because they signal to the design team that the likelihood of obtaining a design with a still higher performance (given the experimentation cost) is low. We find that accounting for the design space structure splits the experimentation process into two phases: the initial *exploration* phase, in which the design team focuses on obtaining information about the design space, and the subsequent *exploitation* phase in which the design team, given their understanding of the design space, focuses on obtaining a "good enough" configuration. Our analysis also provides useful contingency-based guidelines for managerial action as information gets revealed through the testing cycle. Finally, we extend the optimal policy to account for design spaces that contain distinct design subclasses.

Key words: sequential testing; design space; complexity; contingency analysis
History: Accepted by Christoph Loch, R&D and product development; received April 29, 2005. This paper was with the authors 1 year and 5 months for 2 revisions. Published online in *Articles in Advance* March 1, 2008.

1. Introduction

Before launching a new product, firms often test multiple product prototypes to gather performance and market data, and to decrease the technical and market uncertainty. Various studies emphasize the importance of testing and experimentation for successful new product development (NPD) and, at the same time, recognize that testing may consume substantial resources (Allen 1977, Clark and Fujimoto 1989, Cusumano and Selby 1995, Thomke 2003). Thus, these studies highlight a key trade-off involved in experimentation: performance benefits versus costs of additional experiments.

Over the years, a number of studies have examined this fundamental trade-off and its associated drivers. Early on, Simon (1969) observed the need to undertake multiple trials (experiments) to obtain a high performing outcome (design). DeGroot (1968) modeled this process as the repeated sampling of a random variable representing the design performance. In a seminal study, Weitzman (1979) recognizes that the different design performances are not necessarily "realizations" of the same random variable. Instead, he views experimentation as a search over a finite set of statistically independent alternatives (designs) with distinct and known prior distributions of performances. His simple and elegant result is summarized as follows: continue testing as long as the best performance obtained so far is below a "reservation performance."

Clark and Fujimoto (1989) present evidence from practice that design configuration tests are not simply

arbitrary draws from independent performance distributions, but rather are iterative "design-build-test" tasks. Their study forms the basis for Thomke's (1998, 2003) conceptualization of the experimentation process as an iterative "design-build-test-*analyze*" cycle. Thomke's (1998, 2003) research is the first to emphasize the aspect of learning during experimentation, and to highlight that the analysis phase enables a design team to understand the design space better and revise the test schedules accordingly.

Building on Thomke's (1998) empirical findings, Loch et al. (2001) analytically examine which mode of experimentation (parallel versus sequential experimentation) minimizes the overall experimentation costs. In their model, the design team gradually learns through imperfect sequential experiments, and identifies which design is the "best" in fulfilling an exogenously specified functionality.

In this study, we recognize that the extent of *transferable* learning (i.e., relevance of the outcome from testing one design configuration in understanding and predicting the performance of another configuration) is directly related to the design space structure (e.g., the similarities among different design configurations). Thomke et al. (1999) offer a tangible illustration of how such transferable learning occurs in practice: the design team, conducting simulation-based side-impact crash tests of BMW cars, observed during the course of testing that weakening a particular part of the body structure (specifically, a pillar) led to superior crash performance of the vehicle. This insight allowed them to infer the same relationship in other related (similar) designs, and to improve the overall safety of BMW's product offerings.

This paper seeks to answer the following research questions: (i) How does the design space structure influence the optimal number of experiments and (ii) how does the design space structure drive the design team's learning? We motivate our formal treatment of the problem with the following numerical example.

1.1. Motivating Example

Consider a product that can be realized through two design configurations, A and B, and let a subset of design attributes be identical across the two configurations. The design team assesses, a priori, that the performance of configuration A (B) is distributed normally around a mean μ_A (μ_B) with a standard deviation of σ_A (σ_B). The similarities in the design attributes between the two configurations lead the design team to expect that, if A exhibits high (low) performance, then B has a higher likelihood of a high (low) performance. Thus, the scalar performances of configurations A and B ($\mathbf{A}$ and $\mathbf{B}$, respectively) are jointly multivariate normal and exhibit positive correlation $\rho_{AB} = \theta$. We assume that testing reveals the true performance of design configurations (i.e., perfect testing), and suppose that the cost of testing design configuration A (B) is C_A (C_B). Consider the following example of parameters:

$\mu_A = 0.1 \quad \mu_B = 0$	$C_A = C_B = 0.1$
$\sigma_A = \sigma_B = 1$	$\theta = 0.8$

1.2. Base Line Case with No Learning

First, consider the case where the performance dependencies and, thus, the possibility of learning are disregarded, i.e., $\theta = 0$. Furthermore, suppose that the best performance found so far is x ($x = 0$ if no test has been conducted). Given x, continuing onto test design A at a cost C_A results in the (expected) performance $\mathrm{E}[\max\{x, \mathbf{A}\}]$. Hence, the marginal value from testing A is $\mathrm{E}[\max\{x, \mathbf{A}\}] - x - C_A$. Weitzman (1979) shows that this marginal value is positive and further experimentation is beneficial as long as the current best solution (i.e., x) is lower than a threshold (i.e., the *reservation price* of A).[1] Furthermore, Weitzman's (1979) reservation price (denoted by WRP) of A is obtained by equating the above marginal value to zero, i.e., solving for x in the equation $\mathrm{E}[\max\{x, \mathbf{A}\}] - x - C_A = 0$. In our example, the WRP for A is 1.01 and for B is 0.91. Thus, if the performance dependencies between the two configurations are disregarded, the team optimally tests A, and then continues on to test B if and only if the outcome from testing A is less than 0.91.

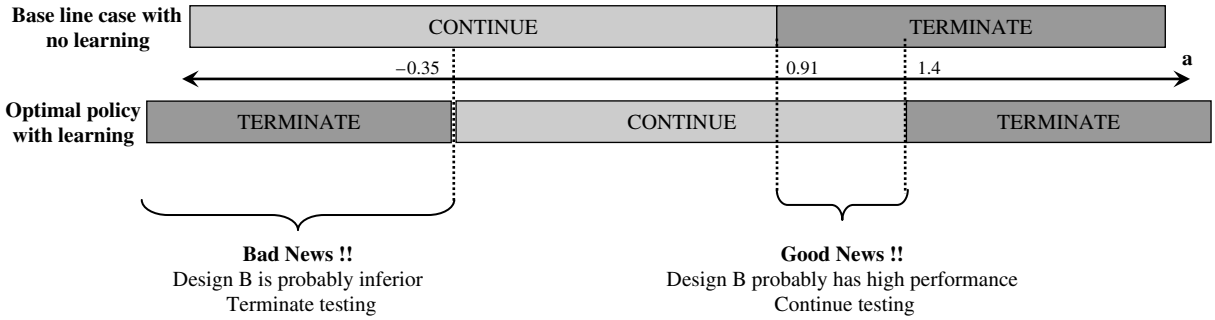
1.3. Testing Cycle with Learning

Now consider the setting where the design team accounts for performance dependencies (i.e., correlation $\theta = 0.8$). Suppose that design A is tested[2] and results in performance a. The conditional distribution of B (i.e., conditional on the realization a) is normal with mean $\mu_B + \theta(a - \mu_A)$ and standard deviation $\sigma_B\sqrt{1-\theta^2}$ (Johnson and Wichern 2002). Because no subsequent learning is possible (i.e., B is the only other design), Weitzman's (1979) result applies from this point on. Hence, the team continues to test B if and only if the WRP of B (conditional on the realization a) is greater than the best performance found so far, i.e., $0.8a + 0.28 > \max(a, 0)$. In summary, when $\theta = 0.8$, the team optimally tests A and then continues to B if and only if $a \in (-0.35, 1.4)$.

