INSTITUTE
OF ECONOMICS

Scuola Superiore
Sant'Anna

LEM | Laboratory of Economics and Management

Institute of Economics
Scuola Superiore Sant'Anna

Piazza Martiri della Libertà, 33 - 56127 Pisa, Italy
ph. +39 050 88.33.43
institute.economics@sssup.it

LEM
WORKING PAPER SERIES

Search and Performance in Ecosystems: The Changing Role of Product Architectures

Axel Zeijen [a]
Luigi Marengo [b]
Stefano Brusoni [a]

[a] ETH Zurich, Switzerland.
[b] LUISS, Roma, Italy.

2023/16 April 2023
ISSN(ONLINE) 2284-0400

Search and Performance in Ecosystems:

The Changing Role of Product Architectures

Axel Zeijen
ETH Zurich
azeijen@ethz.ch

Luigi Marengo
LUISS
lmarengo@luiss.it

Stefano Brusoni
ETH Zurich
sbrusoni@ethz.ch

ABSTRACT

A crucial assumption in organization theory is that product architectures form a stable basis on which firms make strategic choices, over a period of time. However, emerging digital technologies challenge this idea, by allowing firms to redesign architectures at will. In this paper, we explore this novel phenomenon, its effects, and its theoretical implications. We develop an NK model suitable for studying (i) variable interdependence structures between components and (ii) the dynamics of search and adaptation in ecosystems. We find that the possibility to redesign product architectures undercuts the stability on which vertical relationships are based. We distinguish two pathways through which firms can benefit from redesigning product architectures: by enhancing the fitness landscape (landscape redesigns) or by altering the conditions on which inter-firm coordination is based (ecosystem redesigns). The availability of these two pathways depends on a firm's positioning (vertical scope and location in the value chain). Our results shed light on the changing role of interdependence structures in ecosystems, the differential advantages of integration and specialization strategies, and the effects of digital technologies in both technical and organizational domains.

INTRODUCTION

In recent years, research on business and innovation ecosystems (e.g., Adner and Kapoor, 2010) has given new impetus to the joint analysis of technical and organizational change. While their interplay has been a core area of interest in the analysis of industries, the economy, and society since at least Adam Smith and Karl Marx, the dynamics of technologies and organizations have often been studied separately, if only for simplicity. Yet, in recent years, there has been increasing interest in analyzing how technical and organizational systems co-evolve. For example, Davis, Eisenhardt, and Bingham (2009) analyze the varying importance of structure in the presence of different levels of market dynamism. Kapoor (2013) finds that firms can successfully remain integrated—even when the industry as a whole becomes disintegrated—if they make innovation and transaction choices that build off the benefits of integration. Hannah and Eisenhardt (2018) describe a strategy where firms continually reposition themselves within a value system, as they seek to occupy the "bottleneck" in the industry (Baldwin, 2018). Conti, Gambardella, and Novelli study the conditions under which firms may decide to specialize in upstream general-purpose components (2019a), and the conditions under which this strategy is sustainable in the long term (2019b). More generally, Eggers and Park (2018) discuss the conditions that enables incumbents to successfully respond to technical change.

Why do we see this emerging interest right now in positioning decisions? We argue that digital technologies are challenging a commonly held belief, grounded in theory and observation: industries and enterprises tend to organize and strategize on the basis of relatively stable technical structures—product architectures—that define the fundamental design rules (Ulrich, 1995; Baldwin and Clark, 2000) that attribute specific functions to components and set the interfaces among them. Industry life cycle theory (Utterback and Abernathy, 1975; Abernathy and Utterback, 1978) is built on the observation that within industries as diverse as automotive, energy, aviation, software, machine tools, etc., one or very few commonly accepted product architectures usually become dominant and remain so long-term. There is of course a phase of emergence and competition between different product architectures, yet once one emerges as dominant it stays stable for a long time and firms use them as templates for organizing, both internally and externally. This is why, as noted long ago by Henderson and Clark (1990), incumbents are in peril of being replaced by entrants when architectural innovation happens.

Digital technologies may open the door to the possibility of continuously transforming the very architecture of products. For instance, General Electric ("GE") used 3D printing technology to redesign its Advanced Turboprop (ATP) airplane engines—objects that had been architecturally stable since Rolls-Royce introduced its three-shaft engine in 1977. In 2015, GE introduced an engine in which 12 3D-printed components replaced 855 traditional ones. A fuel nozzle tip went from having 20 components to having just one. Changes like these are not merely technical; they have implications for firms' boundaries and product strategies, as well as positioning and value appropriation within their ecosystems. In this paper, we use the term "ecosystem" to capture the space within which firms coordinate to jointly produce a set of necessary components for an overall product system.

The promise of many emerging digital technologies, of which 3D printing is but one example, is that product architectures might become a strategic variable rather than a constraint. This new phenomenon, still in its infancy, requires us to revisit old assumptions about the sources, significance, and consequences of stability. To explore them, we ask: *How does the plasticity of product architectures impact the evolution of innovation ecosystems?* In particular, we are interested in analyzing ecosystem-level patterns of coordination, firm-level degrees of vertical integration and performance, and innovation-related benefits.

This question is important because it focuses in a novel way on the core, distinguishing feature of digital technologies, relative to previous waves of technical change:

"digital technologies [are] inherently dynamic and malleable" (Yoo et al., 2012: 1399). As noted, however, by Cennamo and Santaló (2019: 617), "[…] ecosystems are new forms of organizing independent actors' innovations around a *stable product system*" (italics added). We argue that, at least in some industries, the assumption of a stable underpinning product system might be elusive and exaggerated. Our approach, which builds on trends in several industries exemplified by the GE example above, suggests that this inherent stability in product systems is an assumption waiting to be challenged. Things are changing more quickly in practice than in theory, and we lack the analytical tools to study the evolution of ecosystems characterized by architectural fluidity.

To address our core question, two analytical steps are necessary. First, we need to consider firms that search simultaneously in the organizational and technical domains. Second, we need a way to loosen the assumption that firms are unable to change technical architectures endogenously. We formalize and explore our ideas with a new class of simulation models that enable us to operationalize these two analytical steps. We rely on a simulation model because the trends we study are still in their infancy and comparable data are thus not available, yet sufficient theoretical insight exists to develop a forward-looking model to study their (stylized) effects (e.g., March, 1991; Davis, Eisenhardt, and Bingham, 2007, 2009). We generalize the NK model (Kauffman, 1993; Levinthal, 1997; Gavetti and Levinthal, 2000), which has been extensively used to study search and adaptation in fitness landscapes (Baumann, Schmidt, and Stieglitz, 2019). We build upon and extend recent contributions to this literature that have begun to explicitly consider interdependences across firms (Ganco, Kapoor, and Lee, 2020), the nature of complementarities between system components (Rahmandad, 2019), and architectural change (Albert and Siggelkow, 2022).[1] More precisely, we deploy a two-pronged modeling strategy. First, we make explicit the

[1] The model we develop is in ways a generalization of each of these models, which we then apply to our research question.

interaction between technical and organizational landscapes, and model firms that explore both at the same time. We show how the possibility of this joint search process drives patterns of specialization and performance in a baseline industry NK model**.** Second, firms in our model are able to change technical architectures, through a process we call "redesign." We study what happens when firms use redesigns, both at the level of whole ecosystems (scope, performance) and at the level of firms (who benefits and why).

By modeling the mechanism through which firms may redesign product architectures as well as their ecosystem, we interpret and systematize a number of intuitions related to the impact of digitalization on firms' competitive strategies. First, we model different ways in which digital technologies may be leveraged by firms. In particular, we find that different types of firms may benefit from different "redesign" choices in different ways, depending on their scope and positioning. This is important, because extant research has produced conflicting predictions about the ultimate impact of digitalization processes. Our model explains that outcomes depend on firm-level features (e.g., vertical positioning, specialization levels) and, in so doing, our model provides guidance for future empirical work.

Second, we show that the benefits of innovation accrue disproportionately to those (few) firms that introduce it, at the expense of all others. We model two different types of "redesign" mechanisms: landscape redesign (which improves the landscape in which firms search) and ecosystem redesign (which allows firms to leverage different product architectures while simultaneously repositioning themselves in the ecosystem). This latter kind of redesign is less studied (e.g., Albert and Siggelkow, 2022, focus only on the former), brings potentially greater advantages, and is available mostly to downstream and integrated firms. We show that ecosystem redesign generates instability on the ecosystem level, as it provides the option of escaping bottlenecks for those firms that introduce such changes.

Previous work (e.g., Ganco, Kapoor, and Lee, 2020) did not consider this important feature, and remained focused on the analysis of fitness in a predefined landscape.

Lastly, we propose our generalization of the NK model as an important methodological contribution, as it allows us to endogenize core features of product architectures and the structure of interdependences, and also to jointly analyze technical and organizational structures. By turning interdependences into continuous variables, rather than binary and random attributes, we add an important dimension to the study of complex systems.

PRODUCT AND INDUSTRY ARCHITECTURES: A GENERALIZED NK APPROACH

Product architectures serve as the template upon which industries stabilize their vertical structures and firms organize internally. They might change endogenously over time (Fine, 1998; Schilling, 2000; Jacobides, MacDuffie, and Tae, 2016; Baldwin and Clark, 2000) as a result of industry-level forces. They may also change discontinuously and exogenously as one technology generation succeeds another, with invasive impacts on incumbent firms (Henderson and Clark, 1990). Wolter and Veloso (2008) noted that architectural innovation, because of its inherent technical and organizational complexity, poses the most "interesting questions" about vertical boundaries (2008: 599), as it can confer advantages in an established market or even open up new market segments. Yet, they added, following Teece (2000), that this type of innovation is also the least common.

Industry lifecycle theory sees the emergence of dominant designs as tipping points in industrial dynamics (Anderson and Tushman, 1990; Utterback and Suarez, 1993). Such dominant designs fix the overall architecture of a technical system (Christensen, Suarez, and

Utterback, 1998; Murmann and Frenken, 2006), allowing rapid changes at the subsystem level. More recent examples from the literature on innovation ecosystems continue to build on the core–periphery division. The differential rate of change between core and periphery is important, and indeed pivotal to the ecosystem challenges to be managed by focal firms (Adner and Kapoor, 2010), but the structure of interdependences itself is assumed to be stable (Ganco, Kapoor, and Lee, 2020). Besides serving as a guideline for coordination, product architectures also shape relative power between segments of an industry architecture. Jacobides, MacDuffie, and Tae (2016), for example, showed how the process of modularization and outsourcing in the automotive industry threatened to shift power away from car manufacturers towards suppliers.

There may also be important effects on relative firm performance. While some architecture changes may be executed with the agreement of all firms involved (assuming a win-win situation), there might be important asymmetries involved. Baldwin (2018) argues that firms may possess “dynamic architectural capabilities” to “understand a large technical system coherently and change it in ways that are competitively advantageous” (2018: 25). However, successful examples of such strategic product-architecture changes are still few and far between, making systematic analysis challenging. For example, Brusoni, Prencipe and Pavitt (2001) studied the capability patterns of systems integrators (those firms that know more than they do) in the civil aviation industry have to maintain in order to be ready to respond to possible architectural challenges.

In order to model the process that enables some firms to redesign product architectures requires an intermediate step. We need a modeling instrument that explicitly accounts for the co-evolution of technological and organizational landscapes. Our modeling strategy aims to capture four critical dimensions: how the product is designed, how industries organize for production and innovation, how a firm's position in an existing industry architecture enables different search strategies, and how firms perform within an industry architecture.

NK models originated in biology to analyze the properties of selection when it operates on complex structures (genotype-phenotype mappings). However, they have been increasingly used in the management literature to analyze processes of boundedly rational search and adaptation. We refer the interested reader to the excellent and comprehensive survey in Baumann, Schmidt, and Stieglitz (2019), and to Csaszar (2018) for a guide to the methodology. Here we review the main features and applications of these models only briefly, to elucidate how we extend them.

