Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Modularity and Innovation in Complex Systems

Sendil K. Ethiraj, Daniel Levinthal,

To cite this article:
Sendil K. Ethiraj, Daniel Levinthal, (2004) Modularity and Innovation in Complex Systems. Management Science 50(2):159-173. https://doi.org/10.1287/mnsc.1030.0145

MANAGEMENT SCIENCE
Vol. 50, No. 2, February 2004, pp. 159–173
ISSN 0025-1909 | EISSN 1526-5501 | 04 | 5002 | 0159

informs®
DOI 10.1287/mnsc.1030.0145
© 2004 INFORMS

Modularity and Innovation in Complex Systems

Sendil K. Ethiraj
University of Michigan Business School, 701 Tappan Street, University of Michigan, Ann Arbor, Michigan 48109, sendil@umich.edu

Daniel Levinthal
The Wharton School, University of Pennsylvania, Philadelphia, Pennsylvania 19104, levinthal@wharton.upenn.edu

The problem of designing, coordinating, and managing complex systems has been central to the management and organization literature. Recent writings have tended to offer modularity as at least a partial solution to this design problem. However, little attention has been paid to the problem of identifying what constitutes an appropriate modularization of a complex system. We develop a formal simulation model that allows us to carefully examine the dynamics of innovation and performance in complex systems. The model points to the trade-off between the destabilizing effects of overly refined modularization and the modest levels of search and a premature fixation on inferior designs that can result from excessive levels of integration. The analysis highlights an asymmetry in this trade-off, with excessively refined modules leading to cycling behavior and a lack of performance improvement. We discuss the implications of these arguments for product and organization design.

Key words: modularity; complex systems; innovation; nearly decomposable systems
History: Accepted by Linda Argote, former department editor; received April 24, 2002. This paper was with the authors 9 months for 3 revisions.

1. Introduction

The problem of designing, managing, and coordinating the myriad activities that make up large-scale complex systems such as economic institutions, products, or organizations has been central to the concerns of the management literature, as well as the social sciences more generally. Simon's (1962) essay on the architecture of complexity is a central building block in the analysis of the properties of complex systems and the articulation of general design principles for such systems. He argued that complex systems that are hierarchical and decomposable tend to evolve faster and toward stable, self-generating configurations. In recent years, the debate over modular versus integrated design logics has resurrected interest in Simon's early ideas (Baldwin and Clark 2000, Sanchez and Mahoney 1996).

There are two broad themes underlying the extant literature on modularity. The first addresses the contingencies under which modular design structures are favored over integrated ones (Alexander 1964, Baldwin and Clark 2000, Langlois 2002, Simon 1962, Ulrich and Eppinger 1999). Modular design structures are advocated as particularly useful when systems become so large and interdependencies between elements of the system so numerous that integrated design efforts become almost impossible (Parnas 1972, Simon 1962). In other words, modular designs are a useful means of managing complexity.

A second theme in the literature revolves around the "power of modularity," wherein the focus is on the advantages that modular design structures have over their integrated counterparts. In the product design context, it is argued that modular design structures are favored over integrated ones when flexibility and rapid innovation are more important than overall performance (Ulrich and Eppinger 1999). Modularization, if carried out properly, is expected to accelerate product innovation primarily through two mechanisms, autonomous (within component) and modular (mix and match of modules) innovation (Baldwin and Clark 2000). In an organizational context, it is argued that modularization of product designs can pave the way for similar modularization of organization designs, thus facilitating coordination of activities via an "information structure" rather than managerial authority or hierarchy (Sanchez and Mahoney 1996). Other authors have suggested that modular designs facilitate outsourcing via contracts and/or alliances of noncritical components or modules (e.g., Schilling and Steensma 2001) and thereby reduce the scope of activities that firms need to pursue in house. Still others have argued that modular designs help multibusiness firms "rapidly respond to altered business conditions by recombining diverse divisional resources and product-market domains" (Galunic and Eisenhardt 2001, p. 1244).

While extant research has yielded a great deal of understanding of the contingencies under which modularity is good and what benefits they provide, there is little systematic research on how decision makers partition designs into modules and what are the risks of partitioning incorrectly. If one side of the coin is that modular designs are the answer to managing complexity, then the flip side is to consider how "good or appropriate" modular designs may be realized in the face of complexity. The following description of Intel's Itanium chip design effort not only illustrates the difficulty of designing an appropriate modular architecture of a complex product system but also how much of a trial and error process it is:

> Developing Itanium, previously known by the code name Merced, has been an intense and unpredictable effort that sometimes teetered on the brink of disaster. Time and again, a project team of as many as 500 circuit engineers, chip architects, and software wizards found it has underestimated the difficulty of its task, more than once sinking into a quagmire of complexity with no obvious way out.... The team broke into separate groups, each working on one piece without knowing just how they would fit together.... The chip's architects had divided functions into separate modules, like letting teams of subcontractors design the rooms of a house. In mid-1996, Mr. Thomas slotted the modules together for the first time in what the team called the floor plan. Bad news: The floor plan was larger than anyone had expected, far too big to fit on a die of silicon that Intel could manufacture economically.... The team found itself sweating through a "die diet" as it worked feverishly to slim down bloated functions and subsystems.... But individual modules, initially only rough designs, kept growing larger as they were refined. After months of struggle, senior Intel managers realized that they could solve the size problem only with a radical step: a new manufacturing process that would let engineers shrink every wire and transistor.... The switch to a new fabrication process appeared to solve most of the Merced project's problems, at the cost of a few months of delay. But the project team soon found itself in a fresh predicament as they worked to tune up the movement of signals across the chip. In a well-functioning chip, signals flit from module to module in a precisely timed choreography, with the speed of the chip as a whole determined by the slowest signals. Merced engineers started looking for those slowpokes and found ways to speed them up via slight changes to the chip design. Soon, however, it became clear that many of these changes were disrupting the chip's delicate signal ballet, forcing engineers of other modules to rework their designs as well. The team found itself in a nightmarish world where a change to one module would ripple through the work of several hundred other people, leaving more problems in its wake (Hamilton 2001, p. 1).

The excerpt above makes it clear how difficult it is to achieve an appropriate modular design for an organization's innovative efforts with regard to a complex product system such as a microprocessor. This should not be surprising, as the problem of designing large-scale, interacting systems has been found to be NP complete (Chapman et al. 2001, Schaefer 1999). As Schaefer (1999) notes, "It would seem unlikely that a firm could ever hope to uncover an optimal modular design partition for a complex product" (p. 325). On the premise that "good" designs, modular or otherwise, are a product of evolution rather than foresight, it is important to understand the nature and significance of alternative design choices and their performance implications.

As a first step in examining these questions, we believe that there is a need for careful examination of the performance implications of over- and undermodularity, given a true underlying structure for a design problem. We attempt to meet this need by setting up a formal model that enables us to understand the evolutionary dynamics of complex systems. Our analysis points to critical trade-offs as the degree of modularity of a complex system is varied. On the one hand, modularity offers the advantage of parallelism (Marengo et al. 2000). However, given that systems are likely to be nearly but not fully decomposable (Simon 1962), fine-grained partitioning of decision problems is likely to result in efforts at improvement with respect to one partition perturbing the performance of other partitions. Moderate amounts of such self-perturbation[1] have a useful result of encouraging search and preventing premature lock-in to inferior designs. However, high levels of such self-perturbation inhibit the ability of a system to systematically improve and exploit the intelligence of prior search efforts.

We explore the implications of over- and undermodularity by carefully examining the influence, individually and in combination, of two key processes of design evolution: local search and recombination. We report two sets of analyses, the first of which systematically examines the benefits of modularity that the literature reports. This also forms the baseline to facilitate interpretation of the second set of results, where we examine the implications of over- and undermodularity given a true underlying structure.

In the first set of analyses, we assume that the designers have discovered the true underlying structure or decomposition of a decision problem, and we explore the interaction among the various processes of adaptation (local search, recombination based on module selection, and recombination based on firm-level selection). Consistent with the literature on modularity, we indeed find evidence for the powerful

[1] The term "self-perturbation" is used, because the performance landscape of individual partitions is perturbed not by some external shock but by adaptation efforts within the same system.

adaptive benefits of modular designs. Among the more nuanced results, we find a complementarity between recombination and local search, with recombination playing a powerful adaptive role in the early stages of design evolution and local search helping to obtain incremental performance improvements in later stages.