[1] Weitzman's (1979) results presented here have been adapted to our setting. His reservation price result is valid under more general settings including different costs and arbitrary distributions, as long as the alternatives are statistically independent.

[2] It is suboptimal not to test any designs because by testing A (B) and stopping immediately, the experimenter makes a strictly positive payoff $= \mathrm{E}[\max\{0, \mathbf{A}\}] - C_A$ ($\mathrm{E}[\max\{0, \mathbf{B}\}] - C_B$). Furthermore, because $\mu_A > \mu_B$, it is strictly better to test A first.

Figure 1 Differences Between Base Line Case Without Learning and Testing Cycle With Learning



A graphical comparison between the two cases is illustrated in Figure 1. The figure points toward two key benefits of learning: (i) cost savings due to early termination, when the acquired information and the associated analysis suggest that the remaining configurations would exhibit inferior performances (i.e., the $a \leq -0.35$ region), and (ii) gains by additional experimentation when the information indicates untapped potential from the untested designs (i.e., the $0.91 \leq a < 1.4$ region).

In this paper, we model the experimentation process as an iterative search over a set of feasible design configurations. We develop a stochastic dynamic program, and we characterize the optimal testing policy. Our goal is to describe the effects of design space structure on learning and, by extension, on the optimal test schedules. Furthermore, we also aim to develop and prescribe intuitive rules of thumb to aid managerial decision making during experimentation.

Our contribution is threefold. First, our conceptualization allows us to demonstrate how the design space structure (i.e., the similarities among the design configurations and the complexity of the performance function) critically determines the optimal amount of experimentation. Furthermore, we offer a more realistic approach to understanding learning in NPD experimentation, beyond Weitzman's (1979) assumption of independent designs/alternatives, and the highly specific assumption of Loch et al. (2001) that testing a design alternative signals whether or not the design is the "best." However, to maintain analytical tractability, we rely on specific assumptions about some facets of the testing process (e.g., we assume that design performances are normally distributed and that testing costs are equal within a class of designs). Yet, these assumptions do not fundamentally relate to the learning aspect of experimentation, and we consider them a fair price to pay for extending the analysis to interrelated design configurations. In addition, in §4, we offer a set of results that illustrate the mathematical intractability in obtaining closed-form solutions once we move beyond our assumptions.

Second, we identify and characterize the qualitative effects on the testing process due to the explicit consideration of the design space structure. Design teams may pursue additional tests not only to identify configurations with higher performance, but also to explore and to gain greater understanding of the design space. We term this pattern the "*exploration phase*." Such aggressive exploration appears in the initial stages of testing and is stronger when the interactions between design attributes are weaker and/or similarities between designs are greater. Once the team gains sufficient understanding of the design space, the likelihood of terminating the testing cycle increases. We term this the "*exploitation phase*." We also find that the length of the exploration phase depends on the extent of similarities between the designs. More similarities lead to a shorter exploration phase. Thus, the experimentation process is split in two phases: an exploration phase where the "design-build-test-analyze" iterations are primarily conducted based on what you *may learn*, and an exploitation phase where the "design-build-test-analyze" cycles are driven by what you *have learned*.

Third, we outline rules of thumb for guiding managerial decisions contingent on the initial test outcomes. When worse than expected performance outcomes are observed, the design team should transition sooner to an exploitation regime if the interactions between the design attributes are deemed weak. Thus, the length of the exploration regime depends critically on how the first few test results compare to the a priori performance expectations, and on the extent of interactions between the design attributes.

The model formulation is presented in §2. Subsection 3.1 presents the solution to the base case together with several important properties of the optimal policy. Subsections 3.2 and 3.3 generalize the base case model to settings where the parameter estimates are uncertain, and the case where the designs belong to different subclasses, respectively. Section 4 concludes with managerial insights, a discussion of the limitations of the current search model and some paths for future research.

2. A Formal Model of Learning in Experimentation

Consider a design team who develops and tests different configurations before launching a new product. The performance of the product design depends on M distinct design attributes. Hence, each design configuration i $(i = 1, 2, \ldots, N)$ is a point in the M-dimensional design space and is represented as a unique combination of the M attributes, namely x_{ij} $(j = 1, 2, \ldots, M)$. For example, within the premises of the Thomke et al. (1999) example, the thickness of the pillar for a design configuration i is one design attribute value x_{i1}.

The team selects the values for the attributes x_{ij} and determines the performance $P_i = P(x_{i1}, \ldots, x_{iM})$ of configuration i; $P(\cdot)$ is the performance function that maps the design space to a real number representing the design performance. As in most new product designs, the performance function $P(\cdot)$ cannot be ex ante-specified, due to incomplete knowledge or inherently uncertain and complex relationships between the attributes (Ethiraj and Levinthal 2004, Loch and Terwiesch 2007). The Thomke et al. (1999) crash test example offers a vivid illustration of how complexity naturally emerges from unforeseen interactions between the pillar and the rest of the design attributes.

Despite the presence of such complexity, the design team can often estimate, at a high level, the extent of potential interactions between the design attributes.[3] Furthermore, because a specific choice of attribute values constitutes a design configuration, limited changes in a subset of the design attributes result in similarities among designs. As the degree of attribute commonality increases, the design configurations appear more similar than others.

We model these two aspects of the design space—potential interactions between attributes and the similarity between configurations—through a performance correlation between the designs. Specifically, the performance levels of two designs are likely to be more related if there is only a small difference between the attribute values of the two designs (a smaller "distance" between the designs) and/or if the extent of interactions between the design attributes is low (a simpler performance function). On the other hand, widely different design attributes across configurations (a larger "distance" between designs) and/or a design space where the design attributes exhibit greater degree of interactions (a more complex performance function) limits the performance dependence among different configurations. To summarize, two designs are more performance correlated if they are less distant and if the performance function is less complex.

The design team may, at least roughly, estimate the correlation between any two designs because it understands the distance between the two designs and the complexity of the performance function. Furthermore, we assume that these correlations are non-negative. That is, a high (low) test outcome for a design configuration signals a higher likelihood for the performance of the other designs to be high (low). This assumption is natural in the context of performance functions (fitness landscapes).[4] For instance, Stadler (1992) demonstrates how families of performance functions (landscapes), including those generated from many combinatorial optimization problems, can be characterized by their correlations and exhibit positive correlations. Furthermore, our assumption of positive correlations conforms to NK landscapes and AR(1) processes, both of which are widely used to represent performance landscapes in complexity literature (Kauffman 1993).