NK models address two fundamental issues in search and adaptation in complex systems. First, search typically takes place in large combinatorial spaces that humans cannot fully explore, due to bounded rationality. Thus, search and selection procedures should be evaluated on the grounds of their ability to locate good portions of the search space. Second, organizations and technologies have internal structures that constrain variation mechanisms, i.e., the generation of new solutions or entities to be tested. Therefore, since selection can only operate on the material provided by variation, the latter may contribute to determine the evolutionary paths more than selection itself: internal constraints (interdependences among components) may determine a rugged selection landscape where selection alone produces early lock-in into the nearest local peak. In his groundbreaking 1997 paper, Daniel Levinthal showed that this logic can provide a simple and intuitive explanation for the persistence of heterogeneity among firms, contrary to the assumption of neoclassical economic models that selection will inexorably drive all firms to the same set of best practices (see also Siggelkow, 2002).

Whereas biological versions of NK models typically rest on the assumption that search can only proceed by small and local variations, applications to social domains have modeled different and more sophisticated search procedures, such as modular search (Ethiraj and Levinthal, 2004b; Brusoni et al., 2007) and search driven by mental models, analogy, and other cognitive processes (Gavetti and Levinthal, 2000; Gavetti, Levinthal, and Rivkin, 2005; Csaszar and Levinthal, 2016). Other papers have used NK models to study organizational search processes where the organizational structure, similarly to the modular structure of a technology, determines which vectors of organizational policies can be tested and selected against a given environment (Marengo et al., 2000; Siggelkow and Levinthal, 2003; Siggelkow and Rivkin, 2006), against a fast-changing environment (Ethiraj and Levinthal, 2004a), or in cases in which organizations can, to some extent, shape the environment to which they adapt (Levinthal and Warglien, 1999; Gavetti, Helfat, and Marengo, 2017). Other papers have also extended these organizational models to the study of incentives (Dosi, Levinthal, and Marengo, 2003; Ethiraj and Levinthal, 2009; Marengo and Pasquali, 2012) and organizational conflict (Marengo, 2020). Finally, and closer to our own application, some models have studied the structure and dynamics of industries rather than single organizations (Adner, Csaszar, and Zemsky, 2014; Lenox, Rockart, and Lewin, 2007).

Our model builds upon a representation of a product system that has some features in common with a NK technological landscape, but also some important differences. Below, we briefly summarize these differences, which underpin our original methodological contribution to the NK model literature. A more detailed and technical description appears in the next section.

First, in contrast to standard NK models of technological search, firms in our model do not search only the technological landscape, but also the organizational one. The latter is based on and constrained by the former, but represents a population of firms, all with different specializations and degrees of vertical integration. Firms produce one or more components and can choose a configuration. They can buy the inputs they need from other firms or vertically integrate upstream and produce them in-house; they can sell their products to firms that need them as inputs, or vertically integrate downstream.

Firms face a tradeoff between the benefits of specialization (finding the best-performing element and sticking to it) and the benefits of integration (exploiting the larger scope of control, as well as a larger set of vertical boundaries to navigate). The search process of individual firms depends on the complexity of the technical system, the total number of active firms, and the path-dependency of firms' choices in terms of specialization and positioning. Furthermore, firms' location up or down the value chain becomes important: firms specialized in downstream components may mix and match components supplied by upstream firms, yet their options may be constrained by such suppliers (Ganco, Kapoor, and Lee, 2020). Upstream firms, conversely, face fewer choice constraints, but are dependent on downstream firms using their components. Such relative benefits emerge endogenously from the introduction of an organizational dimension to the NK model, and capture important strategic facets of research on industry architectures and ecosystems, such as the fact that firms must balance technical performance with architectural leverage—or, more generally, value creation and value capture. Crucially, our organizational search function is able to capture and interpret asymmetries between firms in our modeled industries. For example, in the analysis of Jacobides, MacDuffie, and Tae (2016), OEMs managed to change their industry structure precisely because they maintained control of the interface with the market (among other things). Similarly, in the airplane engine example above, only GE could take the decision to reduce 855 components to 12, dramatically influencing its own supply chain; the converse was not possible. Overall, the importance of this part of our modeling strategy is that firms' performance depends directly on their own choices as well as those made by other agents, as the accessibility and attractiveness of their locations in the landscape is dependent on where other firms are located at that point in time.

Our second main adaptation lies in our treatment of interdependence and coupling. Our model follows the interpretation of a product as a complex system of complementarities between components (e.g., Jacobides, Gawer, and Cennamo, 2018). In doing so, it controls the intensity and sign of interdependences. Hence, instead of randomly drawing fitness contributions for each possible configuration of a subset of interdependent components, we use a functional specification of the fitness contribution of each component with randomly drawn coefficients for the interaction terms (similar to Rahmandad, 2019). This modeling feature enables us to manipulate the characteristics of a product architecture, which is expressed as a set of complementarities between choices with positive or negative values. This is a core element of our model, as we intend to simulate what happens when product architectures become plastic to the agentic efforts of organizations that manipulate them to their advantage.

Through this modeling strategy, we capture the intuition that digitalization might be a threat to firms that enjoy (and take for granted) an advantageous position, and an opportunity for those that do not, because it provides a chance to alter the rules and roles of competition and value generation. Our modeling strategy captures the mechanism behind such shifts, by endogenizing the product architecture in our NK model. We reveal how changes in the technical landscape (which we term "redesigns") generate changes in the organizational landscape. In so doing, we study the changing drivers of performance, and hence the way firms search for value in endogenously changing ecosystems.

In our model, individual firms are able to endogenously adapt the product architecture on which population-level dynamics are based. This may directly or indirectly affect their own and others' performance: directly in the sense that they may simply find "better" architectures, and indirectly in the sense that these architectural adaptations may affect the conditions based on which they and others make organizational choices.

MODEL

We build on the standard NK model, which we expand in two ways. First, we specify a fitness function to describe how components interact to determine the fitness of a system. Second, we assume firms navigate an organizational landscape (where they search for *positioning*) on top of a technical landscape (where they search for *configurations of components*). This organizational landscape describes which components they make and, consequently, their interactions with other firms. Positioning choices available to firms are, for example, to be fully vertically integrated, to specialize in upstream or downstream components, or to take intermediate positions. As is indeed observed in industrial contexts, different firms coexisting in the same industry can end up making different choices in this regard (Kapoor, 2013). Based on these two extensions to the model setup, we can then introduce *redesign* as an additional search mechanism and explore its characteristics and consequences. Here, we describe the features of our model; Appendix A provides a pseudo-code and examples along with computation steps for all relevant mechanisms.

Fitness

NK models assign fitness values to configurations of components, to capture their performance in an external selection environment. The fitness value of a component depends on its own state (0 or 1) as well as on those of K other components. Typically, this is formalized by assigning a different fitness value, randomly drawn from a uniform distribution between 0 and 1, to each of the 2^{K+1} possible states. A system with N components then has 2^N possible configurations, with each of the N components assigned a fitness value, and the fitness of the configuration is taken as the average of the fitness of the N components. The total set of 2^N firm-level fitness values is the fitness landscape that firms explore.

Our adaptation lies in how the vector of 2^{K+1} fitness values is drawn. These values imply positive and negative complementarities. Imagine a stylized interaction between components A and S, which together make up a computer system. The fitness value of the computer assembly (component A: desktop (0) or laptop (1)) depends on the configuration of the screen (component S: an integrated touchscreen (0) or separate monitor (1)). This interaction effect might be captured by the following mapping of configuration [AS] to fitness values: [00]: 0.5; [10]: 0.7; [01]: 0.8; [11]: 0.2. This would mean that, while designing the system as a laptop *or* with a monitor improves its performance (compared to a desktop system with a touchscreen), a laptop design in combination with a monitor is detrimental to performance. In other words, there is a positive impact of switching A and S (separately) from 0 to 1, but a negative interaction effect of switching both A and S to 1 simultaneously. The information provided by these four fitness values is more precise than the parameter K can capture—indeed, the presence of an interaction between A and S is typically visualized as a mark or cross in an interaction matrix. Our approach captures all the information on the structure of interdependence between components, by specifying component fitness as a logistic function as follows:

$$fc_i(c_i, \dots, c_{K+1}) = \frac{1}{1 + e^{-(B \cdot X)}}$$

Here, X captures the configuration of each component state and their interactions. In our stylized example, it is a vector of three elements (whether A is 1, whether S is 1, and whether A and S are both 1). In general, it has $2^N - 1$ elements, including all (higher-order) interaction effects. B is a vector of the same length, with coefficients capturing the impact of each state in the X vector being 1. The fitness values of our stylized computer system assembly would be approximated by the B vector [0.85, 1.39, -3.63]. Both the valence and magnitude of our B vector coefficients are meaningful in the context of complex systems. The valence, as illustrated, captures complementarity (either positive or negative) between choices. This captures the tightness of coupling between components. This fitness function is a generalization of the standard NK fitness specification: for every set of 2^{K+1} random numbers, a B vector exists (in combination with an offset value before the $B \cdot X$ term) that captures these numbers. For consistency, we assume this offset value to be 0, so that our fitness values are centered around 0.5 (and indeed *are* 0.5 when all elements of the X vector—and all states in the configuration—are 0). For more technical details and descriptive features of our specification of NK landscapes, see Appendix B.

Figure 1 illustrates the above example. The upper area represents the system in NK terms. The fitness value of component A depends on both its own state and that of component S (signified by the x in the interaction matrix). Both component A and component S can exist in stylized states 0 or 1. The left panel shows the traditional NK fitness value specification: every unique configuration has a random fitness value. The right panel shows our specification, where fitness is given by (1), and determined by the dot product of two vectors: vector X, identifying the state of the system (whether A is 1, S is 1, and whether A and S are simultaneously 1); and vector B, with random coefficients for each (individual and interaction) effect.

--- Figure 1 here ---

Search in a Techno-Organizational Landscape

We evaluate a model where the production system (more specifically, its design choices) is distributed over vertically related firms, by introducing an organizational landscape. Firms make a subset of vertically related components, and (indirectly) interact with firms making the remaining components. We assume that the supply or value chain overlaps with the interaction structure. That is, our stylized computer assembly (A) is dependent on the screen (S), so we assume that firms specialized in making assemblies must buy screens from firms that make them. Firms are dependent on choices made by upstream firms: if there are no firms making touchscreens (component S in state 0), firms making assemblies are constrained to configurations [AS] 01 or 11, and are better off selecting desktop designs. In more complex settings, they can mix and match components sourced from multiple upstream firms (akin to the ecosystem NK model in Ganco, Kapoor, and Lee, 2020). Conversely, firms are dependent on downstream firms (or the downstream market), which may or may not seek the focal firm's configuration. For example, if all assemblers have opted for laptop designs, it is unattractive to produce separate monitors. When firms are integrated, rather than specialized, they control multiple components and can thus navigate the complexity of the system. When all firms are fully integrated, we are back in the traditional NK model.

Search and evaluation mechanisms in our environments are close analogies to those employed in prior work. First, firms can *search* for a better configuration, given their position in the organizational landscape. This means that they can change suppliers (e.g., a specialized assembler can find a firm that produces a different type of screen, moving from configuration 00 to 01), change configuration (e.g., changing assembly type with the same input component, moving from 00 to 10), or both. We assume firms try and evaluate a random configuration that is both new and feasible. A configuration is feasible if every necessary component is either made by the firm itself, or by any specialized upstream firm(s).[2] Note that we treat bounded rationality differently from most prior models, which allow firms to search only locally (one bit mutation at a time). We instead capture the benefit of integration in being able to make multiple simultaneous (i.e., systemic) changes. In analogy with NK models of modularity employing so-called *S-search* (Brusoni et al., 2007; Fang and Kim 2018), we assume that firms can adaptively modify all components they produce; therefore, more vertically integrated firms have a broader scope for adaptation (consistent with the notion that integrated firms enjoy the benefit of being able to introduce and manage systemic

[2] We assume that integrated firms use the (upstream) components they produce in their own (downstream) systems.

innovations). However, all firms, regardless their degree of vertical integration, are boundedly rational insofar as their adaptive search proceeds only by myopic trial-and-error, both in the sense that they only consider the short-term effect of their actions and that they do not anticipate other firms' reactions.