In the second set of analyses, we relax the assumption that designers have uncovered the true underlying structure and examine the performance implications of over- and undermodularity relative to the true underlying structure. We first explore the relationship between modularity (integration) and the rate of adaptation through local search. We show that there is an important asymmetry between the performance implications of over- and undermodularity. While excessive levels of integration can slow the pace of adaptation and lead to premature lock-in to an inferior outcome, excessive levels of modularity can, in the limit, stymie any possibility of adaptive change. Second, we expand the scope of the search process to include recombination in addition to local search to examine the robustness of the initial result regarding the asymmetry in adaptive performance of over- and undermodularity. We find that when both local search and recombination are present, we again see a greater performance penalty for over- rather than undermodularity.

The next section briefly reviews the literature on modularity and draws out the key assumptions that inform our formal model. Section 3 describes the basic structure of the formal model. In §4 we present the results. Finally, in §5 we discuss the implications of our results for modularity in products and organizations.

2. Complex Systems and Modularity

Simon (1962) defines a complex system as "one made up of a large number of parts that interact in a nonsimple way.... In such systems... given the properties of the parts and the laws of their interaction, it is not a trivial matter to infer the properties of the whole" (p. 468). The complexity here stems primarily from the often unknown nature and magnitude of interactions between different parts of the system and the consequent system performance implications. The nature of the interaction between two parts may be positive (increasing in one another), negative (decreasing in one another), or unrelated. Furthermore, the nature of the interaction may alternate between positive, negative, and unrelated over different ranges of interaction strength. As a result, overall system performance can exhibit highly nonlinear and/or nonmonotonic behavior in response to changes in one or more parts. The Itanium chip design effort described above clearly falls in the domain of the complex. The designers often discovered unknown interactions between different parts of the system and encountered unanticipated changes in performance. In such systems, ostensibly localized adaptation can have serious consequences for overall system performance.

In this paper, the term "complex system" is used to describe such a set of interdependent decisions. This conceptualization is general enough to include product design decisions, such as the Itanium chip above; organization design decisions, such as the division of labor between people, departments, or geographic locations; or broadly the firm itself and its organization of interdependent activities. It is also broadly consistent with a decision-theoretic model of organizations wherein information processing and decision making is decentralized among specialized agents (Radner 1993).

Modularity is a general set of design principles for managing the complexity of such large-scale interdependent systems. It involves breaking up the system into discrete chunks that communicate with each other through standardized interfaces or rules and specifications (Langlois 2002). Organizational structures can be seen as modular to a greater or lesser degree. Grouping activities by functions (unitary) or by products (M-form) and delineating the lines of authority seeks to minimize redundant information flow, reduce conflicts, and improve coordination. Clearly, for modularity to work as designed, it is important to adhere to some rules in creating modules.

Central to the notion of modularity is the distinction between interactions within subsystems and interactions between subsystems of complex systems (Simon 1962). When there are no interactions between subsystems, the system is fully decomposable. However, fully decomposable systems are rarely observed in nature, though near decomposability—where interactions between subsystems are significantly weaker than the interactions within subsystems—is not uncommon (Simon 1982, pp. 212–213). Drawing on Simon (1962), Baldwin and Clark (2000) define a module as "a unit whose structural elements are powerfully connected among themselves and relatively weakly connected to elements in other units" (p. 63). In addition, they argue that "When the complexity of one of the elements crosses a certain threshold, that complexity can be isolated by defining a separate abstraction that has a simple interface. The abstraction hides the complexity of the element; the interface indicates how the element interacts with the larger system" (Baldwin and Clark 2000, p. 64). Thus, the key goal of realizing modular designs is to group strongly interacting elements or parts together and separate weakly interacting ones.

Baldwin and Clark's (2000) discussion of modularity and design points to two senses in which the term "modularity" may be deployed. In the first, it can denote the "real" underlying structure for a given design problem, reflecting not only the partitioning and decomposition of the task but also the design of interfaces among the design elements. In this usage, terms such as under- and overmodularized make little sense.

In a second sense, if the search for the "appropriate" number of modules were trivial, then reserving the term "modular" for this ideal decomposition seems reasonable. However, as noted earlier, the problem of complex system design is not an exact science that will always result in an optimal solution. Not only are systems not fully decomposable, but also apt partitions that capture the nearly decomposable structure of complex systems are not self-evident. Designers are searching on the matrix of design parameters at four levels: (1) the "appropriate" number of modules; (2) the "appropriate" mapping of design elements to the modules; (3) the "appropriate" interactions among the design elements within each module; and (4) the "appropriate" interfaces or interactions between modules. Therefore, given the complexity of the design task—indeed its impossibility (Schaefer 1999)—in making guesses about the optimal number of modules, firms may err on the side of greater modularity or greater integration. Consistent with this second usage, we use the term "module" to refer to a particular partitioning and decomposition of a task and are agnostic as to whether any given modularization does or does not capture the true underlying structure of the complex system under consideration.

The literature suggests that modular designs confer several benefits such as reducing design and development time (Ulrich and Eppinger 1999), allowing parallelism in design and testing (Baldwin and Clark 2000, Loch et al. 2001), multiplying design options through mix and match of modules (Baldwin and Clark 2000), and allowing localized adaptation within hidden modules (Baldwin and Clark 2000). In the models that follow, we attempt to examine how the reported benefits vary with design choices that are over- or undermodularized.

3. Models

The model set-up for examining the performance implications of over- and undermodularity requires the specification of three features of the system: (1) the representation of the true underlying structure of the system and its performance landscape; (2) the means designers use to create modular representations of the true structure (i.e., design choices); and (3) the processes of adaptation on the landscape that allows the evaluation of different modular representations of the true structure. We elaborate each serially.

3.1. Modeling the True Underlying Structure of the System of Interdependent Choices

A firm, or more specifically, a firm's decision problem, which can include product design or organization design decisions, is represented as a row vector of N attributes or decision variables $[a_1, a_2, a_3, \ldots, a_n]$. Without loss of generality, in our model, each decision variable can take on two possible values (0 or 1). Thus, there is a set of 2^N possible organizational forms. For instance, use of group-based work organization might represent a setting of 1, and 0 otherwise. It follows that different settings for the decision variables of the organization have different performance implications. Continuing with the work organization example, choice of work organization is likely to have interactions with the incentive system, accounting methods and systems, quality of employee effort, physical layout, and so on. For instance, it is conceivable that group-based work organizations entail collective workspaces, which in turn requires better group-level monitoring and regulation of employee effort. In other words, some combinations of decision choices may yield performance improvements, while others may undermine it (Ichniowski et al. 1997, Macduffie 1995).

The performance landscapes underlying such a representation of firms tend to be rugged with several peaks and valleys, because tweaking a decision choice that results in subunit performance gains does not always lead to concomitant firm-level performance improvements because of the presence of interactions between decision choices. A stylized real-world analogue of such performance degradation is likely when an individual with no formal ties to a team is rewarded based on team performance. In this case, individual incentives are likely to stimulate higher performance. The performance of the firm depends on the settings (1's or 0's) of the decision variables and the interactions among them. When there are no interactions between decision variables, each decision makes an independent contribution to firm performance. As the interactions between decision variables increase, the contribution of each decision choice to the firm's performance becomes increasingly interdependent. The complexity of the firm stems primarily from the often unknown interactions as well as the functional form guiding the interactions among the decision variables. For this reason, the performance landscapes of such complex decision problems tend to be rugged and nonlinear, with actions on one decision