Let $\vec{P} = (p_1, p_2, \ldots, p_N)$ be the vector of the performances of the N potential design configurations. $\vec{P}$ is random because of the design team's limited knowledge; we assume that the joint distribution of the scalar performances is multivariate normal. Our distributional assumption is realistic for a wide range of performance landscapes. For instance, in fitness landscapes such as the widely used NK landscapes, the performance at each point is approximately normally distributed (Kauffman 1993, p. 53). The normality assumption also results in a tractable model and allows the characterization of the optimal policy properties (our discussion in §4 illustrates that index policies may fail to exist for arbitrary distributions).

Let the design team's a priori expectations of the design performances be $\bar{\mu} = (\mu^1, \mu^2, \ldots, \mu^N)$. The a priori performance uncertainty is summarized in a performance covariance matrix Σ_N. We assume that the a priori variances of the performances of all the designs are identical $= \sigma^2$. Our assumption reflects relatively homogeneous design spaces with configurations that differ with respect to a small, limited subset of design attributes. We offer a limited relaxation of this assumption in §3.3. However, as we demonstrate in §4, the equal variance assumption is not merely convenient, but is also necessary for mathematical tractability.

[3] This is similar to assessing the interaction parameter K in an NK landscape model (Kauffman 1993).

[4] Intuitively, if the distance between two designs is low and the performance function is simple, we would expect the performances to be positively correlated. Alternatively, if the distance between designs is high and the performance function is complex, we would expect the performances to be unrelated (i.e., correlation = 0). Note that in either case the correlations are always non-negative.

In summary, our conceptualization assumes that the design team has a multivariate normal prior on the design performances. Thus, the design team knows the following parameters of the multivariate normal distribution of the design performances: (i) the performance expectations, (ii) the performance uncertainty (represented through the common standard deviation), and (iii) the performance correlations between the design configurations. Note that this does not imply knowledge of the exact performance levels, but only represents the a priori knowledge of design commonalities (based on, for example, shared components).

Suppose that the team undertakes sequential testing, and that the design configurations 1 through k have been tested. We assume that the tests are perfect, i.e., the measurement errors, if any, are negligible, and the true performance of a design is revealed when the design is tested. Then, the conditional joint distribution (conditional on the k test outcomes observed so far) of the remaining $N-k$ designs is again a multivariate normal distribution (Rencher 2002). Furthermore, the conditional distribution (conditional on the k test outcomes observed so far) of each of the remaining $N-k$ designs are univariate normal with the following conditional mean and standard deviation (Rencher 2002):

$$\hat{\mu}^i_{k+1} = \mu^i + \bar{r}^{i\prime}_k Q_k^{-1}(\bar{P}_k - \bar{\mu}_k) \quad (i = k+1, \ldots, N) \tag{1}$$

$$\hat{\sigma}^i_{k+1} = \sigma\sqrt{(1 - \bar{r}^{i\prime}_k Q_k^{-1} \bar{r}^{i\prime}_k)} \quad (i = k+1, \ldots, N), \tag{2}$$

where $\hat{\mu}^i_{k+1}$ and $\hat{\sigma}^i_{k+1}$ are the conditional mean and the standard deviation, respectively, for design configuration i; $\bar{r}^i_k$ is a $k \times 1$ column vector of a priori covariances of design i with all of the k tested designs; Q_k is a $k \times k$ sub-matrix of the covariance matrix Σ_N such that it contains the a priori covariances of all the tested designs; $\bar{\mu}_k$ is a $k \times 1$ column vector of a priori expectations of all the k tested prototypes $(=(\mu^1, \mu^2, \ldots, \mu^k))$; $\bar{P}_k$ is the $k \times 1$ column vector of performances of the k tested prototypes $(=(p_1, p_2, \ldots, p_k))$.

After the kth test, let U be the set of all untested configurations and T the set of all the tested ones. If the cost of testing design i is c_i, and $p = \max_{j \in T} p_j$ is the maximum performance observed until now, the team faces the following dynamic decision problem:

$$V_k(p, \hat{\mu}_{\mathbf{k}}, U) = \max \begin{cases} p \\ \max\limits_{j \in U} (\mathrm{E}[V_{k+1}(\max(p, p_j), \hat{\mu}_{\mathbf{k+1}}, U - \{j\})] - c_j) \end{cases} \tag{3}$$

$$V_N(p, \cdot, \varnothing) = p. \tag{4}$$

Equations (3) and (4) are the Bellman optimality equations corresponding to the decision process. The state is described by the maximum performance realized thus far, the set of untested design configurations, and the vector of the updated mean performance values of the remaining tests. The upper branch of (3) describes the payoff if the team terminates the experimentation; it is equal to p, the highest performance observed. Upon continuation, the expected payoff results from the choice of one of the untested configurations. Note that the final payoffs are given by $p = \max_{j \in T} p_j$. Thus, we assume that a design is available for market launch only if it has been tested. The investment involved in setting up a manufacturing process and a distribution channel makes any launch approval unlikely, unless a profitable business case, including testing data, is presented.

3. Effect of Learning on Optimal Testing

In this section, we derive the optimal dynamic search policy and we demonstrate its key structural properties. We proceed in three stages: In §3.1, we assume that the performance dependencies between any pair of design configurations are identical, that is, the correlations among the design performances are identical. The assumption represents settings where the design configurations lie within a specific scientific domain, and they share a substantial number of attributes (i.e., each design configuration is of the form $(x_1, x_2, \ldots, x_M)$, and the first r attributes $x_1, x_2, \ldots, x_r$ have the same value across all configurations). The design team knows the performance correlations among the designs, but not the test outcome. As stated in the introduction, this reflects knowledge about the distance between design configurations, and the complexity of the performance function. In §3.2, we relax this assumption and allow for uncertain correlations. In §3.3, we relax the assumption of identical correlations between the design configurations.

3.1. Base Case: Identical Correlations

For now, we assume that any pair of design configurations has the same correlation factor. The overall performance uncertainty of the design configurations is summarized by the covariance matrix Σ_N (defined in §2). It also summarizes the dependencies across the performance random variables, that is, the ijth element a_{ij} is the covariance between the performance random variables of design configurations i and j. Then,

$$a_{ij} = \begin{cases} \sigma^2 & \text{if } i = j \\ \theta\sigma^2 & \text{otherwise } (0 < \theta < 1), \end{cases}$$

where θ is the correlation factor.

The experimentation cost is composed of two elements (Thomke 2003): (i) the direct cost of specialized resources required to conduct the experiments, and (ii) the indirect opportunity cost resulting from product introduction delays. We assume that the direct testing costs are constant and independent of the tested configuration (=c'). This is realistic when the cost is driven by the experimentation process rather than the experimentation subject. For example, in some athletic gear products, the designers conduct wind tunnel tests to refine a product's aerodynamic properties (*The Globe and Mail* 2004). These costs are substantial and are usually independent of the tested design configuration.[5] Furthermore, our constant cost assumption coincides with similar assumptions made in the extant literature (Loch et al. 2001). In §4, we offer a formal discussion of the limitations imposed by the constant cost assumption.