In addition to search *given* a position on the organizational landscape, firms can search this landscape, by *integrating* or *disintegrating* vertically. For example, a screen manufacturer may seek to become vertically integrated and produce the entire computer system if there are no downstream firms interested in its separate monitors. Conversely, an integrated firm may seek to cease its screen production to specialize in assemblies, if there are many firms specialized in screens. Similar to our search mechanism, we assume firms evaluate a random feasible configuration where the firm would produce one random (vertically directly related) component more (less) than it currently produces.[3]

Intuitively, we need another indicator for assessing firms' performance as they navigate our organizational and technical fitness landscapes. For example, in our stylized industry, there may be just one firm producing integrated touchscreens (component S in state 0), but many producing separate monitors (component S in state 1). This imbalance would have a twofold effect on downstream firms. First, it shapes their choice of screen option (here, any configuration that requires sourcing integrated touchscreens is unattractive). Second, it may push them to integrate upstream if they do want to produce systems with integrated touchscreens. Similarly, in downstream segments there might be more demand for laptop assemblies than for desktop assemblies. Value-based strategy research (Brandenburger and Stuart, 1996; Gans and Ryall, 2017) captures this intuition by suggesting that a firm needs to contribute to a value system if it is to capture any of the aggregate value created.

[3] We emphasize relatedness to prevent "holes" in a firm's production chains.

In our model, a firm's generated fitness (as determined by its chosen string of 0s and 1s) is accordingly not a complete measure of firm performance. We assume that a firm's performance in the competitive environment is some function of the fitness of its chosen configuration, as well as the extent to which this configuration contributes to a value proposition in the industry. That is, if the firm has difficulties finding necessary inputs (there are more firms seeking the component than offering it), we assume performance to be negatively affected. In the opposite case, if there are many suppliers for a component compared to competing buyers, we assume that this has a positive impact on performance.

Figure 2 illustrates how other firms' choices have a positive or negative effect on the performance of a (focal) firm, beyond the generated fitness of their selected configurations. For example, if few firms are offering component S in state 0 (left panel, top), this has a generally negative effect on the performance of firm *i* if it chooses a configuration that requires sourcing a component S in state 0. If many firms are using component A in state 1 (right panel, bottom), this has a generally positive effect on the performance of firm *i* if it chooses a configuration that includes component A in state 1. Thus, depending on the actions of all firms, regions of the focal firm's fitness landscape can be heightened, lowered, or even restricted (Li and Csaszar, 2019), with all effects endogenously emerging from firms' choices and adaptation to each other.

To simplify our analysis, we formalize this intuition by introducing a stylized function to "correct" the generated fitness, evaluated per component and determined by firms' positions in the organizational landscape. For interactions across upstream firm boundaries, this function is based on the ratio between the number of suppliers providing a sourced component and the number of firms seeking that component. For interaction across downstream firm boundaries, it is based on the ratio between downstream firms seeking the focal component and the number of firms providing it. This approach simplifies many realworld mechanisms surrounding production, competition, and bargaining power, in favor of capturing stylized patterns of stability, coordination, and search for performance. In this paper, results are shown employing a correction factor of $\log_2(x+1)$, where x is the ratio described above.[4] For computational examples of how performance is calculated, see Appendix A.

--- Figure 2 here ---

Modeling Redesign

A redesign alters the structure of interdependences defined by the coefficients of our previously specified B vector. The effect of a redesign on a coefficient can be twofold. First, it may "solve" a negative complementarity, making certain design choices work well together where they did not before (changing the valence of an interaction coefficient from negative to positive). Second, it may weaken the severity of negative complementarity (bringing a negative coefficient towards 0). For example, in our stylized computer example, a hypothetical interface may be found that allows a laptop design to use any available surface as a display (i.e., the strongly negative coefficient capturing the interaction between A and S improves, and might even turn positive). A redesign may also affect the direct effect of a design choice—for example, if this hypothetical interface comes at the expense of the effective resolution of monitors (here, the coefficient for S is lowered). Depending on relative changes along the coefficient vector, the shape and peaks of the fitness landscape of a component change (see also Rahmandad, 2019, for a more elaborate treatment of a similar implementation).

[4] This function has the attractive properties that it returns 0 for x = 0 (with no buyer/supplier, there is no performance), and 1 for x = 1 (indicating balance between suppliers and buyers). As robustness checks, we also employ various root functions of x. Furthermore, as an additional approach, we build an "organizational fitness landscape" specified by the logistic function y = 1 / (x^(-1/log(2)) + 1). This function is centered in (1,0.5), and symmetrically approaches 0 as x^{-1} approaches ∞ and approaches 1 as x approaches ∞. We then balance this landscape with the fitness landscape, either by addition or multiplication. While absolute levels of performance change, our model dynamics are robust.

We assume that firms can redesign subsystems under their control. That is, they can change coefficients of the B vector of any components they make, regardless of whether they themselves make the component indicated by the coefficient. Redesigns are exclusive to the redesigning firms themselves, and do not affect the fitness functions of other firms. In line with our example, an upstream firm making screens could be affected in terms of how its screens affect the performance of a single redesigning downstream firm's assembly, but this doesn't affect how its screens affect the performance of other firms' assemblies.[5]

We model redesign as follows. We assume a redesign to be a form of "distant search" (e.g., Levinthal and March, 1993; Katila and Ahuja, 2002), meaning that firms will only turn to this solution if they cannot find improvements through any other mechanism. When a firm redesigns, it randomly redraws 20% of its B vector coefficients, across the components it produces.[6] It then considers the redesign, both in isolation (the firm may already be better off in its current configuration), and in combination with each of the three improvement mechanisms (search, integration, and disintegration). As an illustration, consider again the situation where a specialized assembler (i.e., a firm making only component A and sourcing component S) faces just one firm producing integrated touchscreens, making any configuration with integrated touchscreens highly unattractive. Yet, there is much competition from firms producing desktop systems with monitors. The specialized assembler is trapped on a low local peak, due to the interaction between the technical and the organizational landscape. Here, the hypothetical redesign that alleviates the negative interaction between laptop designs and monitors may offer the firm a way out. Even if it loses some fitness value by switching to a laptop design with a separate monitor, this loss may be offset by the advantage of being able to switch to the more commoditized input (the separate monitor). If

[5] In modeling terms, each firm is initially assigned the same B matrix of N x $2^N - 1$ random values. Over time, these matrices start to diverge.
[6] We opt for a percentage rather than a fixed number, to avoid disproportionally affecting specialized and integrated firms.

the firm finds a configuration with higher performance, it adopts the redesign and any additional (organizational or configurational) change. Again, we assume firms have only one chance at improvement (in the redesign, and in each of the improvement mechanisms) and have no prior knowledge informing their trials. In our findings, we distinguish between redesigns that offer immediate advantages, which we will call *landscape* redesigns, and redesigns that offer advantages only if coupled with additional (organizational or configurational) changes, which we will call *ecosystem* redesigns.

Our redesign function bears a resemblance to landscape changes in prior literature, such as landscape shaping (Gavetti, Helfat, and Marengo, 2017), innovation (Rahmandad, 2019), and architectural innovation (Albert and Siggelkow, 2022). All these constitute changes to the values of the fitness landscape when firms successfully seek out improvements. There are several key differences, however. First, as described above, our redesign only affects the fitness landscape of the firm that discovers the new design. Second, whereas in Gavetti, Helfat, and Marengo (2017) shaping consists of modifying some environmental characteristics that interact with product components to determine their fitness, in our model firms act directly on some technological interdependences. Third, our approach is markedly different from architectural innovation as in Albert and Siggelkow (2022), who model architectural change as a change in *which* interdependences exist in a system. We instead manipulate the *strength and valence* of interdependences. As presented later, both implementations do share some common results. On this basis, in our model, the landscape change has a more precise effect on performance and industry dynamics, due to the workings of the underlying coefficient set (our B vector). Rather than redrawing every possible position in the landscape (i.e., the fitness values associated with each configuration), our redesign redrawing leaves most untouched, and the magnitude of change between pre- and post-redesign fitness is low. Nevertheless, mutating just a few landscape positions is precisely the change that our model is interested in. Following our running example, if redesign makes an assembly work better with a touchscreen, the redesigning firm becomes less dependent on monitors, while all else remains equal. See Appendix B for a more detailed comparison between our landscape and traditional NK landscapes as interaction structures change.

RESULTS

We discuss our results in two parts. We first explore the features and dynamics of our industry NK model. Subsequently we explore what happens when firms are able to endogenously manipulate the product architecture governing the technological landscape.

Baseline Industry NK Model

We first test our model without redesign, to explore the baseline outcomes of a model that allows for configuration search in conjunction with vertical positioning search. Thus, in a first step, we explore the organization of the industry (averages and distribution of vertical specialization and performance) as we vary the starting parameters of the model. These starting parameters reflect the main features of industries: number of components N, complexity as number of linkages between components K, and number of firms F. Our simulations proceed as follows.

Every simulation run initializes the NK landscape with N components and up to K interacting components per focal component.[7] F firms are initialized as fully disintegrated, and randomly allocated over the components (with each component made by at least one firm). A downstream market is created for each of the most downstream components, with

[7] Given that we assume the supply chain network to overlap the interdependence structure, we prevent cycles of interdependence from occurring.

demand F/N, equally divided between variations “0” and “1” of that component.[8] In each of 100 time periods, the order in which mechanisms are explored (*search*, *integration*, *disintegration*) is randomly determined. In every time period, for each mechanism, firms are lined up in random order to each try and evaluate a new configuration, and firm performance of those firms later in the sequence is updated to take into account the choices of those earlier in the sequence.

Figures 3 and 4 show the behavior and outcomes of firms in our model, with increasing degrees of aggregation. Jointly, they build a picture from the bottom up of how our industry NK model functions. Figure 3 shows the frequency of our model mechanisms (search, integration, disintegration) as a percentage of firms successfully executing them over time, with the different panels displaying different parameter settings and the central panel showing a reference parameter set (i.e., N = 8, K = 3, F = 24). Search for positioning in the organizational landscape is shown in *integration* and *disintegration*, while search for superior configurations within the technological landscape is shown in *search*.[9] Each successful execution accounts for a firm improving its overall performance, although this performance gain might later be undone by the activities of other firms. Consider first the central panel: how do firms in our model search for value over time? We first see a strong wave of integration, explained by the fact that we initialize firms as fully disintegrated[10]. We see a smaller wave of disintegration that picks up after a few periods, meaning that firms reverse their integration decisions or divest their originally assigned components to take up another (more specialized) position in the organizational landscape. We see a wave of search activities

[8] This number serves as an “anchor” for our model, and promotes comparability between parameter settings. Additionally, it makes results of industry dynamics less sensitive to random variation of landscape values.
[9] Note that while search keeps the firm’s vertical scope stable, all three mechanisms can affect simultaneously the fitness of the chosen configuration as well as how “organizationally attractive” this configuration is, e.g., when a firm searches for a configuration for which there is less competition.
[10] When we initialize firms with random degrees of vertical integration, average end-state degrees of vertical integration are correspondingly higher.

that accounts for firms finding new product configurations within their position in the organizational landscape, with comparable frequency to organizational improvements. Over time, the occurrence of each successful adaptation decreases, as firms collectively reach a state of local equilibrium. Thus, agents in our model simultaneously search for performance by adapting both the configuration of components and their vertical scope, eventually settling on a position in the techno-organizational landscape. We display varying parameter settings along the three main model parameters as follows (here and in subsequent results presentations). From left to right, we increase the system size N (from 5 to 12, with corresponding changes to K and F). From bottom to top we increase complexity K. Diagonally from bottom left to top right we increase the number of firms F. When comparing findings across parameter settings, we see that the overall patterns are always present, but vary in strength, intensity, and duration across parameter settings. Varying across system size N, we see that in smaller systems, search is slightly more frequent than organizational mechanisms. Varying complexity K, we see that in more complex industries, firms adapt more frequently, and the overall process of adaptation lasts longer before equilibria are reached. Additionally, in more complex industries, organizational adaptation mechanisms are more frequent, since there are more organizational boundaries to explore. Finally, varying across number of firms F, our main finding is that the speed at which industries stabilize is inversely related to the number of firms in that industry. In industries with few participants, every firm's actions have a relatively large effect on others, leading to instability overall.