Figure 1 **Interaction Matrix of Decision Choices Within the Complex System**

	a_1	a_2	a_3	a_4	a_5	a_6	b_1	b_2	b_3	b_4	b_5	b_6	c_1	c_2	c_3	c_4	c_5	c_6
a_1		x	x	x	x	x												
a_2	x		x	x	x	x												
a_3	x	x		x	x	x												
a_4	x	x	x		x	x												
a_5	x	x	x	x		x												
a_6	x	x	x	x	x													
b_1						x		x	x	x	x	x						
b_2							x		x	x	x	x						
b_3							x	x		x	x	x						
b_4							x	x	x		x	x						
b_5							x	x	x	x		x						
b_6							x	x	x	x	x							
c_1												x		x	x	x	x	x
c_2													x		x	x	x	x
c_3													x	x		x	x	x
c_4													x	x	x		x	x
c_5													x	x	x	x		x
c_6													x	x	x	x	x	

choice having ripple effects on other decision choices (Levinthal 1997).[2]

Simon's (1962) exposition on the architecture of decomposable systems forms the basis for our modeling of the interaction patterns among the decision variables. Figure 1 illustrates an interaction matrix for a problem with 18 decision variables and three modules. This structure represents the true underlying modularity against which performance of over- and undermodularized structures are compared. In the figure, an alphanumeric notation represents each decision variable. The alphabetic portion denotes the module, while the numeric portion denotes the respective decision variable. The x's in each row-column intersection represent the presence of interdependence between decision variables. Reading across a row, an "x" indicates that the row variable is affected by the column variable. Conversely, reading down a column, an "x" indicates that the column variable affects the row variable. Therefore, x's positioned symmetrically above and below the principal diagonal represent reciprocal interdependence between decision variables.

Within each module, each decision variable is tightly coupled with the other decision variables in the same module, that is, reciprocal dependence (Thompson 1967). In addition, modules 2 and 3 have a weakly coupled interface relationship with the next-higher module, denoted by a single x below the principal diagonal. We see that the contribution of an individual variable, a_i, depends on other variables, as indicated in Figure 1. Thus, decision variable a_1 depends on decision variables a_2, a_3, and a_4. In contrast, decision variable b_1 depends on four other decision variables (a_4, b_2, b_3, b_4). As a result, decision variable a_1 can result in 16 (2^4) possible levels of performance, depending on its own value (0 or 1) and the value of the three other decision variables on which it depends, while decision variable b_1 can take on 32 possible levels of performance, depending on its own value and the value of the four other decision variables on which it depends.

Therefore, consistent with our discussion of modularity earlier, the interaction matrix in Figure 1: (1) is nearly decomposable; (2) indicates that the interaction within modules is stronger than the interaction between modules; (3) shows that most of the information within a module is encapsulated in the sense that changing four of the six decision variables will not affect the performance of other modules; and (4) shows that the interfaces are few and small (two decision variables each). In this sense, we have been faithful to the accepted principles of modularity in defining the real, underlying structure of the problem.

The performance contribution (ω_i) of each decision variable (a_i) is determined both by the state (0 or 1) of the ith decision choice and the states of the j other decision choices on which it depends. Thus,

$$\omega_i = \omega_i(a_i; a_i^1, a_i^2, \ldots, a_i^j).$$

The value of ω_i is treated as an i.i.d. random variable drawn from the uniform distribution[3] $U[0, 1]$ for each configuration of a_i and the j other decision choices on which it depends. Thus, for a firm with 18 decision variables there are 2^{18} distinct uniform random numbers associated with each firm configuration. Firm performance Ω is a simple average of the ω_i over the N decision choices:

$$\Omega = \frac{1}{N}\left[\sum_{i=1}^{N} \omega_i\Big(a_i; a_i^1, a_i^2, \ldots, a_i^j\Big)\right].$$

In addition to initializing the performance landscape as characterized above, the states $(0, 1)$ of the vector of decision variables a_i are drawn at random at the start of an experiment. Because any single run is

[2] Our characterization of a performance landscape here bears a close link to the work of Kauffman (1993). A critical distinction, though, lies in the fact that the systems that we explore have some unknown but inherent structure, while Kauffman examines purely random interaction structures. The inherent structure imposed by modularity serves to create a locally correlated landscape (Bar-Yam 1997) that is in contrast to a rugged landscape of random interactions explored in work by Kauffman (1993) and others (Levinthal 1997). Thus, the modeling is a blend of the conceptual ideas of Simon (1962) and the technical apparatus of Kauffman (1993).

[3] The results are robust to alternative distributional assumptions. The results with exponential and log-normal distributions are qualitatively similar to that reported here. These results are available from the authors.

sensitive to the inherent randomness in both the initial states of the decision variables and the initialization of the performance landscape, we replicated each experiment 100 times with different starting seeds for both the specification of the performance landscape and the starting state of the system to remove the stochastic component endemic to any single run. The reported results, unless noted otherwise, are averaged over 100 runs.

3.2. Modeling the Design Representation of the True Structure

In each of the experiments described here, a modular structure—that is, the number of partitions or modules that would guide design decisions—was specified. We determined module composition (i.e., the decision variable(s) that would be assigned to a design module) for a given value of N, such that the design modules were equal in size. More generally, for a system with N decision variables and M true modules, we created K design structures or modules s, where the kth module s_k comprised (N/K) decision variables. We assume each module to be equal in size[4] to simplify the decision problem into simply choosing K. Thus, the composition of each module s_k was determined according to the following rule:

$$s_k = [a_{(k-1)N/K+1}, \ldots, a_{kN/K}] \quad \text{for all } k \text{ where } k \le K.$$

To examine the implications of over- and undermodularity, we varied the values of K such that $K > M$ and $K < M$, respectively. Such a decomposition of a decision problem is akin to the partitioning of the Itanium chip design effort into a number of teams with roughly equal assignment of decision responsibilities. The representation is founded on the premise that designers do not have perfect knowledge of the true structure of the decision problem. They are making guesses about the correct partitioning, and these guesses may be classified as overmodularized ($K > M$) or undermodularized ($K < M$) with reference to the true structure (M).

3.3. Processes of Adaptation

The literature emphasizes two main benefits of modular designs. The first is the ability to engage in parallel and autonomous adaptation (i.e., local search) within modules, and the second is the ability to mix and match or recombine equivalent modules from different systems. We model both processes of adaptation with and without selection at the module and firm level, respectively, and examine their sensitivity to over- and undermodularization.

3.3.1. Local Search. Local search or module-level innovation describes within-module or team-level innovation attempts. One of the primary objectives of modular designs is to facilitate relatively simple local innovation attempts by each team. Continuing with the Itanium chip design example, each design team engaged in the optimization of the module for which it was responsible. Note that such local search efforts, though they enhance performance at the module level, do not necessarily do so at the system level. We saw from the description that several successful local search efforts indeed turned out to be globally dysfunctional.

We implement such forms of local search for module improvement as follows. In each period of the experiment, the actors within each module attempt to enhance the performance of their particular module. Actors are assumed to "see" the performance of their given module and can anticipate what incremental changes from the existing decision set would imply for module performance. Thus, adaptation occurs through a process of off-line, local search implemented simultaneously in each of the modules (Marengo et al. 2000, Rivkin and Siggelkow 2003).[5] Within each module, a decision choice is selected at random, and actors within each module evaluate the efficacy of flipping the decision choice $(0, 1)$ by the criterion of improvement in module performance. The innovation is implemented if there is a perceived performance increase in the module. On the other hand, if there is perceived degradation (or no improvement) in performance, then the innovation is discarded. Such innovation attempts occur in parallel in each of the modules.[6]

More formally, consider a decision choice $a_j \in M_\kappa$ that is flipped to a'_j. Then, if

$$\frac{1}{n_{M\kappa}} \sum_{n_{M\kappa}} \omega_i(a_i; a_{\neg(i,j)}, a'_j) > \frac{1}{n_{M\kappa}} \sum_{n_{M\kappa}} \omega_i(a_i; a_{\neg(i,j)}, a_j),$$

$$\text{then} \quad \begin{cases} M_\kappa = (a_{\neg j}, a'_j) & \text{otherwise,} \\ M_\kappa = (a_{\neg j}, a_j) \end{cases}$$

where a_i is the state of decision choice i in the module before the local search attempt, a'_i is the state of

[4] We also implemented modules that were unequal in size. However, this approach does not discriminate between incorrect assignment of decision variables to modules, and the incorrect choice of number of modules. Therefore, for our analysis we use the more controlled experiment of equal-sized modules.