The second component of the experimentation cost, the opportunity cost from delayed product introductions, can be significant. Clark (1989, p. 1260) notes that "each day of delay in market introduction costs an automobile firm over $1 million in lost profits." We account for such losses as follows: a firm that tests k designs before terminating the test cycle incurs a delay loss $L(k)$. Using the total *effective* cost of the kth test $c_k = c' + L(k) - L(k-1)$ allows us to derive the optimal policy for general loss function structures (see the online appendix, provided in the e-companion).[6] This explicit loss function has *two* advantages over a discount factor: (i) we posit that factors, such as competition and the short product life cycle, affect the opportunity loss significantly more than the time value of money captured through a discount factor (Kavadias and Loch 2003); and (ii) using identical discounting for both the costs and the performance levels is misleading in an NPD context, as the risks associated with the two may be very different (for instance, technical risks versus market commercialization/acceptance risks).

Suppose that testing is sequential, and configurations 1 to k have been tested. Due to the performance dependencies, the outcomes of the tested configurations also convey information about the untested configurations. Proposition 1 outlines this dynamic information updating.

Proposition 1. *Assume that k configurations have been tested. The updated (marginal) performance distribution of an untested configuration i is normal with mean and standard deviation*

(i) $\hat{\mu}^i_{k+1} = \hat{\mu}^i_k + \theta(p_k - \hat{\mu}^k_k)/(k\theta + 1 - \theta) = \mu^i + (\theta/(k\theta + 1 - \theta))(\sum_{j=1}^{k}(p_j - \mu^j))$,

(ii) $\hat{\sigma}^i_{k+1} = \sigma\sqrt{1 - (k\theta^2/(k\theta + 1 - \theta))}$.

All the proofs are given in the online appendix. Given the learning and the evolution of knowledge outlined in Proposition 1, Theorem 1 presents the optimal testing policy for the base case.

Theorem 1. *Index the design configurations in descending order of their a priori expected performances, i.e., $i > j \Rightarrow \mu^i \leq \mu^j$. Define*

$$\chi^i_{k+1} = \left[\max_{j \leq k}\{p_j\} - \left(\mu^i + \frac{\theta}{k\theta + 1 - \theta}\left(\sum_{j=1}^{k}(p_j - \mu^j)\right)\right)\right].$$

Then, there exist constant thresholds Θ_i $(i = 1, \ldots, N)$ such that, after the completion of the kth configuration test, stop if and only if $\chi^i_{k+1} \geq \Theta_{k+1}$ for all $i > k$, otherwise test the design configuration with the largest expected performance.

An interesting insight emerges from the following interpretation of the optimal policy. The term $-\chi^i_{k+1}$ $(=\hat{\mu}^i_{k+1} - \max_{j \leq k}\{p_j\})$ expresses a normalized value that accounts for the remaining potential from testing, i.e., the difference between the expected performance of the next configuration and the best performance achieved so far. The term $-\Theta_{k+1}$ is a constant threshold. Thus, our policy has an intuitive economic interpretation: the design team terminates experimentation when the remaining potential drops below a threshold value (i.e., $-\chi^i_{k+1} \leq -\Theta_{k+1}$). If testing continues, the configuration with the largest remaining potential (or alternatively the design with the largest mean) is tested next.

Our optimal policy exhibits certain structural similarities with the Weitzman (1979) policy, adapted for our specific distributional assumptions to enable a meaningful comparison. Θ_{k+1} represents a normalized value that makes management indifferent between proceeding or stopping, a quantity very similar to the classic definition of a reservation price. Reinterpreting χ^i_{k+1} $(=p - \hat{\mu}^i_{k+1})$ as the normalized value obtained so far (normalized by accounting for the mean) allows us to also see the parallel between our policy and Weitzman's (1979) policy: testing stops when the normalized value χ^i_{k+1} obtained from testing exceeds a reservation price.

Our *choice rule*, i.e., which configuration to test next, is the same as Weitzman's rule. We can verify the claim as follows: consider two search opportunities $\mathbf{X} \sim N(\mu_x, \sigma)$ and $\mathbf{Y} \sim N(\mu_y, \sigma)$, and testing costs $c_x = c$ and $c_y = c$. The difference between the Weitzman reservation prices of $\mathbf{X}$ and $\mathbf{Y}$ is equal to the difference between their means, i.e.,

[5] "Before the 2003 Tour (de France), Armstrong spent about $15,000 an hour, for wind tunnel tests to create a new helmet" (*San Jose Mercury News* 2004).

[6] An electronic companion to this paper is available as part of the online version that can be found at http://mansci.journal.informs.org/.

$\mathrm{WRP}(\mathbf{X}) - \mathrm{WRP}(\mathbf{Y}) = \mu_x - \mu_y$.[7] Hence, the priority ranking induced by the reservation prices is identical to the one induced by the means, and Weitzman's (1979) adapted policy translates into testing the design with the highest mean.

However, our *stopping rule*, i.e., when experimentation should be terminated, exhibits key differences from Weitzman's rule. We rewrite our policy statement in Corollary 1 to enable easier comparison to Weitzman's (1979) stopping conditions.

Corollary 1. *Index the design configurations in descending order of the a priori performance expectations, i.e.,* $i > j \Rightarrow \mu^i \leq \mu^j$. *Define*

$$\Omega_{k+1} = \mu^{k+1} - \frac{\theta}{k\theta+1-\theta}\mu^k + \frac{\theta}{k\theta+1-\theta}\sum_{i=1}^{k-1}(p_i - \mu^i) + \Theta_{k+1},$$

and $\lambda_k = \theta/(k\theta+1-\theta)$. *After the kth test, optimally continue if and only if* $p_k \in ((\max_{i\leq(k-1)}\{p_i\} - \Omega_k)/\lambda_k, \Omega_k/(1-\lambda_k))$. *Otherwise terminate testing.*

Corollary 1 shows that accounting for performance dependencies changes the stopping rule. Continuation is optimal only when the current test outcome is within an interval. This result stands in contrast to Weitzman (1979), where continuation occurs only when the test outcome is below a threshold. Thus, the structure illustrated in Figure 1 generalizes. The key difference between the two policies is driven by the ability to learn during the experimentation process. Low test outcomes signal low future performances and, hence, decrease the attractiveness of additional experimentation. High design performances, on the other hand, signal a high remaining potential making additional experimentation worthwhile.

Our results add to the discussion initiated by Loch et al. (2001) concerning the impact of learning on experimentation strategies. Our stopping conditions emerge endogenously, and they involve balancing the performance benefits and learning against experimentation costs. The results also demonstrate that the stopping conditions depend on the design space structure and the realized performance levels as opposed to an exogenously prespecified confidence level.

In Proposition 2, we characterize the properties of the stopping thresholds Θ_k. They qualitatively describe the optimal test termination decisions of the design team.