--- Figure 3 here ---

Figure 4 shows the results of the same model, displaying the development of vertical scope and performance over time. Note that the line for vertical scope essentially tracks the difference between integration (increase) and disintegration (decrease) in Figure 3. We again organize our results per varied parameter, in this figure by including the reference parameter set in each figure. As a whole, our model replicates a well-known finding from previous modeling studies of search in complex systems (e.g., Ethiraj and Levinthal, 2004b): systems converge at different levels, with different speeds, depending on model parameters. Here, more specifically, we see a fast convergence in less complex (low K) industries, as well as in those with many firms (high F). More complex industries tend to become more vertically integrated (as expected), but this does not hold for industries with many firms F. This suggests that the existence of many firms mitigates the coordination problem necessitating integration under complexity. Aggregate performance evolves similarly to vertical scope, yet increases in any parameter lead to higher eventual outcomes. See Appendix C for an elaboration of our baseline model findings along model parameters. One noteworthy additional finding is the emergence of bottlenecks: in industries with few firms (low F), we find a large variance in performance. Conversely, with many firms, performance is much more equally distributed. In other words, bottlenecks emerge when firms are dependent on the choices of a few others.

--- Figure 4 here ---

To sum up, our purpose up until this point was to test our newly developed industry NK model against baseline expectations, and understand its main dynamics. Consistent with prior literature, we find that complexity drives vertical integration. Furthermore, the ability for industrial participants to coordinate activities (and become specialized) depends on the number of firms active in the industry. Our technical and organizational landscapes interact in such a way that firms need to navigate both in tandem, leading to the emergence of bottlenecks and performance differences, and to differences in vertical positioning. All of the above assumes stable technical architectures; we now turn to the situation where these become endogenously malleable by firms.

Redesign

In the second set of simulations, we introduce a single exogenous change halfway through the simulation: from period 101 on, firms are able to redesign the technical system they operate in. The redesign mechanism is activated at the end of each period, with firms lined up in random order. Each firm generates a redesign, and first evaluates its performance in isolation. Then, it evaluates the adaptation mechanisms (search, integration, disintegration) in random order, in search of additional performance gains. See Appendix A for an additional description and computational example. While we assume away search costs (of redesigns, or indeed of any other choices), redesigning technical systems is non-trivial and inherently risky (Henderson and Clark, 1990; Albert and Siggelkow, 2022). We operationalize this aspect by modeling it as a last resort that is only considered when search using readily available mechanisms yields no benefit: Firms can only use the redesign mechanism if they could not find any improvement in the current or previous time period.[11]

We discuss the results emerging from these simulation experiments as follows: We first evaluate the broad impact of the availability of redesign for industries, and subsequently explore in more detail the mechanisms that underlie the overall effects. We first examine the development and distribution of vertical scope and performance before and after our introduced "shock": what happens to the scope and performance of firms when they can manipulate the product architecture that governs the technical landscape? And do all firms experience a similar effect?

Figure 5 shows the frequency of successful improvements, organized along the same parameter settings as before. It shows that across parameters, when redesign is an option, firms use it—but other improvement mechanisms briefly become more frequent as well. What does this mean? When redesign is an option, firms do not simply improve on a new dimension

[11] In robustness checks, we increase the number of periods firms must go without improvements to engage in redesign. While effects become correspondingly weaker, they do not change qualitatively.

of technical architecture (as then we would have seen only the redesign line pick up). However, nor does redesign lead to a “reset” of industrial organization, as the waves of adaptation are much smaller than at the beginning of the simulation. Instead, the results suggest that the redesign option disturbs the local equilibrium and requires some firms to adapt by engaging in renewed search in the techno-organizational landscape.

--- Figure 5 here ---

A next step in understanding the implications of the redesign possibility is to investigate the main outcome variables of the model: vertical scope and performance. Figures 6 and 7 display the distribution of scope and performance over time, with standard deviations indicated by the shaded areas. Note again that the averages between t = 0 and t = 100 in Figures 6 and 7 correspond to the values reported in Figure 4. Here, we see that the effects at the industry level are not distributed equally. Interestingly, average vertical scope increases or decreases depending on parameter settings. Specifically, in only those industries with fewer firms, firms become more vertically integrated—suggesting that, on average, the possibility to redesign allows firms to become slightly more specialized. However, in all cases, the distribution of average vertical integration remains stable, indicating there remains a mix of specialized and integrated firms. As expected, average performance always increases when we give firms an additional way to improve performance. However, interestingly, the variance of performance always increases, suggesting a growing inequality between firms as a result of the possibility to redesign. Note that this provides a stark contrast to our baseline model findings, where only industries with few firms were characterized by uneven distribution of performance.

--- Figures 6 and 7 here ---

We go one layer deeper in investigating the performance implications of redesigns for all firms. The previous sets of results suggest that redesign affects complete populations of firms as it may disrupt the inter-firm equilibrium conditions, and that this leads to large performance changes. So, are some firms benefiting more than others—and if so, which ones, and why? To gain more insights, we lined firms up on two dimensions: vertical scope and upstreamness. We include the dimension of upstreamness to capture the potential importance of asymmetries in search patterns of firms located upstream and downstream in the value chain (Ganco, Kapoor, and Lee, 2020). The upstreamness of a component is here normalized as the distance to the downstream market (ranging from 1 to N) divided by the length of that value chain.[12] The upstreamness of a firm, then, is the average of the upstreamness of its components. Given that more integrated firms necessarily operate both upstream and downstream components, their upstreamness is by construction more average, and our grid of firms looks like a triangle rather than a squared grid. We line up our firms and measure their performance right before we enable the possibility to redesign, and measure their performance again at the end of the simulation. The difference in performance is shown in Figure 8, for three levels of complexity K, and helps us understand which types of firms benefit more from being able to redesign. Performance increase is indicated by the color scale of the cells.

We find two groups of firms that outperform the rest of the population. The first are upstream suppliers (found at the top left of each triangular graph). These firms benefit from redesign for two main reasons. First, they operate in relatively stable spaces. This enables them to redesign frequently, as they do not have to adapt to their surrounding conditions. Additionally, their performance landscapes are less complex than those of firms making downstream components, allowing them to find peaks more quickly. The second group of high-performing firms are more integrated, downstream firms (found at the bottom of each triangular graph). The reason why these firms outperform their upstream counterparts in particular can be traced back to the asymmetric benefits of being able to redesign: Firms can

[12] Our initialization allows for parallel (pooled) steps in value chains, so maximum length may differ between simulation runs. For this reason, we use a normalized number.

change their dependence on upstream components, potentially allowing them to bypass bottlenecks. Going back to the example of GE redesigning its airplane engine, GE could decide to overturn its supply chain; upstream firms in the supply chain had no such choice. Looking from left to right between panels, we see that complexity acts as a moderator. In more complex industries, the benefits of redesign shift away from upstream specialists toward downstream integrators. This finding is consistent with the idea that the advantages of integration, when redesigning, increase with complexity (e.g., Teece, 1986). That is, when firms have to navigate a larger network of components and upstream suppliers, they are able to use the redesign mechanism more to their advantage. Additionally, in more complex industries, upstream components are typically used in many downstream components, meaning that even specialized suppliers are involved in many value chain interactions. As a result, they are affected more by choices made by other firms in the industry.

--- Figure 8 here ---

The firm-level performance implications allow us to see and speculate about why some firms benefit most from being able to redesign. However, can we be more specific about how different firms use the option to redesign, and how they benefit from it? To answer this question, we zoom in on what exactly our firms do when they redesign. Recall that our modeling assumption was that firms can explore up to one redesigned technical architecture, and evaluate one search, integration, and disintegration change with that hypothetical architecture in hand. Redesigns themselves constitute the redrawing of (a subset of) the coefficients shaping the technical fitness landscape. When a firm redesigns for the first time, then, there is a good chance that its current configuration will perform better with the redesign than without it (in which case the redesign is adopted). It might also be that the redesign lowers performance, but a performance increase can still be found with the additionally explored adaptation mechanisms (in which case both the redesign and the additional change are adopted). Here we assign labels to these different types of redesign, and we evaluate their effects and the characteristics of firms that are able to exploit them.

Redesigns that provide performance enhancements purely by increasing the performance of a firm in its current position in the techno-organizational landscape can usefully be called “landscape” redesigns. This rather straightforward finding closely resembles findings in prior modeling studies (what Albert and Siggelkow, 2022, call “pure” architectural innovation, and what Rahmandad, 2019, calls “innovation”). Our second type of redesign is subtler and more complex in nature, as it involves a changed product architecture that enables a firm to successfully reposition itself in the techno-organizational landscape. For example, such a redesign might allow a firm to change supplier (from a scarce component to a more common one). Thus, the technical redesign would benefit the firm by affecting the ecosystem and its inter-firm interactions as a whole. We call these redesigns “ecosystem” redesigns to signal their impact in the (inter-)organizational domain.

Figure 9 displays aggregate statistics about the two types of redesign. It shows (left) the average performance increase that a firm experiences when the redesign is successfully executed, and (right) the probability that firms manage to do so. For completeness, it also includes those redesign events with overlapping characteristics (i.e., those where both the redesign itself and the subsequent repositioning on the techno-organizational landscape confer benefits). We first turn to landscape redesigns. We would expect such landscape improvements to become both less frequent and less beneficial over time, as each successful redesign shrinks both the pool of potential redesigns and the room for improvement. We do indeed see this result in the figure. Initially, ecosystem redesigns are just as beneficial for firms as landscape redesigns, but can still confer performance improvements throughout the simulation. This is remarkable, given that they do not improve firms’ performance in their current state. At the same time, they are much less likely to occur throughout the simulation.

To explain the enduring improvements conferred by ecosystem redesigns, it is helpful to revisit the reason why firms get "stuck" on local peaks in the techno-organizational landscape. In many cases, this is due to choices made by other firms, which can essentially restrict the focal firm from reaching certain segments of the landscape. An ecosystem redesign, then, is a way out of this trap. Finally, and as we would expect, observed "combination" redesigns essentially provide the sum of performance benefits of landscape and ecosystem redesigns, yet are exceedingly uncommon.

--- Figure 9 here ---

Finally, we connect our observations about firms and redesign events. We do so by measuring, for every successful redesign event, the most relevant characteristics of the firm at the time it finds the redesign, and the type of redesign that is found. Figure 10 displays the results (leaving out combination redesigns) for different complexity settings. Generally, it shows that ecosystem redesigns are successfully employed by firms that are positioned further downstream in the supply chain, more vertically integrated, and less successful at the time of the redesign. We describe each of these findings in turn.

--- Figure 10 here ---

As suggested by the previous set of findings, upstream and downstream firms each benefit from redesigning in different ways. This finding is supported by the top-left panel in Figure 10. Upstream firms are in a position of relative stability, where they can optimize the technical fitness landscape. Downstream firms rely on ecosystem redesigns (e.g., allowing them to change their dependence on the choices of upstream suppliers), which are rarer but more consequential. This effect is attenuated with increasing complexity, as in more complex industries more upstream firms also have a significant number of suppliers they depend on.

Regarding vertical integration (top-right panel), here we also see a clear pattern of firms exploiting the benefits of redesign differently depending on their characteristics. Relatively integrated firms operate across more boundaries and are faced with larger fitness landscapes overall. As such, they are more likely to find redesigns in combination with configuration changes, in either the organizational or technical landscape. Again, this effect is moderated by industry complexity: in more complex industries, integrated firms benefit more often from ecosystem redesign. Specialist firms operate in relatively simple landscapes and depend on few other firms (regardless of complexity). As such, they are more likely to find global peaks in their fitness landscape and mainly benefit from redesigns if they further heighten this peak (i.e., landscape redesigns).

With regard to performance at time of redesign (bottom-left panel), lower-performing firms are typically those that are trapped on low local peaks, due to their dependence on other firms around them that happen to be making unfavorable choices. It is these firms that benefit from finding a new path away from these local peaks (i.e., ecosystem redesigns), independent of complexity. Together with the other results here, this finding supports the explanation in the previous section that upstream specialists are in a relatively stable space to optimize their fitness landscape.