[5] This capability, as suggested by Rivkin and Siggelkow (2003), may be a function of effective accounting systems that are able to facilitate the evaluation of module-level performance. Indeed, activity-based costing is widely deployed to track the performance of organizational subunits (Cooper and Kaplan 1992).

[6] We also implemented a sequential innovation regime where, in each period, a single module is randomly selected for innovation. The results of this experiment are qualitatively similar, though the process unfolds more slowly. We decided that the parallel innovation process is more appropriate, because it directly exploits the benefits of modularity.

decision choice i after the local search attempt, and n is the number of decision choices in the module. The equation shows that the outcome of the local search attempt is retained if the performance of the module is higher as a result. Such a process of local search has been a central tenet of behavioral models of organizational learning (Cyert and March 1963, March and Simon 1958, Nelson and Winter 1982, Vincenti 1990) and are widely used in the operations research literature as well, especially to tackle problems that are hard to solve optimally (Ahuja et al. 2002 provide an overview; see also Rayward-Smith et al. 1996).

3.3.2. Recombination. Recombination involves the substitution of one module with another. We explore three variants of recombination. First, we explored *recombination* in conjunction *with firm selection*, whereby one firm copies a randomly chosen equivalent module from another firm, with the likelihood of adopting modules from higher-performing firms being higher.[7] This is akin to imitating practices from firms that are observed to be high performers. Second, we explored *recombination with module selection*. Instead of a random exchange of modules between firms, we calculate the fitness of the candidate modules for exchange and replace the lower-performing module with the higher-performing module. This may be seen as intelligent recombination or "transfer of best practices" (Szulanski 1996), where information about performance of individual modules is available to the designer. This is particularly relevant in cases where a firm has multiple product teams that work on independent designs.[8] Finally, we explored *recombination* in conjunction *with both firm- and module-level selection*. In all cases, in each period, each firm has an opportunity to engage in recombination.

More formally, consider two functionally equivalent modules M_α^s and M_α^t (i.e., two models that contain the same set of decision variables) from two different firms, s and t, respectively. Then, recombination in period $p+1$ is represented as

$$M^s_{\alpha(p+1)} = M^t_{\alpha(p)}.$$

Recombination with firm selection was implemented as above with the condition that the target of replication (i.e., firm t) was selected based on its performance. Selection processes are modeled as being proportionate to fitness using the roulette wheel algorithm (Goldberg 1989). Each system is mapped to a roulette wheel such that each firm's slice is proportional to its fitness. A random number is drawn and the firm on the roulette wheel that spans this random number is selected. The probability that a firm will be selected equals its performance level divided by the sum of the performance of all systems in the population at that time. More formally,

$$p(s_i) = \frac{\Omega_i}{\sum_{i=1}^{S} \Omega_i},$$

where $p(s_i)$, the probability of selecting the ith firm, is given by the ratio of the performance, Ω_i, of the ith firm to the total performance of all S firms in the population. The cumulative probability is then computed as

$$P(s_i) = \sum_{j=1}^{i} p(s_j).$$

A total of S random numbers r_S distributed i.i.d. in the interval $[0, 1]$ are drawn and the firms with cumulative probabilities that span the random numbers are selected according to the following rule.[9]

$$P(s_{i-1}) \le r_S \le P(s_i).$$

Recombination in conjunction with selection at the module level was implemented as follows. If

$$\frac{1}{n_{M\alpha}} \sum_{n_{M\alpha}} \omega_i(a_i^s; a_{-i}^s) < \frac{1}{n_{M\alpha}} \sum_{n_{M\alpha}} \omega_i(a_i^t; a_{-i}^t), \quad \text{then } M_\alpha^s = M_\alpha^t,$$

where $n_{M\alpha}$ is the number of decision choices in each module.

4. Analysis

4.1. Experiment 1: Baseline Results

The formal model outlined above includes two evolutionary processes: local adaptation within modules and recombination with and without selection at the

[7] Our implementation of recombination with firm selection bears a similarity to the crossover operator employed in the genetic algorithm (GA) literature (see Holland 1992). However, there are important differences as well. First, whereas crossover involves the *exchange* of policy variables between two firms, recombination as we model it involves the *substitution* of a module from one firm with a module from another firm. Second, the number of decision choices for crossover is randomly chosen and there is no attempt to exchange only equivalent decision variables. The recombination implemented here substitutes only functionally equivalent modules. The net effect of these two differences is that, whereas crossover increases the policy variable diversity in the population, recombination has the opposite effect—it reduces the diversity of policy variables. Thus, the results of recombination reported here are not strictly comparable with the results from the GA literature.

[8] Intel, for instance, usually has multiple teams working on the next generation of its microprocessor. In such cases, one form of learning between teams occurs through the adoption of each other's designs.

[9] Note as is standard in models of evolutionary processes, a single system can be selected for replication multiple times (Wilson and Bossert 1971). This property is necessary if the distribution of forms is to change in a population.

module and firm level. To carefully examine the effects of over- and undermodularity, it is important to first understand the performance benefits or the power of modularity and the conditions under which they are realized. Toward this goal, the first set of experiments is designed to explicate the adaptive benefits of modularity without changing the variable of main interest, that is, the degree of modularity. Thus, in the set of experiments reported in this section, we assume that the designers have uncovered the true underlying modular structure, that is, the design structure maps perfectly with the true structure.

We draw out the implications of the search processes using a series of nested models with increasing levels of complexity. We sequentially model the following evolutionary processes, both individually and in combination: recombination (§3.3.2) and offline local search (§3.3.1).

All the experiments reported here, unless specified otherwise, model the interaction among 10 firms on a given landscape. Each of the 10 firms is represented by a row vector of 30 decision choices. The 10 firms have identical design structures, with the number of design partitions equaling the real underlying decomposition (i.e., five partitions). However, the initial settings (i.e., 1's and 0's) for the decision choices for each of the 10 firms are independently specified by random assignment. Figures 2 and 3 present the results for different models, based on the average of 100 independent runs of these populations of 10 organizations.

4.1.1. Recombination. Figure 2 presents a series of models that examine the value of recombination in conjunction with selection at the module and firm level, respectively. From Figure 2 we see that recombination with firm selection (thin, dotted line) surprisingly provides only marginal performance improvement. Recombination can potentially serve two purposes. First, it can provide a new starting point for processes of local search, especially when firms are stuck on local peaks (Kauffman 1993). Second, it can also create a new organizational form and in turn generate greater diversity for selection processes (Levinthal 1997). The first effect is not present here, because there is no local search in this initial analysis. The second effect is also muted, because recombination acts on a relatively modest population of 10 firms. With only 10 firms in the population, selection quite rapidly reduces the diversity of firms by picking the higher-performing ones. With the selection process rapidly reducing diversity, the subsequent random exchange of modules between homogeneous firms adds little to performance. We see in Figure 2 that as we expand the number of firms in the population from 10 to 100 (thick, straight line), performance is superior to that of the 10-firm model, thus amplifying the positive variation-creation role of recombination.