[7] Let $\mathrm{WRP}(\mathbf{X}) = R_x$ and $\mathrm{WRP}(\mathbf{Y}) = R_y$. Then, by definition, $0 = \mathrm{E}[\max\{X, R_x\}] - R_x - c$ and $0 = \mathrm{E}[\max\{Y, R_y\}] - R_y - c$. Because $\mathbf{X}$ and $\mathbf{Y}$ are different only in their means, the distribution of $\mathbf{X}$ is identical to the distribution of $(\mathbf{Y} - \mu_y + \mu_x)$. Hence, $0 = \mathrm{E}[\max\{\mathbf{X}, R_x\}] - R_x - c = \mathrm{E}[\max\{\mathbf{Y} - \mu_y + \mu_x, R_x\}] - R_x - c = 0 = \mathrm{E}[\max\{\mathbf{Y}, R_x + \mu_y - \mu_x\}] - (R_x - \mu_y - \mu_x) - c$. That is, the reservation price of $\mathbf{Y}$ is equal to $R_x + \mu_y - \mu_x$, i.e., $R_y = R_x + \mu_y - \mu_x \Rightarrow (R_x - R_y) = (\mu_x - \mu_y)$.

Proposition 2. *Define* $\mathrm{WRP}_k = \{x\colon \mathrm{E}[\max\{x, X_k\}] - x = c\}$ *where* $X_k \sim N(0, \sigma^2(1 - k\theta^2/(k\theta+1-\theta)))$. WRP_k *expresses the Weitzman stopping threshold for search opportunity* X_k. *Then,*

(A) Θ_k *is a real constant, and is decreasing in* k.
(B) Θ_k *is increasing in* σ.
(C) Θ_k *is decreasing in* θ.
(D) $\Theta_k \geq \mathrm{WRP}_k$.

Claim (A) characterizes the normalized stopping thresholds, Θ_k, as decreasing in the number of tests conducted. This occurs for two reasons: (i) fewer design configurations remain to be tested and, hence, the learning possibilities are fewer, and (ii) the residual uncertainty is lower.

Claims (B) and (C) demonstrate the effect of the residual uncertainty on the optimal stopping values. Higher residual uncertainty may arise from fewer commonalities between design configurations (lower θ), greater complexity of the performance function (lower θ), or greater a priori uncertainty concerning the configuration performances (higher σ). Irrespective of the source, higher residual uncertainty makes testing more attractive by increasing the normalized reservation price. The result resembles the value-enhancing nature of uncertainty (variance) in a traditional option setting: because the design team is assured of retaining the highest performing design configuration upon stopping, the downside loss of conducting one more test is bounded by the testing cost. At the same time, the upside benefits from experimentation depends on the amount by which the performance can vary upward, which increases with higher residual uncertainty (variance).

Claim (D) demonstrates that the stopping thresholds in the optimal policy may be larger compared to Weitzman's (1979) optimal stopping thresholds WRP_k, which do not consider design correlations and learning. This result illustrates the relevance of learning in experimentation and highlights one of our key contributions. As opposed to myopic stopping proposed in past studies (Weitzman 1979, Loch et al. 2001), we find that testing may continue to provide information about the design space.

Our results stand in contrast to Adam (2001), who examines the search problem where the experimenter has priors about the distributions of the search alternatives and updates these priors as the search proceeds. However, he assumes that "learning exhibits enough monotonicity to prevent the searcher from optimally going through a 'payoff valley' to reach a 'payoff-mountain' later on" (Adam 2001, p. 264). Under this assumption, an extension of the Weitzman's (1979) policy, where stopping is still myopic, remains optimal. In contrast, we do not constrain the form of learning, and we find that optimal stopping becomes nonmyopic.

To summarize, experimentation offers *two* benefits: (i) the direct benefit of identifying configurations with higher performance (a key goal identified in the past NPD literature), and (ii) the indirect benefit from learning about the untested design configurations. Because of the second benefit, the design team may keep experimenting for the purpose of learning about the design space even when the remaining untested alternatives by themselves do not justify experimentation.

We illustrate the result with a modified version of the introductory numerical example. Consider the parameters:

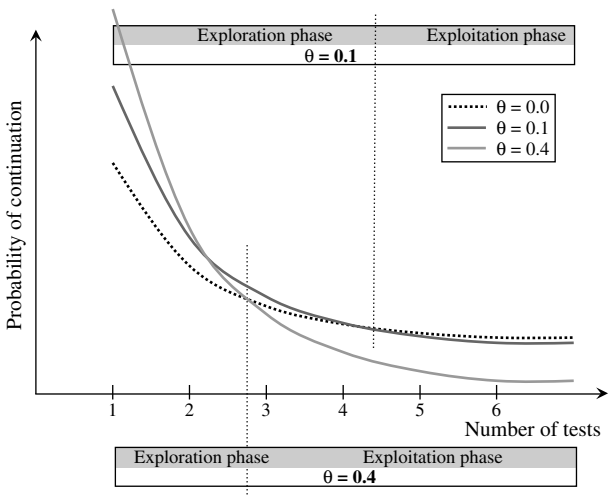
$\mu_A = \mu_B = -0.91$	$C_A = C_B = 0.1$
$\sigma_A = \sigma_B = 1$	$\theta = 0.1$

The expected one-step benefit from testing A ($=\mathrm{E}[\max\{0, \mathbf{A}\}] - C_A$) is 0. Thus, the myopic stopping rule (Weitzman 1979, Adam 2001) dictates immediate termination without testing any of the configurations. However, we can learn about the performance of configuration B from analyzing the test result of configuration A (say, $\theta = 0.1$). Hence, testing A and obtaining a test result a, prompts the design team to optimally test B if and only if the reservation price of B, $0.89 + (-0.91) + 0.1(a - (-0.91))$, is greater than the maximum obtained so far, $\max\{a, 0\}$.[8] Hence, testing B is optimal if and only if $a \in (-0.71, 0.079)$. Because there is a nonzero probability of testing B, this implies that the optimal policy involves testing A and then testing B if and only if $a \in (-0.71, 0.079)$.

Our result exhibits properties of "strong learning" (Adam 2001), that is, experimentation for the purpose of learning. Figure 2 illustrates this phenomenon. It plots the probability of continuation as a function of the number of tests conducted, for different levels of design performance dependencies.[9]

We observe that the dependencies across configurations (a higher correlation θ) lowers the likelihood of early termination, but increases the probability of later termination. Define the case with no dependencies (i.e., $\theta = 0$ or no learning) as the "benchmark curve." Then, we may call the phase with greater than the benchmark setting continuation probability the *exploration phase*, because in this phase, tests are conducted for the purpose of learning about the design space. Similarly, we call the phase with lower continuation probability the *exploitation phase*, where continuation, if pursued, is for the explicit purpose of obtaining a better performance. The precise transition from "exploration" to "exploitation" would be difficult to measure in practice, as the parameters are never precisely known. Still, an important qualitative insight for managers emerges: early on, conduct more "design-build-test" iterations than what the remaining alternatives seem to justify, in order to learn. Later on, conduct "design-build-test" iterations only to exploit the remaining promising alternatives.