Finally, the bottom-right panel displays average performance improvement. Ecosystem redesigns are more performance-enhancing (as also seen in Figure 9), but this effect is positively moderated by complexity K. This moderation effect is due to the number of inter-firm interdependences that ecosystem redesigns can possibly affect. Ecosystem redesigns provide value by reshaping the way firms depend on others, making them especially beneficial in more complex industries.

DISCUSSION

New digital technologies are considerably increasing the plasticity of product architectures. We have argued in this paper that one important implication of this plasticity is that firms can proactively strategize on redesigning product architectures, as opposed to reactively adjusting to them. We have extended traditional NK models by adding an organizational dimension to the search process, and are able to explore the process of redesign in what we call techno-organizational landscapes.

Our model corroborates well-known stylized facts surrounding vertical organization in the absence of redesign possibilities, and offers important insights about redesigns when they are possible. Table 1 summarizes our insights. In particular, we put forward two distinct ways in which firms may leverage digital technologies: landscape redesigns and ecosystem redesigns.

-- Table 1 here --

Examples of landscape redesigns include product improvements such as using 3D printing to adopt new geometries for complex, mission-critical components (e.g., redesigned fuel nozzles in the jet engine industry, which saved waste material and improved performance). This is the kind of innovation studied in Albert and Siggelkow (2022). Examples of ecosystem redesign include the redesign of the ATP engine we saw at the outset, and more broadly the emergence of companies like Uber and Airbnb, which have reinvented the ecosystems of established industries. These are the situations discussed by Pisano and Teece (2007) with reference to the reengineering of industry architectures enabled by digital technologies.

These two types of redesign capture in a parsimonious way the fundamental features of competition in ecosystems in which digital technologies impact how product architectures are developed. Our model captures how different firms, while facing similar technological developments, take different actions and benefit from them in different ways.

Our findings, in particular the distinction between landscape and ecosystem redesigns, can explain conflicting expectations regarding the impact of digital technologies, and potentially reconcile them. Digital technologies have received plenty of attention, in both academic and popular discourse. The popularity of the topic has also created vastly diverging views about it. For example, Lanzolla et al. (2021: 342) note that the adoption of digital technologies has been predicted to drive specialization (e.g., Brews and Tucci, 2004) as well as integration (e.g., Yoo et al., 2012). To some extent, the problem is related to the reliance on overly broad terms (e.g., “blockchain,” “IoT,” “additive manufacturing”, “AI”) that encompass very different phenomena. Our study shows why thinking of digital technologies as industry-wide drivers of change (under banners such as “digitalization” and “fourth industrial revolution”) is potentially misleading. If, as we suggest, one important aspect of digital transformation is that product or technology architectures become a strategic variable rather than a constraint, then the outcomes will differ at the firm level, because the benefits of transformation depend strongly on *how* it is applied, the characteristics of the firms who do so, and their initial positioning.

These findings suggest a new impetus for research on technology strategy. Scholars interested in the question of how to benefit from innovation (cf. Teece, 1986) have highlighted the importance of industry architectures as a key determinant for value capture and suggested that attempts at shaping such architecture should be at the heart of any digital technology strategy (Jacobides, Knudsen, and Augier, 2006; Pisano and Teece, 2007; Jacobides, MacDuffie, and Tae, 2016). Baldwin (2018) suggests that firms may possess “dynamic architectural capabilities,” which allow them to understand and manipulate both product and industry architectures.

Our model gives texture to the ongoing discussion about industry architecture re-engineering. It points to two distinct redesign choices through which firms can benefit from being able to change product structures. Landscape redesigns enhance performance *given* a firm's location in the techno-organizational landscape. Ecosystem redesigns alter the conditions shaping the techno-organizational landscape itself, allowing the firm to find a higher-performing position. Both types of redesign are, in essence, changes in a product architecture (our vector of complementarities)—yet ecosystem redesigns change industry architecture as well. Importantly, these two strategic moves are available to firms with different organizational profiles. Firms in relatively protected pockets in the industry architecture employ landscape redesign to optimize the environment they operate in. Firms subject to more volatile conditions employ what we call ecosystem redesign to "escape" from unattractive positions, by coupling technical and organizational change.

These findings point towards a reassessment of architectural innovation (Henderson and Clark, 1990), its origins, and its effects. The prevailing view on architectural change is that it is exogenous and intergenerational. Digital technologies provide firms with the tools to manipulate product architectures in a more proactive way, and our model provides insights into *how* firms might benefit from doing so. This is not to say that architectural change has become easier or less risky for incumbents. Our model, for example, is silent on intra-organizational processes. A promising line for future research is to investigate the organizational limits to perpetual architectural change, in the context of digital technologies.

Additionally, our findings enhance our understanding of the relative benefits of integration and specialization. Prior research has painted a general picture of modes of integration and specialization each providing unique benefits and thus sustainable positions (Kapoor, 2013), yet has mostly assumed product architectures to be stable. Research on systems integrators (Prencipe, 1997; Brusoni, Prencipe, and Pavitt, 2001) suggests that downstream firms should anticipate uneven rates of change in upstream components—for example, by expanding their knowledge boundaries. Our study suggests that the ability to control and manipulate product architectures provides an important new dimension to the benefits of integration—it enables firms to manipulate industry architectures as well. Our model, again, is silent about the intra-organizational processes enabling or guiding architectural change; presumably firms' knowledge bases and capabilities play an important role. Indeed, GE, our running example, is an archetypical systems integrator that "knows more than it makes," and with the new tools of digital technologies can influence and change more than it directly controls.

Our insights and contributions follow from the joint analysis of technical and organizational systems, whose evolution we have argued have so far mostly been studied separately. We have done so by specifying and studying search in what we call techno-organizational landscapes. This novel perspective was developed as a generalization of the NK modeling framework and builds on recent work that has examined separate aspects of innovation patterns (in particular ecosystem structures, Ganco, Kapoor, and Lee, 2020; architectural innovation, Albert and Siggelkow, 2022; and the nature of complementarities, Rahmandad, 2019). Our integrated model allows us to study the impact of digital technologies as affecting dynamics spanning technical and industry structures.

We expect this perspective to contribute to ongoing research surrounding the connection between technical and industry structures (Colfer and Baldwin, 2016). The literature on novel industry structures, such as platforms and ecosystems, emphasizes structural asymmetries both in relationships (e.g., between contractual relationships and those requiring coordination to exploit complementarities (Adner and Kapoor, 2010; Jacobides, Gawer, and Cennamo, 2018; Ganco, Kapoor, and Lee, 2020) and in industry size (e.g., between small platform firms and vast complementor ecosystems that they leverage). Our model can explicitly model and study such asymmetries and the technical architectures (e.g., complementarity structures) they are built on. For example, our findings show that change in product architecture does indeed benefit a few at the expense of the many, reinforcing the idea of architectural advantage as key for success in complex industry architectures (e.g., Jacobides, Knudsen, and Augier, 2006).

The perspective and model we develop help us understand the role of product architectures in the evolution of innovation ecosystems. They expose a process of ecosystem redesign, by which changes in product architecture can be leveraged to destabilize ecosystem architecture. In reality, firms often count (and rely) on stability in product architectures, and redesign thus affects the foundations of long-term planning and strategizing. Future research could take this prediction as a point of departure for untangling sources of stability in industrial organization. In the absence of stable product architectures, what factors could play a stabilizing role? One likely candidate might be network effects, which promote stability for all actors even when some have the power to change interfaces (for an example in digital platforms, see Cennamo and Santaló, 2019). As noted above, one core element in the discussion about platform-mediated ecosystems is that "digital technologies [are] inherently dynamic and malleable" (Yoo et al., 2012: 1399), at least in their peripheral elements. Our model, which captures trends visible in several industries, suggests that this inherent stability in product systems is an assumption waiting to be challenged. Relatedly, one might argue that the role of institutions or regulators who coordinate ecosystems by setting standards or fixing interfaces (Nelson and Sampat, 2001) will become even more important. Recent work focused on the evolution of the AI ecosystem seems to support this intuition (e.g., Jacobides, Brusoni, and Candelon, 2021). These and other factors of stability might become even more crucial, in the presence of decreasing stability in product architecture.

To conclude, the generalization of the NK model we propose suggests a way ahead to reconnect the analysis of technological and organizational landscapes. The quest for methods and tools to analyze the co-evolution of technologies and organizations is certainly not new (e.g. Nelson, 1994). Our modeling framework allows for the analysis of complex choices that capture something fundamental about firms' strategy in the present day, but also reconnect the discussion about innovation ecosystems to the long and distinguished tradition in organization theory that problematized the relationship between technologies and organizational structures (e.g. Perrow, 1967; Lawrence and Lorsch, 1967). We believe our framework extends this line of work and provide a useful way ahead to study how organizations strategize and organize at times of rapid architectural changes.

REFERENCES

Abernathy, W. J., and J. M. Utterback
1978 "Patterns of industrial innovation." Technology Review, 80: 40–47.

Adner, R., and R. Kapoor
2010 "Value creation in innovation ecosystems: How the structure of technological interdependence affects firm performance in new technology generations." Strategic Management Journal, 31: 306-333.

Adner, R., F. A. Csaszar, and P. B. Zemsky
2014 "Positioning on a multiattribute landscape." Management Science, 60: 2794–2815.

Albert, D., and N. Siggelkow
2022 "Architectural search and innovation." Organization Science, 33: 275-292.

Anderson, P., and M. L. Tushman
1990 "Technological discontinuities and dominant designs: A cyclical model of technological change." Administrative Science Quarterly, 35: 604-633.

Baldwin, C. Y.
2018 "Bottlenecks, modules and dynamic architectural capabilities." In D. J. Teece and S. Heaton (eds.), The Oxford Handbook of Dynamic Capabilities. Oxford: Oxford University Press.

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

Baumann, O., J. Schmidt, and N. Stieglitz
2019 "Effective search in rugged performance landscapes: A review and outlook." Journal of Management, 45: 285-318.

Brandenburger, A. M., and H. W. Stuart
1996 "Value-based business strategy." Journal of Economics and Management Strategy, 5: 5–24.

Brews, P. J., and C. L. Tucci
2004 "Exploring the structural effects of Internetworking." Strategic Management Journal, 25: 429–451

Brusoni S, L. Marengo, A. Prencipe, and M. Valente
2007 "The value and costs of modularity: A problem solving perspective." European Management Review, 4:121–132.

Brusoni, S., A. Prencipe, and K. Pavitt
2001 "Knowledge specialization, organizational coupling, and the boundaries of the firm: Why do firms know more than they make?" Administrative Science Quarterly, 46: 597-621.

Cennamo, C., and J. Santaló
2019 "Generativity tension and value creation in platform ecosystems." Organization Science, 30: 617–641

Christensen, C. M., F. F. Suarez, and J. M. Utterback
1998 "Strategies for survival in fast-changing industries." Management Science, 44: 207–220.

Colfer, L. J., and C. Y. Baldwin
2016 "The mirroring hypothesis: theory, evidence, and exceptions." Industrial and Corporate Change, 25(5), 709-738.

Conti, R., A. Gambardella, and E. Novelli
2019a "Specializing in generality: firm strategies when intermediate markets work." Organization Science, 30: 126–150.

Conti, R., A. Gambardella, and E. Novelli
2019b "Specializing in general purpose technologies as a firm long-term strategy." Industrial and Corporate Change, 28: 351–364.

Csaszar, F. A.
2018 "A note on how NK landscapes work." Journal of Organization Design, 7: 1-6.

Csaszar, F. A., and D. A. Levinthal
2016 "Mental representation and the discovery of new strategies." Strategic Management Journal, 37: 2031–2049.

Davis, J. P., K. M. Eisenhardt, and C. B. Bingham
2007 "Developing theory through simulation methods." Academy of Management Review, 32: 480-499.