Modularity not only offers the possibility of recombination with firm selection, but also the possibility of substituting one apparently superior module with another. The thin straight line in Figure 2 plots the results of recombination with module-level selection without firm selection. Contrasting this model with recombination with firm selection (thin, dotted line in Figure 2) demonstrates the superiority of recombination with module selection over recombination with firm-level selection. Interestingly, however, recombination with module selection does worse as the number of firms in the population increases (thick, dotted line). The results from recombination with firm selection with 100 firms indicate that the value of

Figure 2 **Power of Modularity: Recombination with Module and Firm Selection** ($N = 30$, $M = 5$, 100 **Runs**)

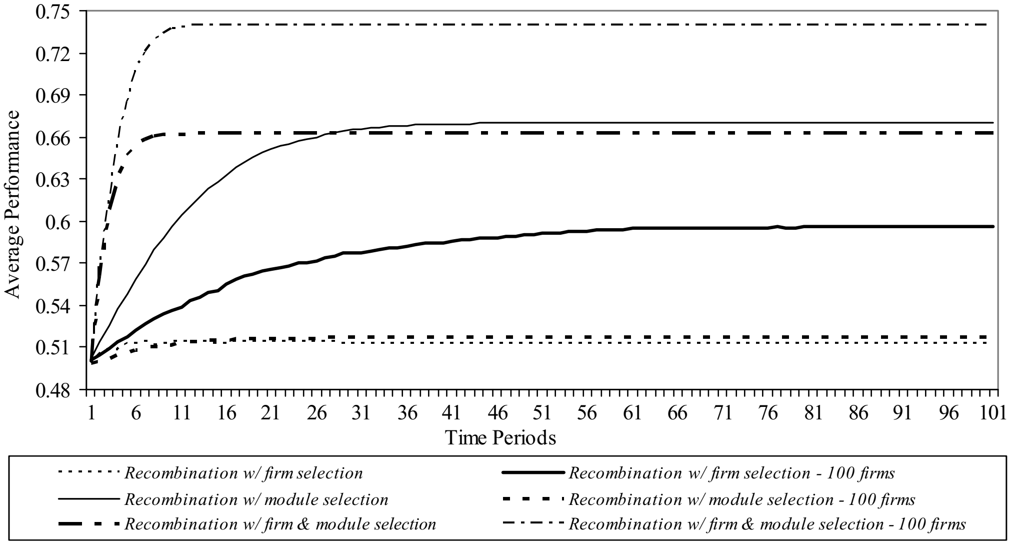
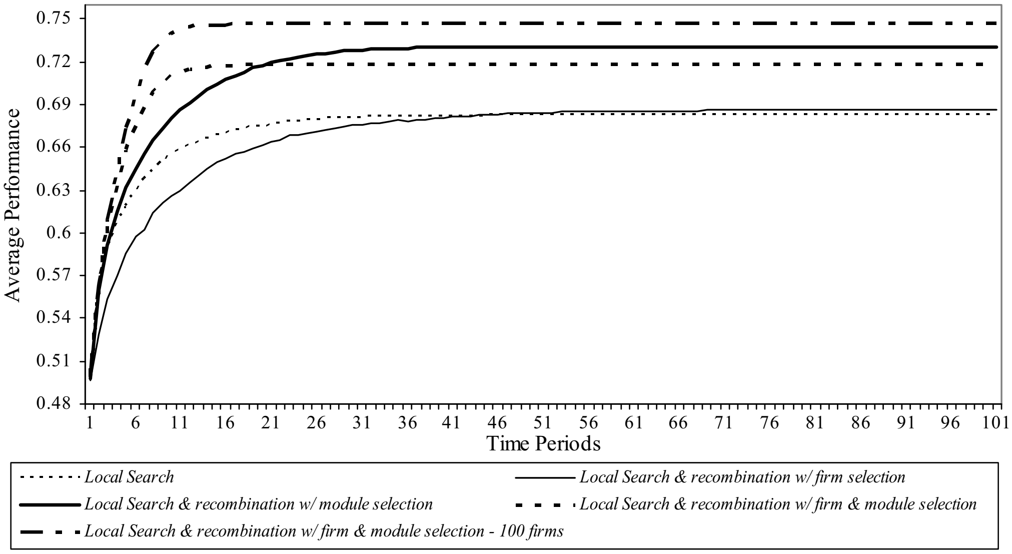


Figure 3 **Power of Modularity: Local Search and Recombination with Firm and Module Selection (*N* = 30, *M* = 5, 100 Runs)**



recombination, which contributes to firm adaptation by adding variety to the population, is enhanced by an increase in the size of the relevant population. However, in the case of recombination with module selection with 100 firms, the challenge of winnowing out inferior modules via module selection is compounded by the increase in the number of firms. The selection force at the module level essentially operates at the dyadic level, that is, comparing pairs of modules of two firms. Such dyadic selection performs well when the size of the population is relatively small (10 firms), but when the number of firms increases (100 firms), selection at the dyadic level is relatively weak and results in poor performance.

Finally, the contrast between recombination with firm and module selection (thick, dashed line in Figure 2) and recombination with module selection (thin, straight line) indicates that the introduction of firm selection provides few performance benefits to recombination with module selection alone. However, when the diversity of firms in the population increases, then the complementary nature of recombination and module and firm selection is again enhanced (see thin, dot-dash line).

4.1.2. Local Search. In Figure 3, the thin, dotted line presents the results of the local search process alone (see §3.3.1). Note that in the absence of selection or interaction among the 10 firms, the average performance of any single firm is unrelated to the number of firms in the population. We see that the process of local search yields steady system performance improvements with a rapid increase in the first 25 periods and a subsequent slowdown as the firms climb to local peaks on the landscape.

4.1.3. Relationship Among Adaptive Processes. The results from the models that include both processes of adaptation—local search and recombination—reveal important relationships among them (see Figure 3). First, we find that local search and recombination with firm selection (thin, straight line) does not perform significantly better than local search alone (thin, dotted line). Consistent with the results reported above, recombination with firm selection does not yield significant benefits when the diversity of firms in the population is low. This finding is replicated when local search is included in the model.

In contrast, local search and recombination with module selection (thick, straight line) outperforms local search (thin, dotted line) alone. There appear to be significant complementary benefits of recombination with module selection and local search. Whereas recombination enhances the rate of improvement in performance, generating steep increases in the initial periods, local search helps attain incremental improvements in the later periods of the experiment.

This complementary role of local search and recombination with module selection is robust to the inclusion of firm-level selection. This property is seen by contrasting local search and recombination with module selection (thick, straight line) and local search and recombination with module and firm selection (thick, dotted line). On the face of it, the inclusion of firm-level selection appears to be marginally dysfunctional. That is, recombination without firm selection obtains a higher level of performance than recombination with firm-level selection. In the latter case, the operation of selection on the 10 firms tends to prematurely destroy valuable module diversity in the population of firms. On the other hand, without firm selection this diversity is preserved, providing a richer and more varied context for the processes of recombination and local search to operate. However, as before, we see that as we expand the number of firms in the population to 100 (dashed line), firm-level selection again plays a useful and complementary role.

To summarize, in the first set of experiments we assumed that the designer has solved the problem of discovering the underlying decomposition of the firm's decision problem (i.e., design structure of five modules). Given this assumption, we sought to evaluate the conditions under which the benefits of modularity are realized. Our results on the different models suggest that local search alone is quite effective in improving firm performance over time. However, its benefits are incremental and take relatively long for fruition. Recombination, on the other hand, provides rapid and steep increases in performance, provided the designer can make informed decisions about the mixing and matching based on module-level performance information. We also find that firm performance information seems to be less useful when there are few firms from which to choose. However, when the number of firms increases, recombination with module selection alone tends to be an inefficient route to performance improvement. As the number of firms increase, it is more efficient to screen poor-performing firms without examining their module performance. As the number of candidate firms in the population increases, firm- and module-level performance information, respectively, are complements rather than substitutes.

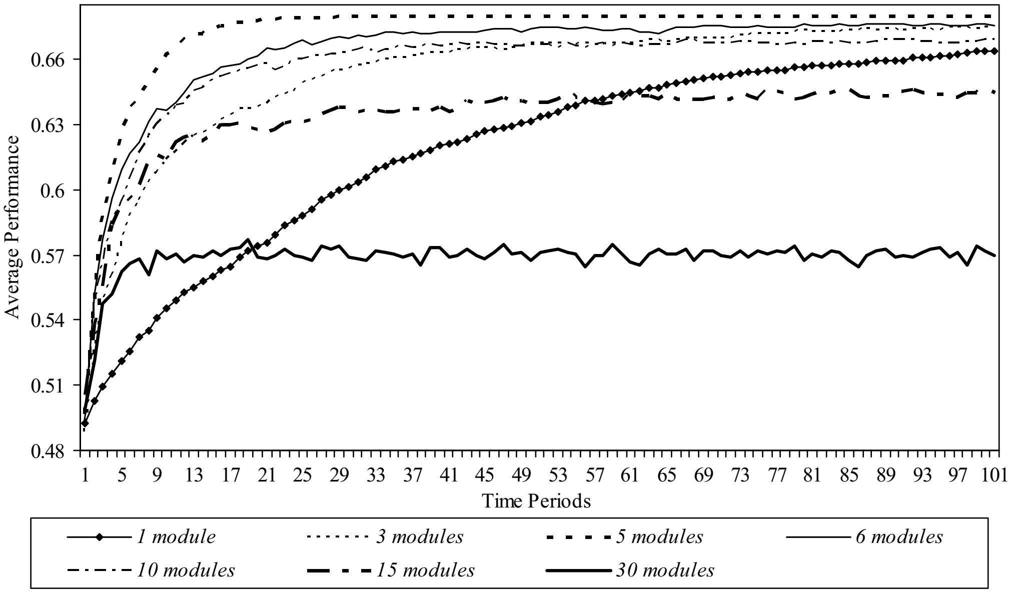
4.2. Experiment 2: Degree of Modularity and Effectiveness of Adaptation