Figure 2 Learning Effects Arising from Performance Dependencies



Our insight offers an explanation of earlier observations by Ward et al. (1995) and Sobek et al. (1999) about Toyota's product development process, where a large number of different design options are explored upfront before converging to a final solution. Toyota follows this approach "to explore broad regions of the design space simultaneously" (Ward et al. 1995, p. 49) and to "refine [their] map of the design space before making decisions" (Ward et al. 1995, p. 56). Although the mechanism of experimentation we examine in this paper is different (sequential as opposed to simultaneous testing), the underlying driver of the exploration phase is similar, namely, the potential to learn about the design space.

The following additional insights can be drawn from Figure 2: the likelihood of continuation during the initial tests and the duration of the exploration phase decrease in the complexity of the performance function and the distance between designs. This occurs for the following reason: the test outcomes hold greater information in design spaces where the complexity of the performance function is lower and/or where the distance between designs is lower. However, intensive testing continues for a shorter period of time as learning accumulates after fewer tests.

[8] The argument is identical as in §1.

[9] Figure 2 was generated for the specific problem instance with $N = 8$, $\sigma^2 = 80$, $c = 30$, $\mu_i = 140\ \forall\, i = 1, \ldots, 8$ and $\theta \in \{0, 0.1, 0.4\}$. For each parameter setting of θ, 10,000 simulation experiments were conducted and the probability of continuation after the kth test was calculated as the fraction of experiments where the optimal policy proceeds beyond the kth test.

To further build intuition about the drivers of the total number of tests, we perform comparative statics with respect to the design space characteristics. It is important, from a practitioner perspective, to outline such contingency-based guidelines (Loch et al. 2006). For ease of exposition, assume that the performance expectations of all the design configurations are identical ($=\mu$). Let $\mathcal{N}(\mu, \sigma)$ be the total number of tests conducted under the premises of the optimal policy, when the expected configuration performance is μ and the uncertainty is σ^2.

Proposition 3.

(i) *The number of tests increases in the initial expectations of design performance, i.e.,* $\partial\mathcal{N}(\mu, \sigma)/\partial\mu \geq 0$.

(ii) *The number of tests increases in the initial design performance uncertainty, i.e.,* $\partial\mathcal{N}(\mu, \sigma)/\partial\sigma \geq 0$.

Proposition 3 states formally that more experiments should be conducted on a design space where configurations have higher a priori performance expectations and higher performance uncertainty. These results confirm intuition, but they emerge for different reasons. High initial expectations lead to the design team foreseeing a higher potential value (as expressed by the $-\chi_k^i$ term), and thus result in additional testing. In contrast, higher initial performance uncertainty leads to increase in the stopping threshold (Θ_k), and leads to increased experimentation so as to reduce the (residual) uncertainty.

Next, we examine the impact of performance correlations (i.e., similarities between design configurations and the complexity of the performance function) on the number of experiments. The effect is not unidirectional. It depends on the performance realizations of the initially tested designs. Therefore, we perform a contingency analysis: we assume that k configurations have been tested and have yielded performances $p_1, \ldots, p_k$. The following definitions describe different contingency scenarios.

Definition 1. Suppose k design configurations have been tested yielding the following performances: $p_1, \ldots, p_k$. Then, we define the scenario (test outcomes) as

- *worse than expected* if the average configuration performance realized is lower than the average expected one, i.e., $\sum_{i=1}^{k} p_i/k < \sum_{i=1}^{k} \mu^i/k$;
- *better than expected* if the average configuration performance realized is higher than the average expected one, i.e., $\sum_{i=1}^{k} p_i/k > \sum_{i=1}^{k} \mu^i/k$.

Given these two contingencies, Proposition 4 describes the optimal actions:

Proposition 4. *Assume k configurations have been tested.*

(A) *If the test outcomes are worse than expected, the design team should transition to exploitation phase when the design performances are strongly correlated (high θ).*

(B) *If the test outcomes are better than expected, there is no unidirectional result.*

(C) *If the test outcomes are better than expected with a sufficiently high difference, the design team should continue exploration phase when the design performances are strongly correlated (high θ).*

Proposition 4 highlights the subtle moderating role of the performance correlations (due to the design space structure) on optimal contingency actions. Suppose that the test outcomes are, on average, poorer than initially expected. The information that these test results convey about the rest of the configurations depends on the underlying structure of the design space. When the performance correlations are greater, reflecting similar design configurations and a simple performance function, the initial test outcomes are more informative about the performances of the remaining configurations. Hence, the design team extrapolates the initial poor performances to the rest of the design space, and, consequently, transitions to the exploitation mode (i.e., increased likelihood of termination). On the other hand, if the performance correlations are weak, the initial poor test outcomes are not necessarily indicative of the remaining design performances. Consequently, the team may continue exploration even when worse-than-expected test outcomes are encountered at the early stages.

Claims (B) and (C) follow a similar (but more complex) line of reasoning. If the test results are better than expected, the design team uses the initial test outcomes as a signal that the remaining designs have higher than the (initially) expected performances. This signal is stronger when the design performances are more correlated, and, thus, results in continued exploration. At the same time, greater dependencies also result in faster learning (rapid uncertainty reduction) and an earlier transition to exploitation phase. Which of these opposing effects is stronger depends on the specific parameter values. When the realized performance levels are much better than expected, the effect of the "high performance potential" dominates, leading to continued exploration.

Table 1 summarizes the potential managerial contingency actions: After testing a few designs, the design team assesses the actual performance realizations and their difference from the initial expectations. If the test outcomes are below the a priori expectations, then the design team assesses that a worse-than-expected scenario has emerged. In the event that performance function is simple and/or the distance between design configurations is lesser, such initial bad news implies that there is not much more to expect from the design space at hand; thus, the team should most probably attempt to reduce the remaining experimentation cost by transitioning to exploitation phase. If, on the other hand, the performance

Table 1 Impact of Design-Space Structure on the Contingency Plan

	Much better than expected (Very good news)	Worse than expected (Bad news)
High correlation θ (Simple performance function less distance between designs)	Continue exploration Increased continuation probability	Transition to exploitation Lower continuation probability
Low correlation θ (Complex performance function greater distance between designs)	Transition to exploitation Lower continuation probability	Continue exploration Increased continuation probability

function is complex and/or the distance between design configurations is greater, then targeting reduction of the remaining experimentation costs is probably misplaced, as the initial bad news is not indicative of the remainder of the design space. Similarly, if test outcomes emerge significantly better than expected, then the design team should plan on further exploring the design space due to the high remaining potential. As before, this continued exploration is more likely when the initial good news is more indicative of the design space (i.e., when the performance function is simple and/or when the distance between designs is lesser).

The importance of Table 1 stems from the fact that it outlines guidelines for decision making and it demonstrates the nontrivial impact of the design space structure on the optimal test schedule. In that light, the contingency actions outlined may also be viewed as guidelines to achieve more reliable project monitoring and control (Loch et al. 2006).