Davis, J. P., K. M. Eisenhardt, and C. B. Bingham
2009 "Optimal structure, market dynamism, and the strategy of simple rules." Administrative Science Quarterly, 54: 413-452.

Dosi, G., D. A. Levinthal, L, and L. Marengo
2003 "Bridging contested terrain: Linking incentive-based and learning perspectives on organizational evolution." Industrial and Corporate Change, 12: 413–436.

Eggers, J. P., and K. F. Park
2018 "Incumbent adaptation to technological change: The past, present, and future of research on heterogeneous incumbent response." Academy of Management Annals, 12: 357-389.

Ethiraj, S. K., and D. A. Levinthal
2004a "Bounded rationality and the search for organizational architecture: An evolutionary perspective on the design of organizations and their evolvability." Administrative Science Quarterly, 49: 404- 437.

Ethiraj, S. K., and D. A. Levinthal
2004b "Modularity and innovation in complex systems." Management Science, 50: 159-173.

Ethiraj, S. K., and D. A. Levinthal
2009 "Hoping for A to Z while rewarding only A: Complex organizations and multiple goals." Organization Science, 20: 4–21.

Fang, C., and J. Kim
2018 "The power and limits of modularity: A replication and reconciliation." Strategic Management Journal, 39: 2547–2565.

Fine, C. H.
1998 Clockspeed: Winning Industry Control in the Age of Temporary Advantage. New York: Basic Books.

Ganco, M., R. Kapoor, and G. Lee
2020 "From rugged landscapes to rugged ecosystems: Structure of interdependencies and firms' innovative search." Academy of Management Review, 45: 646-674.

Gans, J., and M. D. Ryall
2017 "Value capture theory: A strategic management review." Strategic Management Journal, 38: 17–41.

Gavetti, G., and D. A. Levinthal
2000 "Looking forward and looking backward: Cognitive and experiential search." Administrative Science Quarterly, 45: 113-137

Gavetti, G., C. E. Helfat, and L. Marengo
2017 "Searching, shaping, and the quest for superior performance." Strategy Science, 2: 194-209.

Gavetti, G., D. A. Levinthal, and J. W. Rivkin
2005 "Strategy making in novel and complex worlds: The power of analogy." Strategic Management Journal, 26: 691-712.

Hannah, D. P., and K. M. Eisenhardt
2018 "How firms navigate cooperation and competition in nascent ecosystems." Strategic Management Journal, 39: 3163–3192.

Henderson, R. M., and K. B. Clark
1990 "Architectural innovation - the reconfiguration of existing product technologies and the failure of established firms." Administrative Science Quarterly, 35: 9-30.

Jacobides, M. G., S. Brusoni, and F. Candelon
2021 "The evolutionary dynamics of the artificial intelligence ecosystem." Strategy Science, 6: 412-435.

Jacobides, M. G., C. Cennamo, and A. Gawer
2018 "Towards a theory of ecosystems." Strategic Management Journal, 39: 2255-2276.

Jacobides, M. G., T. Knudsen, and M. Augier
2006 "Benefiting from innovation: Value creation, value appropriation and the role of industry architectures." Research Policy, 35: 1200-1221.

Jacobides, M. G., J. P. MacDuffie, and C. J. Tae
2016 "Agency, structure, and the dominance of OEMs: Change and stability in the automotive sector." Strategic Management Journal, 37: 1942-1967.

Kapoor, R.
2013 "Persistence of integration in the face of specialization: How firms navigated the

winds of disintegration and shaped the architecture of the semiconductor industry." Organization Science, 24:1195–2013.

Katila, R., and G. Ahuja
2002 "Something old, something new: A longitudinal study of search behavior and new product introduction." Academy of Management Journal, 45: 1183-1194.

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

Lanzolla, G., A. Lorenz, E. Miron-Spektor, M. A. Schilling, G. Solinas, and C. L. Tucci
2020 "Digital transformation: What is new if anything? Emerging patterns and management research." Academy of Management Discoveries, 6: 341–350.

Lawrence, P., and J. Lorsch
1967 Organization and Environment. Cambridge, MA: Harvard University Press.

Lenox, M. J., S. F. Rockart, and A. Y. Lewin
2007 "Interdependency, competition, and industry dynamics." Management Science, 53: 599–615.

Levinthal, D. A.
1997 "Adaptation on rugged landscapes." Management Science, 43: 934-950.

Levinthal, D. A., and J. G. March
1993 "The myopia of learning." Strategic Management Journal, 14: 95-112.

Levinthal, D. A., and M. Warglien
1999 "Landscape design: Designing for local action in complex worlds." Organization Science, 10: 342-357.

Li, C, and F. A. Csaszar
2019 "Government as landscape designer: a behavioral view of industrial policy." Strategy Science, 4: 175-192.

March, J. G.
1991 "Exploration and exploitation in organizational learning." Organization Science, 2: 71-87.

Marengo, L.
2020 "Organizational politics and complexity: Coase vs. Arrow, March, and Simon." Industrial and Corporate Change, 29: 163-181.

Marengo, L., G. Dosi, P. Legrenzi, and C. Pasquali
2000 "The structure of problem- solving knowledge and the structure of organizations." Industrial and Corporate Change, 9: 757-78

Marengo L, and C. Pasquali
2012 "How to get what you want when you do not know what you want: A model of incentives, organizational structure, and learning." Organization Science, 23: 1298–1310.

Murmann, J. P., and K. Frenken
2006 "Toward a systematic framework for research on dominant designs, technological innovations, and industrial change." Research Policy, 35: 925–952.

Nelson, R.R.
1994 "The co-evolution of technology, industrial structure, and supporting institutions." Industrial and Corporate Change, 3: 47–63.

Nelson, R. R., and B. N. Sampat
2001 "Making sense of institutions as a factor shaping economic performance." Journal of Economic Behavior and Organization, 44: 31–54.

Perrow, C.
1967 "A framework for the comparative analysis of organizations." American Sociological Review, 26: 194-208.

Pisano, G. P., and D. J. Teece
2007 "How to capture value from innovation: Shaping intellectual property and industry architecture." California Management Review, 50: 278–296.

Prencipe, A.
1997 "Technological capabilities and product evolutionary dynamics: A case study from the aero engine industry." Research Policy, 25: 1261-1276.

Rahmandad, H.
2019 "Interdependence, complementarity, and ruggedness of performance landscapes." Strategy Science, 4: 234-249.

Schilling, M. A.
2000 "Toward a general modular systems theory and its application to interfirm product modularity." Academy of Management Review, 25: 312-334.

Siggelkow, N.
2002 "Evolution toward fit." Administrative Science Quarterly, 47: 125-159.

Siggelkow, N., and D. A. Levinthal
2003 "Temporarily divide to conquer: Centralized, decentralized, and reintegrated organizational approaches to exploration and adaptation." Organization Science, 14: 650-669.

Siggelkow, N., and J. Rivkin
2006 "When exploration backfires: Unintended consequences of multi-level organizational search." Academy of Management Journal, 49: 779-795.

Teece, D. J.
1986 "Profiting from technological innovation: Implications for integration, collaboration, licensing, and public policy." Research Policy, 15: 285-305

Teece, D. J.
2000 Managing Intellectual Capital: Organizational, Strategic, and Policy Dimensions. Oxford: Oxford University Press.

Ulrich, K.
1995 “The role of product architecture in the manufacturing firm.” Research Policy, 24: 419–440.

Utterback, J. M., and W. J. Abernathy
1975 “A dynamic model of process and product innovation.” Omega, 3: 639–656.

Utterback, J. M., and Suarez, F. F.
1993 “Innovation, competition, and industry structure.” Research Policy, 22: 1-21.

Wolter, C., and F. M. Veloso
2008 “The effects of innovation on vertical structure: Perspectives on transaction costs and competences.” Academy of Management Review, 33: 586-605.

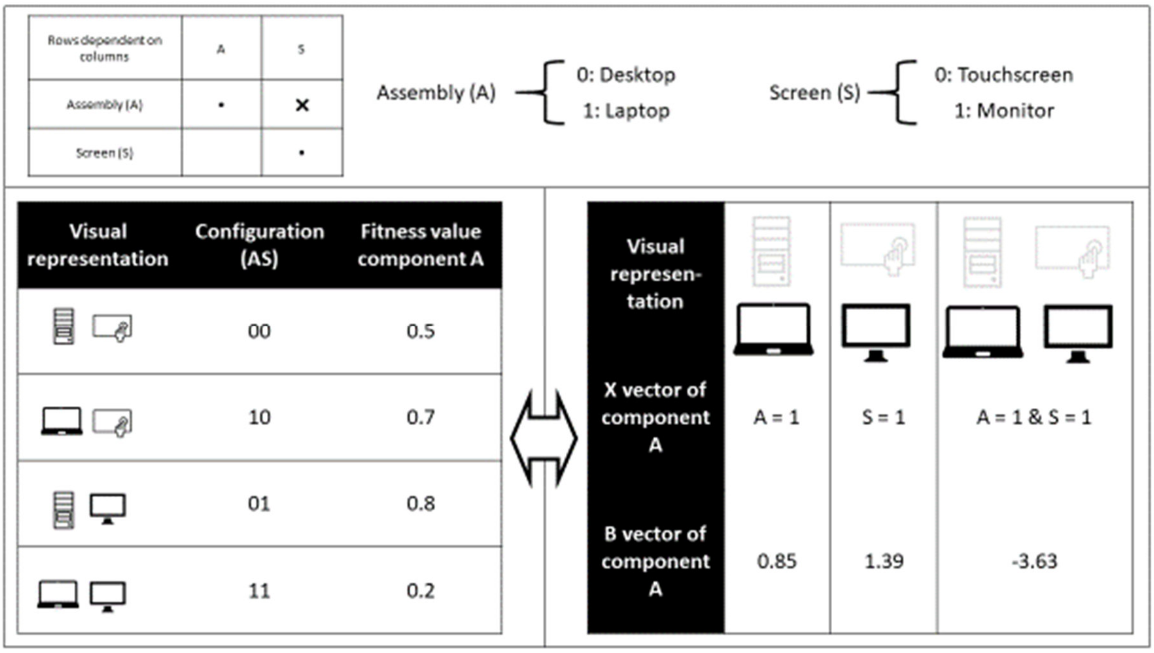
Yoo, Y., R. J. Boland Jr, K. Lyytinen, and A. Majchrzak
2012 “Organizing for innovation in the digitized world.” Organization Science, 23:1398–14

TABLE AND FIGURES

Table 1: Two types of redesign and their characteristics

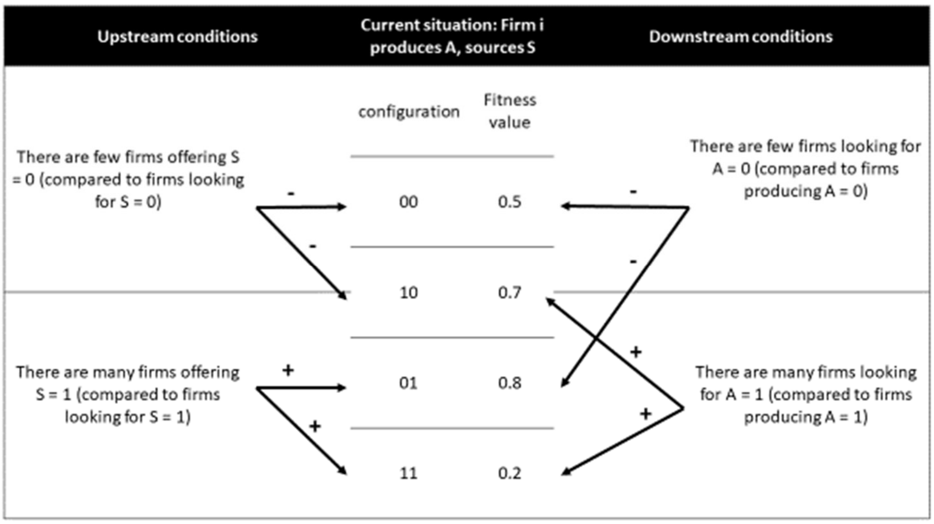
Type	Measure (coding)	Generalization (description)	Simulation observations	Characteristics of firms successfully using mechanism
Landscape redesign	Cases where the firm finds a redesign that improves performance, and does not improve due to any other adaptation mechanism	Firms finding a better architecture—i.e., a better functioning of a system given the chosen components. The current position on the technical fitness landscape is improved	Often available initially, found less frequently later on Performance improvements quickly dissipate	Upstream firms in stable environments Specialists (small span of control)—leveraging speed of searching small landscapes Firms with relatively high performance—landscape redesign offers incremental benefits
Ecosystem redesign	Cases where the firm finds a redesign that does not provide a performance improvement in itself, but does provide one in combination with another adaptation mechanism	Firms improve their performance by changing the “rules of the game” governing the interaction between organizational and technical fitness landscapes, due to a reshaping of the technical fitness landscape	Relatively infrequent Performance effects remain high over time	Downstream firms—systems integrators exploiting the benefits of mixing and matching Integrated firms (large span of control)—leveraging large search space of potential benefits Firms with relatively low performance—escaping bottlenecks