Having examined the power and benefits of modularity and the conditions under which they are realized, we now turn to the central question of interest in this paper, that is, whether and how the degree of over- or undermodularity reinforces or undermines the benefits of modularity observed in the first experiment. We retain the same true structure as described in the first experiment, but relax the assumption that designers have uncovered the true structure. We set up several alternative design structures that vary along the continuum of over- and undermodularized and examine the implications of the search processes in each of the design structures. We sequentially report the results of the experiments, first with local search only and subsequently with local search, recombination, and selection in conjunction.

4.2.1. Local Search. We first explore the question of how the degree of modularity affects the rate and effectiveness of within-module local search. In other words, given the underlying real decomposition of the decision problem (five modules in the experiments described here), how does over- or undermodularization affect local adaptation outcomes and firm performance over time?

Figure 4 plots the average performance level of firms that vary in their degree of modularity over 100 periods. First, examining just the performance trends, we find that structures with less than or equal to five modules (i.e., the real decomposition) exhibit a monotonically increasing performance trend over time. On the other hand, for designs with greater than

Figure 4 **Local Search ($N = 30$, $M = 5$, 100)**



five modules, performance is nonmonotonic over time and the maximum performance level declines with increasing degree of modularity. At the extreme, when the number of modules is equal to the number of decision choices (i.e., 30 modules), performance over the simulated time period cycles up and down with little improvement in performance after period 10. Indeed, for an individual firm, we see a relatively dramatic, sine-like wave cycling up and down in performance that, in Figure 4, is "smoothed" through averaging across the 100 runs.

The monotonic performance increase in the case where the structure corresponds to the real decomposition is not surprising, because design decisions are nearly, though not fully, aligned with the performance outcome. The results on the over- and undermodularized structures are less self-evident. In the case of the undermodularized structures (i.e., less than five partitions), the gradual monotonic increase is driven by two considerations. First, and more straightforward, the rate of progress is gradual because larger modular structures allow for fewer parallel innovations in each period, that is, a two-module structure will attempt only two local innovations in each period as opposed to a four-module structure, which will attempt twice that number. Consequently, we observe a slower but increasing performance level over time for the undermodularized designs. Second, making innovation decisions on larger modules creates greater constraints on innovation because of the potentially larger number of interactions of each decision variable with other decision variables in the same module.[10] This not only reduces the frequency of successful local search, but also reduces the likelihood of adopting innovations that would reduce overall performance as a result of ignoring the presence of interactions among decision choices.

In the case of the overmodularized designs, the graphed lines in Figure 4 reflect a jagged performance curve over time. With greater modularity, designers tend to ignore the interactions among the decision choices. For instance, if a module of six decision choices is split into two modules, the innovation decision in each module is made contingent only on the interaction with two other decision choices rather than five. Ignoring these interactions results in innovations that may enhance local (within-module) performance but may degrade global performance. With increasing modularity, the performance degradation gets magnified by increasing the number of apparent local innovation opportunities in each period and increasing the likelihood that these innovations, while locally attractive, may be dysfunctional for the firm.

We carried out t tests[11] of differences in average performance over the 100 independent runs, comparing each of the modular design decisions with the real underlying decomposition (five modules). At the beginning of the simulation, there were no differences in the performance of the different design structures ($-1.177 < t < -0.058$). However, by period 25, there were significant differences between the performance of the five-module design and the over- and undermodularized designs ($1.739 < t < 17.265$). As expected, the differences were larger at the tails ($t = 17.265$ for modularity of 1 and 15.992 for modularity of 30). At the end of the 100th period, the five-module design asymptotes at an average performance level of 0.680. On the other hand, the fully integrated system (i.e., one module) reaches a performance of 0.663[12] by the 100th period, and the maximally modularized design (i.e., 30 modules) cycles between an average performance of 0.565 and 0.575. At the end of the simulation, the performance differences between the five-module design and the 10-, 15- and 30-module designs, respectively, are statistically significant ($t > 2.12$). The three-module and six-module design performances are statistically no different from that of the five-module design ($t < 0.89$), suggesting that small deviations from the real decomposition are less damaging.

We also examined the robustness of the results to changes in N (the number of decision variables), M (the number of true underlying modules), and K (the extent of over- and undermodularity). In particular, we implemented local search with $N = 48$ ($M = 6$, 12 and $2 \leq K \leq 24$) and $N = 60$ ($M = 10$, 12 and $2 \leq K \leq 30$). The results were quite similar to that with $N = 30$ and $M = 5$.[13]

In sum, the results of the first set of experiments suggest several general findings. First, there is a systematic relationship between degree of modularity and innovation outcomes. In particular, consistent with prior work (Marengo et al. 2000), the effectiveness of innovation is lower, or even disruptive, the greater the deviation of the design structure from the true underlying structure. Second, and more surprising, we find that, in the long run, erring on the side of greater integration poses lower performance penalties than erring on the side of greater modularity.

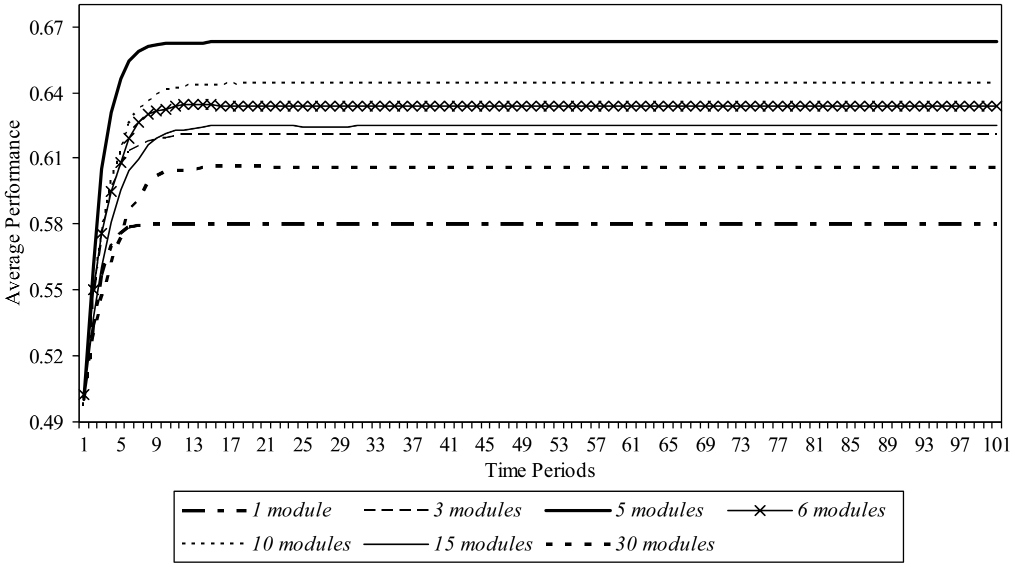
[10] A decision variable need not interact with all other variables within the same module. However, a larger module increases the likelihood of within-module interaction.

[11] The Scheffe tests, which correct for multiple comparisons, were also significant.

[12] The fully integrated design does not reach its asymptote by period 100, though its subsequent rate of progress is quite modest. By the 1,000th period, a performance level of 0.6847 is obtained.

[13] These results are not reported because of space constraints, but they are available from the authors.