3.2. Uncertain Performance Dependencies

Thus far, we have implicitly assumed that the design team can at least estimate the performance correlations between the design configurations. However, when the designs involve highly innovative technologies, the performance correlations arising from the configuration similarities may be unknown. In such cases, we assume that the design team specifies a likelihood for the performance correlations. In this section, we analyze the impact of uncertainty in performance correlations on optimal experimentation.[10]

The design team holds the belief that the performance correlations are weak (low θ_L) with probability p or strong (high θ_H) with probability $(1-p)$. The sequential experiments yield an additional signal $\mathcal{S}$, in addition to the design performance result. We assume that the signal S_k received after the kth test reveals the true correlation with probability λ, and it is independent from past performances $p_1, \ldots, p_k$. In other words, understanding the true performance correlations require additional information, such as exploring relevant scientific theories, and formulating abstract explanatory models of the design space. These signals are far more general than the performance outcome information; therefore, we assume that the dependence between S_k and the past performances is negligible. Theorem 2 presents the optimal policy.

THEOREM 2. *If the uncertainty regarding the performance correlations has been resolved, then use Theorem* 1. *Otherwise, index the design configurations in the descending order of their a priori expected performance, i.e., $i > j \Rightarrow \mu^i \leq \mu^j$. Define $p_{\max}^{k+1} = \max_{j \leq k}\{p_j\}$ as the highest realized configuration performance. There exist path dependent thresholds, $\Gamma_{k+1}(\sum_{j=1}^{k} p_j)$ $(i = 1, \ldots, N)$, such that after the kth test the design team should stop if and only if $p_{\max}^{k+1} \geq \Gamma_{k+1}$. Else they should proceed with the design configuration with the highest updated expected performance.*

The optimal policy shows marked similarity in structure to Theorem 1. However, the stopping thresholds are no longer path independent, and they depend on the sum of the realized configuration performances. Mirroring Corollary 1, the following corollary gives an alternative formulation of the optimal policy.

COROLLARY 2. *There exists thresholds τ_l and τ_h such that there is an optimal policy where after the kth test termination is optimal if $p_k \leq \tau_l$ or $p_k \geq \tau_h$.*

Corollary 2 suggests a surprising robustness of our initial results. Termination of the experimentation process occurs either because: (i) the test outcome is high enough so that the design team does not expect to receive a much higher value from additional experimentation ($p_k \geq \tau_h$); or (ii) the test outcome is so low that the design team expects the remaining designs to be low performing as well ($p_k \leq \tau_l$).

3.3. General Case: Multiple Design Classes

In this section, we extend our base case model to more general design space topographies. We consider that there exist distinct subclasses with stronger configuration dependencies within classes and weaker across. The new setup represents design problems that can be solved through substantially different approaches (e.g., recent efforts to substitute bar codes with radio-frequency identification tags imprinted on different materials like paper, polymers, etc; see PRC 2004).

Consider M distinct classes of design configurations. Each class j contains N_j different configurations. A priori, the design configurations of class j have unknown performances with identical[11] expectations (μ_j) and performance uncertainties $((\sigma_j)^2)$.

[10] We thank the referee team for suggesting this extension.

[11] We restrict the performance expectations to be identical within a class to reduce the notational burden. Extensions to a more general setting are straightforward.

In addition, the attributes that define a class (for instance the underlying technology) lead to significantly correlated performances within a class ($\theta = \theta_j$ within a class), and negligible dependencies across classes ($\theta = 0$ across classes).

Suppose the design team has sequentially tested k design configurations, Theorem 3 provides the dynamic optimal policy for the new setting.

THEOREM 3. *Define the surplus performance observed thus far as the following normalization, i.e.,* $\chi_j^{k+1} = [\max_{j \le k}\{p_j\} - \hat{\mu}_j^{k+1}]$. *Then, there exist constants* Θ_j^i $j = 1, \ldots, M$, $i = 1, \ldots, N_j$ *such that after the kth test the design team should stop if and only if* $\chi_j^{k+1} \ge \Theta_j^{k_j}$ *for all design classes, otherwise continue with the class that has the highest* $\Theta_j^{k_j} - \chi_j^{k+1}$.

Theorem 3 demonstrates that the stopping decision ("how many more experiments should we undertake") is relatively robust and is structurally similar to Theorem 1. Our policy permits an interesting analogy to Weitzman (1979), if we view the design subclasses as Weitzman's (1979) boxes. Under this interpretation, the optimal decision is equivalent to terminating the test cycle only if the best performance obtained so far is greater than the (updated) reservation prices and continuing with the box which has the highest (updated) reservation price. However, note that this analogy can only be taken so far because, in our multiple class setting, repeated sampling of a box (design class) is feasible, unlike in the one-trial-per-box case analyzed by Weitzman.

Also, similar to Corollary 1, it can be verified that the design team should stop testing the designs within a given subclass when an observed outcome from the subclass is either too high or too low (i.e., continuation only when the outcome is within an interval). Furthermore, the overall test cycle is terminated when any of the observed test outcomes is very high, because this would indicate that the likelihood of obtaining a still higher performance from any of the design classes is low, given the experimentation cost.

4. Discussion

This study examines the learning aspect of iterative testing cycles during the experimentation stage of a NPD process. We conceptualize each design configuration as a vector of attributes. These attributes jointly affect the final performance of each configuration in complex ways, due to possible coupling effects that are not fully known to the design team. Designs also may exhibit similarities with respect to their attributes, thus making some designs closer to others in the design space. Past experience and domain knowledge may allow the design team to approximate such a design space structure through a surrogate metric that accounts for both the complexity of the performance function and the distance between the designs: the correlation between design performances. We develop a normative model and we derive the optimal dynamic testing policy for design configurations that have different and uncertain performance levels.

Test continuation is optimal when the previous test outcomes range between two thresholds. Test outcomes below the lower threshold indicate an overall low-performing design space, where subsequent tests are unprofitable. Test outcomes above the upper threshold, on the other hand, indicate that the design team cannot expect a much higher value (than the highest value already found) by undertaking additional experiments. The upper threshold result is consistent with the optimal policy of Weitzman (1979). Furthermore, we also demonstrate that the design team may optimally pursues experimentation even after a "good" design is found, which, in the absence of performance correlations, would justify stopping. The potential to learn about the design space adds value to the testing process above and beyond the immediate performance benefits.

These results help us understand the role of the design space structure on learning and on optimal experimentation: less experiments with cost savings, when the information suggests that the untested design configurations may be performing low; or additional experimentation seeking higher performance outcomes, when the information indicates untapped potential.

Furthermore, we demonstrate that the consideration of learning splits the overall experimentation process into two distinct phases. An *exploration* phase that occurs at the beginning of the testing process and is characterized by a strong tendency to continue testing even when outcomes are good. During this phase, the "design-build-test-analyze" iterations allow the design team *to learn* about the remaining design options. We also find that greater correlations among the design performances enable faster learning and thus reduce the length of the exploration phase. Once the design team gains sufficient understanding about the design space, experimentation transitions into an *exploitation* phase. At that point, the likelihood of terminating testing and choosing the best design (found so far) increases. During this phase, the design team, based on what they *have learned*, conducts the "design-build-test-analyze" iterations only to test the remaining promising alternatives.