Figure 1: Illustration of fitness specification



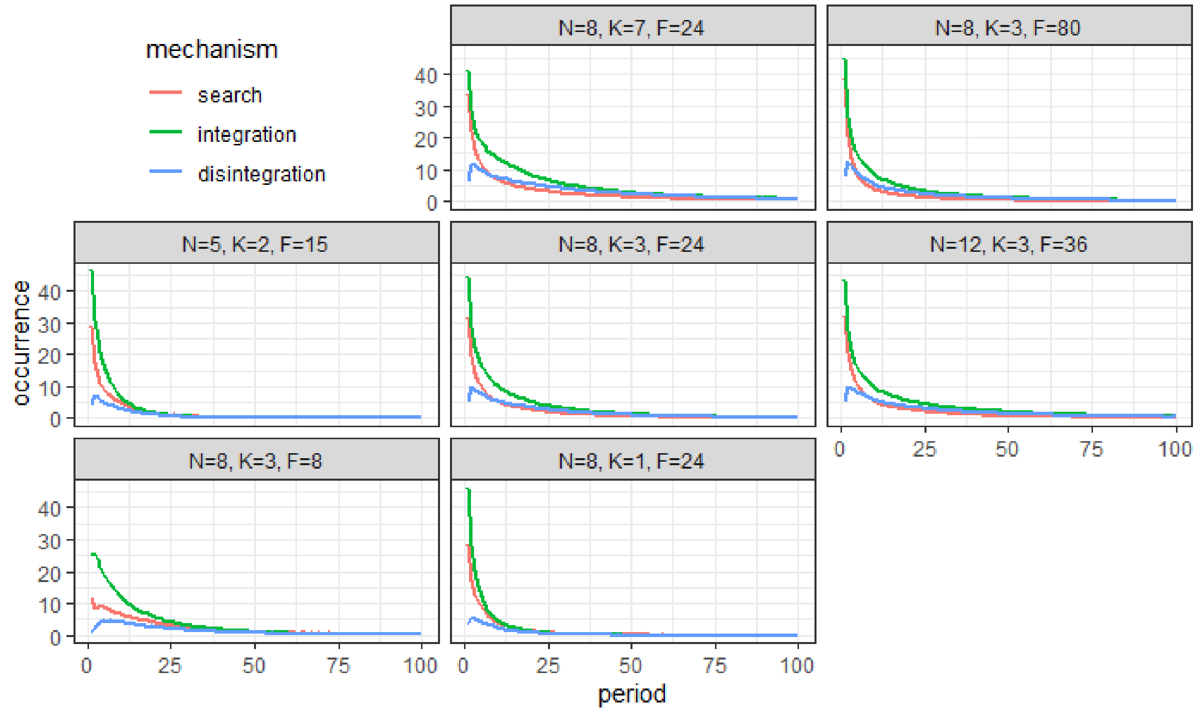
Note: The top panel displays the system architecture as a design structure matrix (DSM), where the performance of assembly (A) is dependent on the configuration of both assembly (A) and screen (S) (denoted by the X in the matrix). The left panel displays a "standard NK" mapping of configurations to fitness values. The right panel displays our functional specification (resulting in the same ultimate fitness values).

Figure 2: Illustration of dependence of configuration fitness on up- and downstream conditions



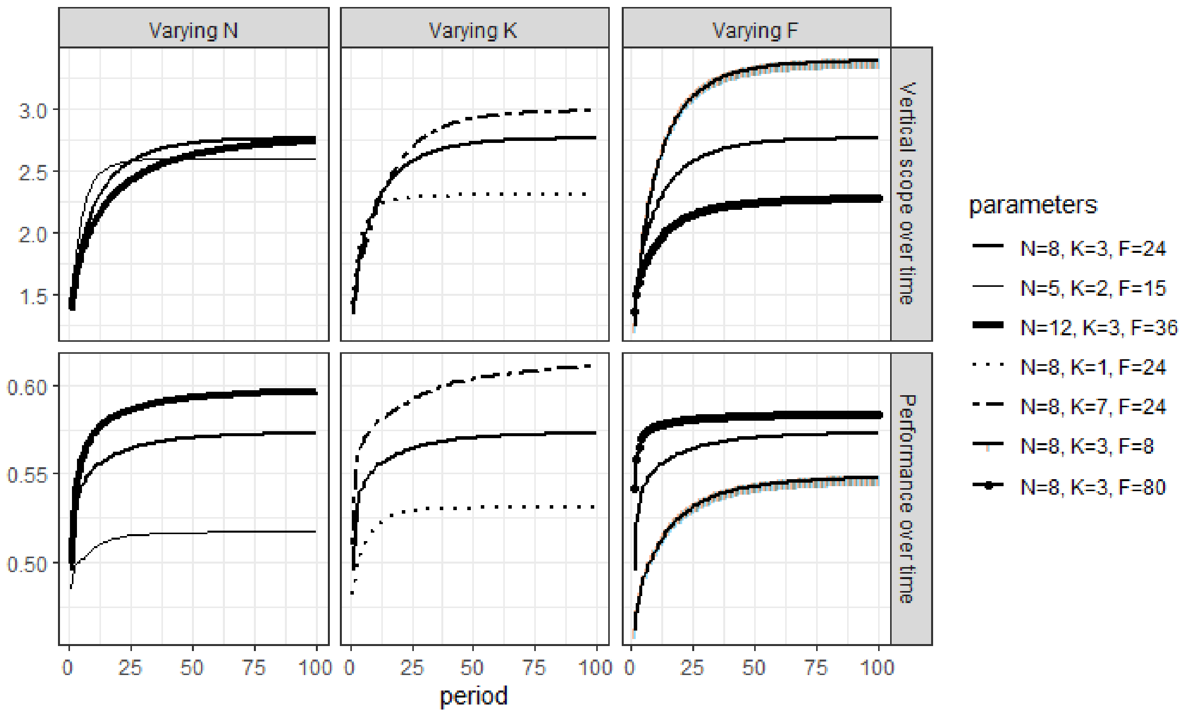
Note: The central column displays the "raw" fitness values of configurations of the assembly (A), based on the states of assembly (A) and screen (S). The left panels display how conditions in the upstream screen manufacturing segment affect the performance of assembly firms employing different configurations. The right pane displays how conditions in the downstream market affect the performance of assembly firms employing different configurations.

Figure 3: Model dynamics of the baseline industry NK model



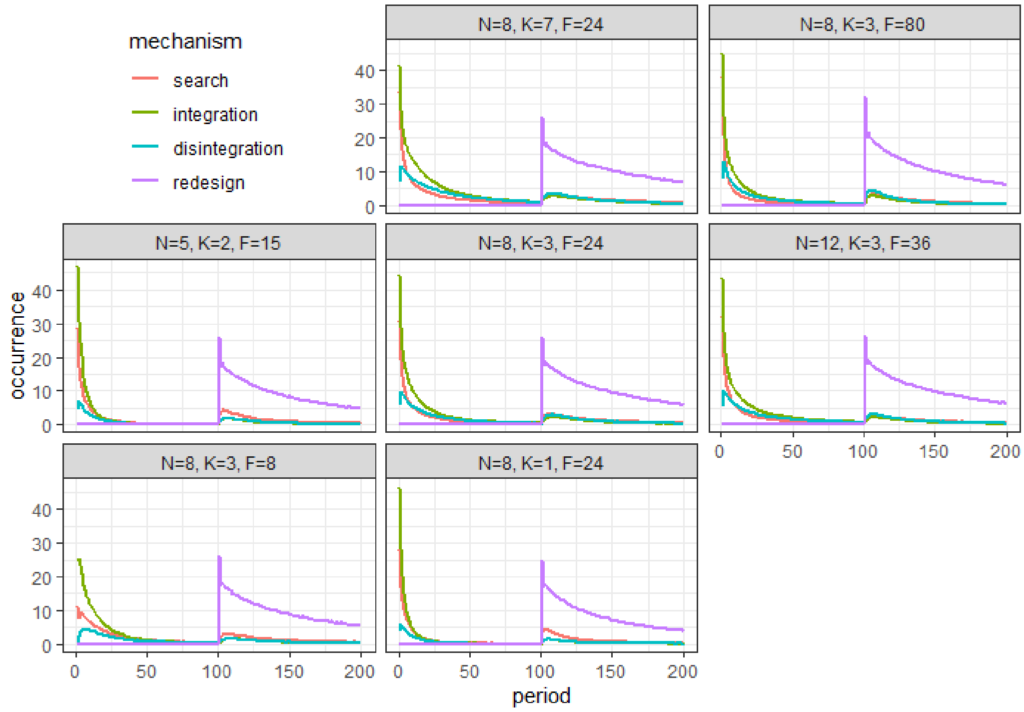
Note: The center panel indicates a reference parameter set. Surrounding panels vary parameters along N (left to right), K (bottom to top), and F (bottom-left to top-right). Colored lines indicate per mechanism what percentage of firms finds performance increases (i.e., adapts successfully) using the indicated search function.

Figure 4: Temporal dynamics of vertical scope (top) and performance (bottom) in baseline industry NK model



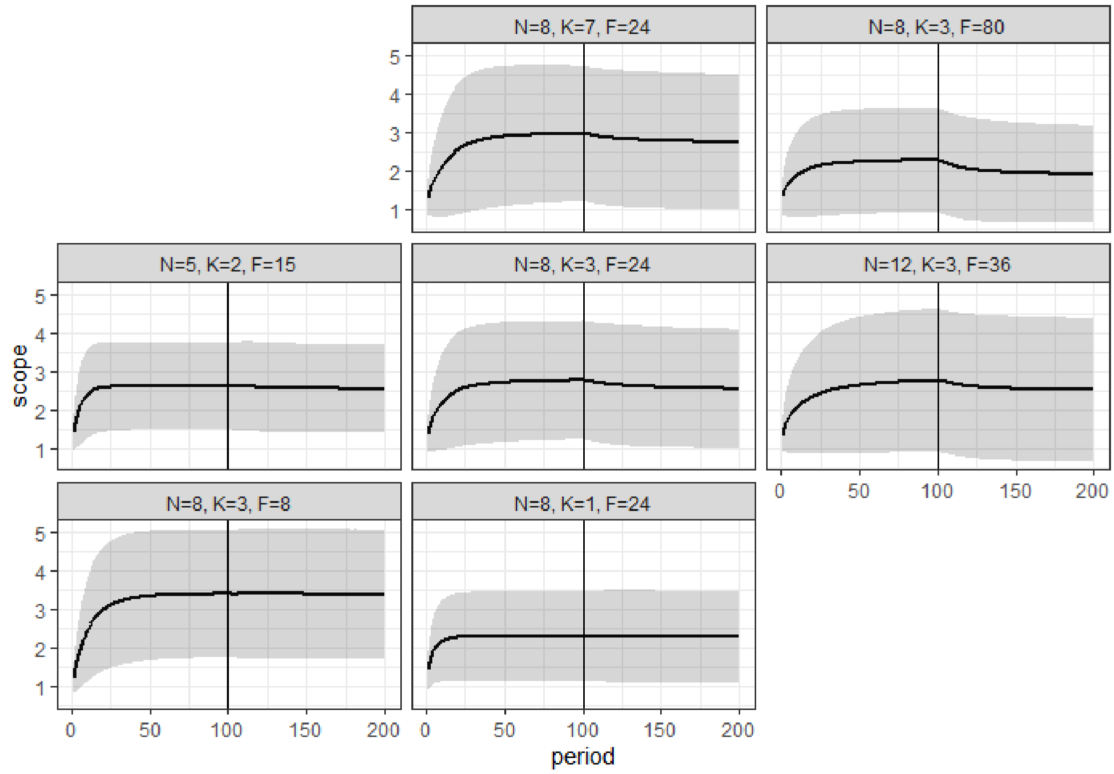
Note: Reference parameter set is included in each panel.