Figure 5 **Recombination Without Local Search ($N = 30$, $M = 5$, 10 Firms, 100 Runs)**



Greater levels of modularity offer not only the advantages of higher levels of parallelism in search, but, in a world of imperfect modularization, also increase the degree to which innovation efforts in one module perturb the performance of other modules, often in dysfunctional ways. This cross-module perturbation is a variant of what Kauffman has termed "dancing landscapes" (Kauffman 1993, p. 243).

In moderation, this perturbation of the performance landscape of one partition by changes in another may facilitate long-run adaptiveness of the system by preventing it from locking into inferior local optima. However, at higher levels of modularization, and in cross-module perturbation, the system is unable to exploit effective configurations previously identified. The system tends to engage in excessive, and largely ineffective, search based on local perceptions of what constitutes progress.

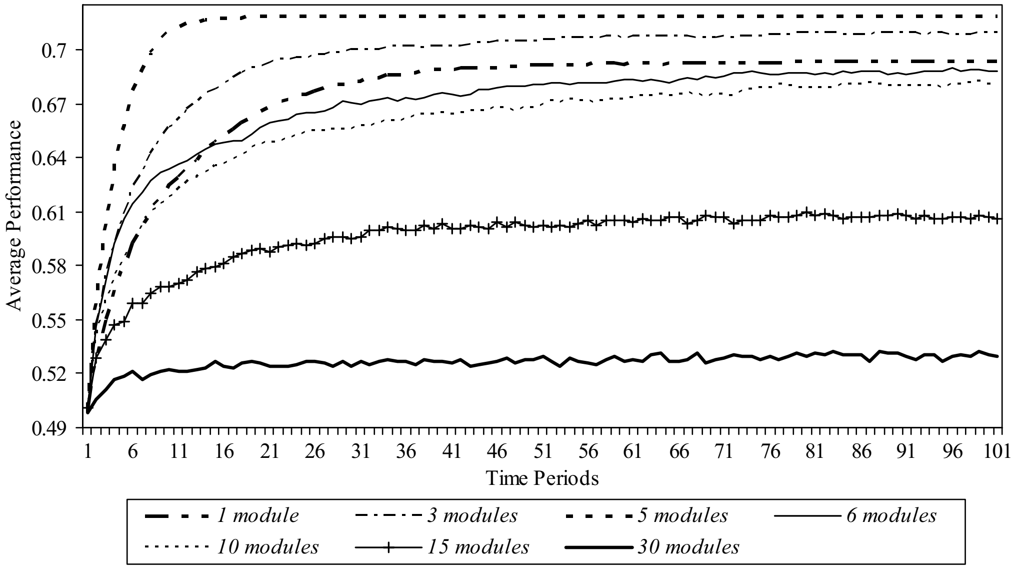
4.2.2. Local Search and Recombination. The experiment above explored the implications of varying degrees of modularity in the context of local adaptation. In the second set of analyses, we considered the effectiveness of recombination for modules that may not correspond to the true decomposition of the firm. A question remains as to whether the prior results regarding the dangers of overmodularity suggested by the initial analysis carry over when processes of recombination are incorporated as well.[14]

With recombination but no local search, we observe the opposite of our prior results—erring on the side of greater modularity appears to be preferred to erring on the side of undermodularity (see Figure 5). While the model structure with the correct decomposition (five modules) outperforms all models ($t > 2.00$), the performance degradation is higher in the case of undermodularized designs compared with the corresponding overmodularized designs. Recombination aids in the rapid increase in performance. By the 15th period, all models approach their maximum performance. The results suggest that selection-based recombination favors overmodularized rather than undermodularized designs. The process of recombination accounts for this result. Recombination helps firms avoid local peaks, because improvement in several decision choices occurs simultaneously. However, this also suggests that as the size of each module gets larger, a greater number of decision choices will get replaced and increase the likelihood of incorrect changes in decision choices. This reduces the chances of firm-level performance improvement.

However, introducing local search into this structure, as illustrated in Figure 6, demonstrates that the dangers of overmodularity identified in the initial analysis remain. As expected, the five-module design, corresponding to the true decomposition, outperforms all other design structures, asymptoting at 0.718. By the 100th period, the difference between the real decomposition and the overmodularized designs is statistically significant ($t > 2.40$), while the difference between the five-module and three-module designs is not significant. The 15-module and 30-module designs exhibit a small improvement in performance over the starting value, while the

[14] As in the analysis in §4.1, we again assessed robustness of the results by varying N (48 and 60), M (6 and 8), K (ranging from 2 to 30), and the number of firms in the population (10 and 100). The overall pattern of results was essentially the same, with the actual asymptotes being higher in the case of a population of 100 systems. These results are available from the authors.

Figure 6 Recombination with Local Search (*N* = 30, *M* = 5, 10 Firms, 100 Runs)



remaining design choices exhibit fairly similar patterns, asymptoting between 0.69 and 0.71.

This finding indicates an interesting interaction between the processes of local search and recombination. When the designs are undermodularized, there is a positive relationship between recombination and local search. Local search compensates for the relative ineffectiveness of recombination in the case of an undermodularized system. On the other hand, under very high levels of modularity, local search and recombination appear to be counteracting each other, with each process tending to reverse the innovative outcomes of the other. For module selection to be a useful mechanism of adaptation, the modules themselves need to be relatively fixed and interchangeable between firms. With local search in the context of small-sized modules, the process of change within the module and across modules via module selection conflict with each other. Thus, our initial finding of an asymmetric performance penalty for over- versus undermodularized systems is robust to the inclusion of the full set of adaptive processes.

To provide systematic evidence of performance differences between undermodularized and overmodularized designs, we performed two sets of regressions. First, we regressed performance in the 100th period in the analyses presented in Figure 6 against a dummy variable "UNDER," where the variable is coded 1 if it is undermodularized with respect to the true structure and coded 0 if it is overmodularized. The coefficient on "UNDER" was positive and significant ($\beta = 0.0751$; $t = 12.28$), suggesting that undermodularized designs have significantly higher performance than overmodularized designs.

In a second set of regressions, we sought to estimate the penalty (slope) for deviation in modularity. First, we regressed performance in the 100th period on "UNDER-MOD"—the deviation from the true structure for the undermodularized designs. Thus, the value of "UNDER-MOD" is −4 (one-module design), −2 (three-module design), or 0 (five-module design). The results indicate that a unit increase in undermodularity reduces performance by 0.0059. Similarly, we regressed performance in the 100th period on "OVER-MOD"—the deviation from the true structure for the overmodularized designs. Thus, the value of "OVER-MOD" is 25 (30-module design), 10 (15-module design), 5 (10-module design), or 0 (five-module design). The results indicate that a unit increase in overmodularity reduces performance by 0.0075.

If we assume that the two regressions are independent, we can compute the statistical significance of the difference in the two coefficients (difference in coefficients/difference in standard errors). Such a computation yields a t value of 2.34, thereby providing robust support for our argument about the asymmetric performance effects of over- and undermodularity.

5. Discussion

The analysis reported here has yielded several general findings. First, we find that given an unknown underlying decomposition of a system, designers are better off erring on the side of integration rather than greater modularity. This finding is robust to the introduction of module recombination between systems and is consistent with models of complexity theory that have shown systems to exhibit chaotic behavior when modularity is taken to the extreme (Kauffman 1995). This finding is also consistent with the experience of the Itanium chip design team described in the introduction. The design team began with an architecture that they believed was an appropriate modularization of the chip's functions. However, the presence of significant cross-module interdependencies hampered and disrupted the local module-level innovation and refinement efforts of the autonomous teams. The gains from parallelism seem to encounter a trade-off with the increased time spent in testing and integration of the system. The only circumstance under which overmodularity might be marginally preferred is when a firm decides to pursue innovation only through recombination and is not engaged in any local adaptation, which is clearly not the case in the Itanium example; more generally, instances of such behavior are difficult to identify.

Second, effective recombination requires complementary selection processes. Recombination carried out in the presence of information about module performance yields tangible gains. For recombination based on firm-level selection to be an effective source of adaptive change, ex ante heterogeneity in designs and modules is a necessary condition. In the absence of such variety, as our experiments demonstrate, recombination based on firm-level selection can destroy valuable diversity in the population of design choices and result in inferior performance outcomes. The interesting implication here is that when there is a limited diversity of available designs, there are significant performance benefits from recombining high-performing modules, even when the overall system or organization designs are not necessarily high performing. However, when there is a high level of heterogeneity in design choices, firm-level selection becomes a necessary condition for effective adaptation through recombination. Filtering high-performing designs as candidates for recombination generates significant performance benefits.