Finally, we analyze how the optimal continuation decisions change with the initial test outcomes. Our findings are summarized in Table 1, and they provide a useful guideline for contingency planning. When

the performance correlations of designs are high, poor test outcomes at the initial stages imply that there is not much more to expect from the remaining design space. Thus, the team should stop or transition to the exploitation phase to minimize the experimentation cost. If, on the other hand, the design correlations are low, then extrapolating from initial poor results is premature and exploration may continue.

Our policy exhibits a robust structure along two extensions: (i) the design team uncertainty about the performance dependencies (i.e., unknown correlation), and (ii) nonhomogeneous performance dependencies (i.e., a design space with distinct subclasses).

4.1. Limitations and Future Research

In our analytical model, we employed a set of assumptions in order to develop a qualitative understanding of the impact of the design space structure on learning and on the optimal amount of experimentation. However, three of these assumptions challenge the mathematical generality of our model and thus merit further discussion: [A1] normal distribution of design performances; [A2] equal testing cost within a class; and [A3] equal variance (uncertainty) in design performances within a class.

Theorem 4 offers an impossibility result that demonstrates that the three key assumptions (listed above) are not merely made for convenience, but are, in fact, necessary to achieve mathematical tractability.

Theorem 4. *Index policies are suboptimal if any one of the three assumptions*, A1, A2, *or* A3 *is relaxed to account for arbitrary structures*.

Theorem 4 demonstrates that the three key assumptions do indeed place our model at the boundaries of mathematical tractability (defined as the existence of an index policy). Despite of this inherent limit of generalizability, the model does offer an increased understanding of the key implications of design space structure on optimal experimentation. We consider our balance between realism and mathematical tractability to be a fair price to pay for developing this understanding. Loch et al. (2001) highlight the effect of problem parameters, such as test fidelity and the delay costs, on the total experimentation cost. Dahan and Mendelson (2001) examine the tradeoff between performance benefits and experimentation cost when experiments are run in parallel (thus eliminating any potential for learning). Our approach thus complements these past efforts by explicitly accounting for the design space structure and its effects on learning and on the amount of experimentation.

Finally, the limitations of the current conceptualization of the optimal experimentation and some of the emerging industry practices suggest a need for re-evaluating the search theoretic model as a whole. Firms are increasingly seeking innovation through collaborative efforts; such settings present challenges in understanding how firms manage distributed (or delegated) experimentation (Terwiesch and Loch 2004). At the same time, it is also imperative to recognize that experimentation and search are performed by individuals and, hence, are subject to behavioral traits (see Zwick et al. 2003, for a discussion of how certain behavioral elements affect sequential choices). Such extensions would require further field observation of how actual testing teams arrive at decisions/judgments during experimentation cycles.

5. Electronic Companion

An electronic companion to this paper is available as part of the online version that can be found at http://mansci.journal.informs.org/.

Acknowledgments

The authors thank Christoph Loch, Beril Toktay, the associate editor, and two anonymous reviewers for their valuable suggestions and comments that have greatly helped improve this paper.

References

Adam, K. 2001. Learning while searching for the best alternative. *J. Econom. Theory* **101** 252–280.

Allen, T. J. 1977. *Managing the Flow of Technology*. MIT Press, Cambridge, MA.

Clark, K. B. 1989. Project scope and project performance: The effects of parts and supplier strategy in product development. *Management Sci.* **35**(10) 1247–1263.

Clark, K. B., T. Fujimoto. 1989. Lead-time in automobile development: Explaining the japanese advantage. *J. Tech. Engrg. Management* **6** 25–58.

Cusumano, M., R. Selby. 1995. *Microsoft Secrets*. Free Press, New York.

Dahan, E., H. Mendelson. 2001. An extreme-value model of concept testing. *Management Sci.* **47**(1) 102–116.

DeGroot, M. H. 1968. Some problems of optimal stopping. *J. Roy. Statist. Soc. B.* **30** 108–122.

Ethiraj, S. K., D. Levinthal. 2004. Modularity and innovation in complex systems. *Management Sci.* **50**(2) 159–173.

Globe and Mail, The. 2004. Nike suits will make Canucks so swift. (June 24) S3.

Johnson, R. A., D. W. Wichern. 2002. *Applied Multivariate Statistical Analysis*, 5th ed. Prentice Hall, Upper Saddle River, NJ.

Kauffman, S. A. 1993. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, Oxford, New York.

Kavadias, S., C. Loch. 2003. Dynamic prioritization of projects at a scarce resource. *Production Oper. Management* **12** 433–444.

Loch, C., C. Terwiesch. 2007. Coordination and information exchange. C. H. Loch, S. Kavadias, eds. *Handbook of New Product Development Management*, Chap. 11. Elsevier-Butterworth/Heineman, Oxford, UK.

Loch, C., A. DeMeyer, M. Pich. 2006. *Managing the Unknown: A New Approach to Managing High Uncertainty and Risk in Projects*. John Wiley & Sons, London.

Loch, C. H., C. Terwiesch, S. Thomke. 2001. Parallel and sequential testing of design alternatives. *Management Sci.* **47**(5) 663–678.

PRC. 2004. Microsystems packaging research center: Leading the SOP and Nano paradigms in partnership with global industry. Georgia Tech Packaging Research Center, Atlanta.

Rencher, A. 2002. *Methods in Multivariate Analysis*. Wiley Series in Probability and Statistics, New York.

San Jose Mercury News. 2004. Technology has role in cyclist's training. (June 28).

Simon, H. A. 1969. *The Sciences of the Artificial*, 1st ed. MIT Press, Cambridge, MA.

Sobek, D., A. Ward, J. Liker. 1999. Toyota's principles of set-based concurrent engineering. *Sloan Management Rev.* **40** 67–83.

Stadler, P. F. 1992. Correlation in landscapes of combinatorial optimization problems. *Eur. Physical Lett.* **20** 479–482.

Terwiesch, C., C. H. Loch. 2004. Collaborative prototyping and pricing of custom-designed products. *Management Sci.* **50**(2) 145–158.

Thomke, S. H. 1998. Managing experimentation in the design of new products. *Management Sci.* **44**(6) 743–762.

Thomke, S. H. 2003. *Experimentation Matters*: *Unlocking the Potential of New Technologies for Innovation*. Harvard Business School Press, Cambridge, MA.

Thomke, S., M. Holzner, T. Gholami. 1999. The crash. *Sci. Amer.* **280**(3) 92–97.

Ward, A., J. Liker, J. Cristiano, D. Sobek. 1995. The second toyota paradox: How delaying decisions can make better cars faster. *Sloan Management Rev.* **36** 43–61.

Weitzman, M. L. 1979. Optimal search for the best alternative. *Econometrica* **47** 641–654.

Zwick, R., A. Rapoport, A. K. C. Lo, A. V. MuthuKrishnan. 2003. Consumer sequential search: Not enough or too much? *Marketing Sci.* **22**(4) 503–519.