Translating these results into implications for product design, we can expect that in an internal selection environment (i.e., product development and innovation within an organization), the requisite high levels of heterogeneity may not be present. Organizations do not typically have more than a handful of teams working on alternative designs. In the case of Itanium chip project, both Intel and Hewlett-Packard bet on only one design. Given that the project outlay in this case was estimated to be $5 billion (Markoff and Lohr 2002), it is unrealistic to expect firms to pursue multiple system designs from which to select. Consequently, within the firm, the value of recombination may be limited. In a market context, we can expect a greater degree of system and module variety, and the possibility of realizing the benefits of recombination is higher. However, performance information at both the module and system level is a jointly necessary condition for realizing the benefits of recombination. For products such as the Personal Computer, where there are independent markets for modules (e.g., microprocessor, disk drive, etc.), the value of recombination is undisputed. However, for products such as the Windows® operating system, there are no independent markets for the individual modules, making it difficult to obtain module performance information. Thus, the value of market-based recombination is also contingent on the existence of markets for modules.

In sum, while we acknowledge the benefits of modularity, we caution against unabashed enthusiasm for ever-increasing degrees of modularity in design decisions. Modularity, like any design variable, involves trade-offs (Alexander 1964). Consistent with the presence of such trade-offs, Fleming and Sorenson (2001) show that intermediate levels of modularity tend to produce the most useful inventions. The apparent clarity of module-level innovative effort may mask, in dysfunctional ways, the broader system complexity. Excessive modularization may blind the designer to potentially important interactions between decision choices and result in dysfunctional perturbations in module- and system-level performance that constrain evolution to inferior designs. The speed and efficiency gains from modularization will be offset by the increased time spent in the testing and integration phase, where the consequences of ignored dependencies will come to the fore.

Designers engage in acts of creation, but unlike a divine creator, they lack omniscience. Choices of modules are guesses about appropriate decompositions—decompositions that even in reality are only partial (i.e., nearly decomposable). In making these guesses, our analysis suggests that there should be no presumption of a "promodularity" bias.

Ultimately, we need to model the evolution of modularity itself. We have begun to explore the trade-offs in making these choices, but a fuller examination of their dynamics is yet to be made. How do designers decide on the number of modules, and how do they map decision choices to modules? How do designers' independent choices affect the social interaction of products in the marketplace, that is, the ability to mix and match modules and the degree of module compatibility? In several aspects, our analysis is partial and incomplete. We examined the implications of modular design choices on innovation. However, modularity has other important implications for strategy, organizational coordination, incentives, and so on. Understanding the full implications of modular designs requires a better understanding of the trade-offs among these various objectives. We hope the present analysis helps us move along this trajectory.

Acknowledgments

The authors gratefully acknowledge financial support from the William and Phyllis Mack Center for Technological Innovation at The Wharton School. The authors thank Jan Rivkin, Nicolaj Siggelkow, Linda Argote, the associate editor, and three anonymous reviewers for detailed and helpful comments on an earlier draft of this paper. Errors and omissions remain our own.

References

Ahuja, R. K., O. Ergun, J. B. Orlin, A. P. Punnen. 2002. A survey of very large-scale neighborhood search techniques. *Discrete Appl. Math.* **123**(1–3) 75–102.

Alexander, C. 1964. *Notes on the Synthesis of Form*. Harvard University Press, Cambridge, MA.

Baldwin, C. Y., K. B. Clark. 2000. *Design Rules: The Power of Modularity*. MIT Press, Cambridge, MA.

Bar-Yam, Y. 1997. *Dynamics of Complex Systems*. Addison-Wesley, Reading, MA.

Chapman, W. L., J. Rozenblit, A. T. Bahill. 2001. System design is an NP-complete problem. *Systems Engrg.* **4** 222–229.

Cooper, R., R. S. Kaplan. 1992. Activity-based systems: Measuring the costs of resource usage. *Accounting Horizons* **6**(3) 1–13.

Cyert, R. M., J. G. March. 1963. *A Behavioral Theory of the Firm*. Prentice-Hall, Englewood Cliffs, NJ.

Fleming, L., O. Sorenson. 2001. Technology as a complex adaptive system: Evidence from patent data. *Res. Policy* **30** 1019–1039.

Galunic, D. C., K. M. Eisenhardt. 2001. Architectural innovation and modular corporate forms. *Acad. Management J.* **44**(6) 1229–1250.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman, Reading, MA.

Hamilton, D. P. 2001. Intel gambles it can move beyond the PC with new microprocessor. *The Wall Street Journal* (May 29) 1.

Holland, J. H. 1992. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, 2nd ed. MIT Press, Cambridge, MA.

Ichniowski, C., K. Shaw, G. Prennushi. 1997. The effects of human resource management practices on productivity: A study of steel finishing lines. *Amer. Econom. Rev.* **87**(3) 291–313.

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

Kauffman, S. A. 1995. *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity*. Oxford University Press, New York.

Langlois, R. 2002. Modularity in technology and organization. *J. Econom. Behavior Organ.* **49** 19–37.

Levinthal, D. A. 1997. Adaptation on rugged landscapes. *Management Sci.* **43**(7) 934–950.

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

Macduffie, J. P. 1995. Human resource bundles and manufacturing performance: Organizational logic and flexible production systems in the world auto industry. *Indust. Labor Relations Rev.* **48**(2) 197–221.

March, J. G., H. A. Simon. 1958. *Organizations*. John Wiley & Sons, New York.

Marengo, L., G. Dosi, P. Legrenzi, C. Pasquali. 2000. The structure of problem-solving knowledge and the structure of organizations. *Indust. Corporate Change* **9**(4) 757–788.

Markoff, J., S. Lohr. 2002. Intel's huge bet turns iffy. *New York Times* (September 29) 3.

Nelson, R., S. Winter. 1982. *An Evolutionary Theory of Economic Change*. Belknap Press, Cambridge, MA.

Parnas, D. L. 1972. On the criteria to be used in decomposing systems into modules. *Comm. ACM* **15**(12) 1053–1058.

Radner, R. 1993. The organization of decentralized information processing. *Econometrica* **61**(5) 1109–1146.

Rayward-Smith, V. J., I. H. Osman, C. R. Reeves, G. D. Smith. 1996. *Modern Heuristic Search Methods*. Wiley, Chichester, U.K.

Rivkin, J., N. Siggelkow. 2003. Balancing search and stability: Interdependencies among elements of organizational design. *Management Sci.* **49**(3) 290–321.

Sanchez, R., J. T. Mahoney. 1996. Modularity, flexibility, and knowledge management in product and organization design. *Strategic Management J.* **17**(Winter) 63–76.

Schaefer, S. 1999. Product design partitions with complementary components. *J. Econom. Behavior Organ.* **38**(3) 311–330.

Schilling, M., K. H. Steensma. 2001. The use of modular organizational forms: An industry-level analysis. *Acad. Management J.* **44**(6) 1149–1168.

Simon, H. A. 1962. The architecture of complexity. *Proc. Amer. Philos. Soc.* **106** 467–482.

Simon, H. A. 1982. *The Sciences of the Artificial*, 2nd ed. MIT Press, Cambridge, MA.

Szulanski, G. 1996. Exploring internal stickiness: Impediments to the transfer of best practice within the firm. *Strategic Management J.* **17**(Winter) 27–43.

Thompson, J. D. 1967. *Organizations in Action*. McGraw-Hill, New York.

Ulrich, K. T., S. D. Eppinger. 1999. *Product Design and Development*, 2nd ed. McGraw-Hill, New York.

Vincenti, W. G. 1990. *What Engineers Know and How They Know It: Analytical Studies from Aeronautical History*. Johns Hopkins University Press, Baltimore, MD.

Wilson, E., W. Bossert. 1971. *A Primer of Population Biology*. Sinauer Associates Inc., Sunderland, MA.