Figure 5: Model dynamics of the industry NK model with redesign possibilities



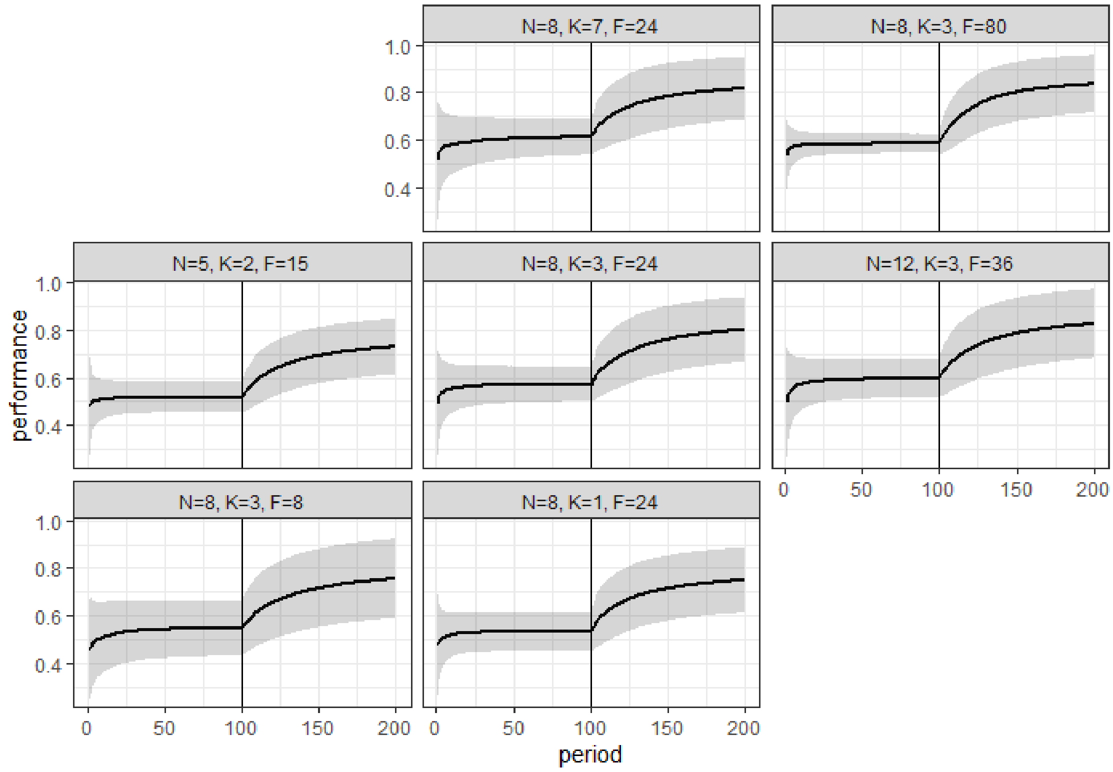
Note: The center panel indicates the reference parameter set. Surrounding panels vary parameters along N (left to right), K (bottom to top), and F (bottom-left to top-right). Colored lines indicate per mechanism which percentage of firms finds performance increases (i.e., adapts successfully) using the indicated search function.

Figure 6: Temporal scope dynamics surrounding the availability of redesign



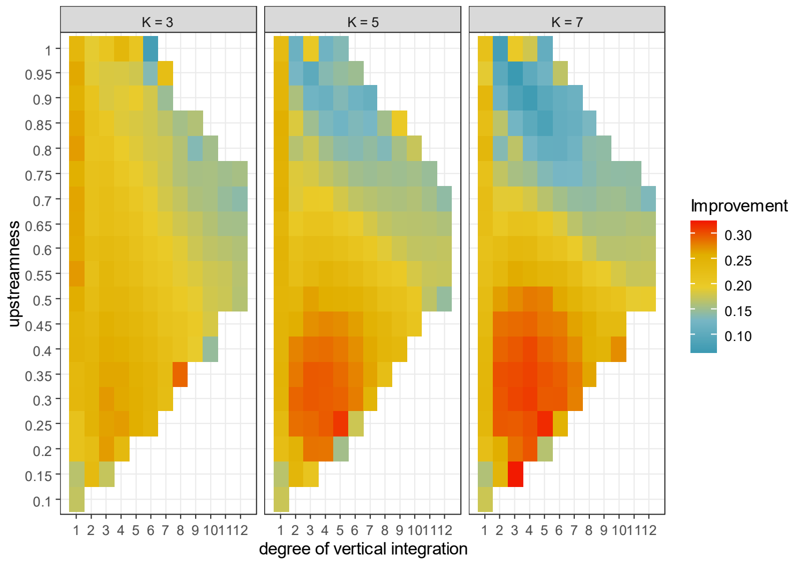
Note: The center panel indicates the reference parameter set. Surrounding panels vary parameters along N (left to right), K (bottom to top), and F (bottom-left to top-right). Grey shaded areas display standard deviation. Vertical lines indicate when redesign becomes available to firms.

Figure 7: Temporal performance dynamics surrounding the availability of redesign



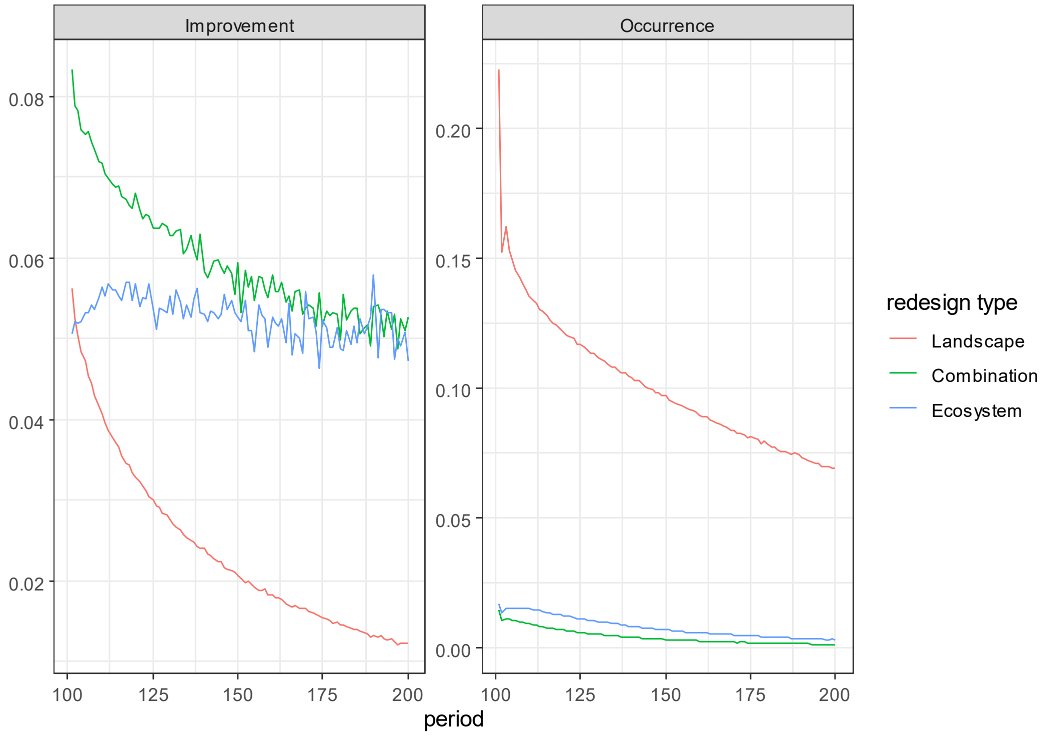
Note: The center panel indicates the reference parameter set. Surrounding panels vary parameters along N (left to right), K (bottom to top), and F (bottom-left to top-right). Grey shaded areas display standard deviation. Vertical lines indicate when redesign becomes available to firms.

Figure 8: Firm performance increase after the availability of redesign, by positioning



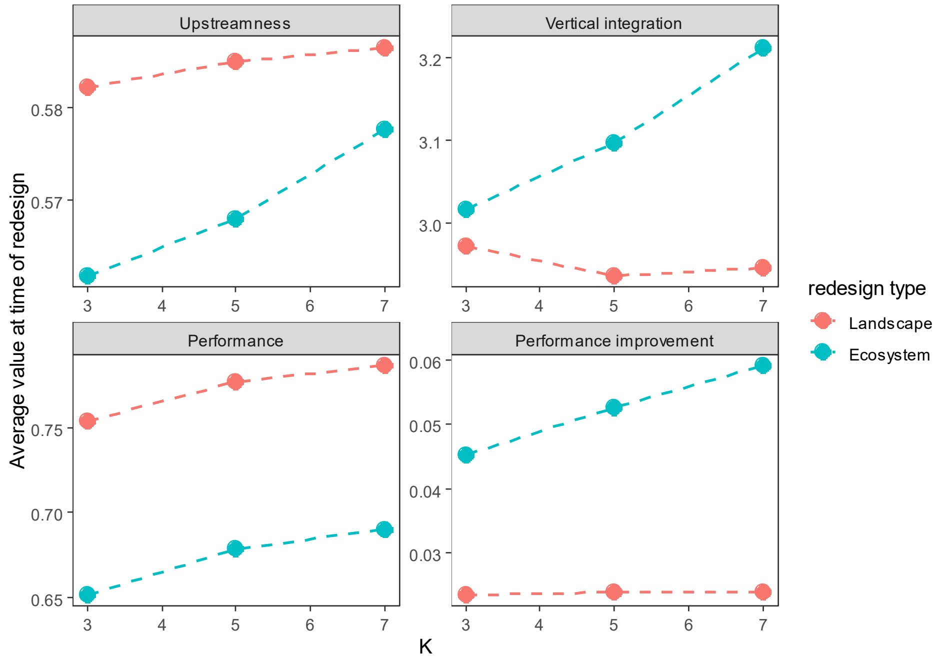
Note: N = 12, F = 24, Panel display varying K. The color scale indicates average performance improvement. Upstreamness is the average distance to the downstream market (ranging from 1 to N) of all components produced by the firm, normalized by dividing over the maximum distance of the generated supply chain.

Figure 9: Characteristics of different redesign types



Note: N = 8, K = 3, F = 24. "Landscape" redesigns confer benefit purely by increasing the fitness value associated with the current configuration. "Ecosystem" redesigns confer benefits when firms simultaneously find additional adaptations (e.g., integration). "Combination" redesigns confer benefits in both ways. Y-axes show the average performance gain by the redesign (left), and the percentage of firms employing that type (right).

Figure 10: Characteristics of firms by redesign type



Note: N = 8, F = 24. "Landscape" redesigns confer benefit purely by increasing the fitness value associated with the current configuration. "Ecosystem" redesigns confer benefits when firms simultaneously find additional adaptations (e.g., integration). Panels "Upstreamness," "Vertical Integration," and "Performance" refer to firm characteristics before the successful redesign. "Performance improvement" refers to the performance gain by the redesign.

APPENDIX A: DESCRIPTION OF MODEL MECHANICS

Pseudo-code for model mechanisms

Generation of environment, NK structure and firms

Interaction structure

1. Create N components.
2. For each of the N components, do the following:
    a. Draw a random number between (and including) 1 and K. This is the effective K.
    b. Add a dependence effect on itself.
    c. Assign effective K dependences on random other components.
    d. For each of these added dependences, check if cycles of dependence have emerged.[13] If so, drop that dependence.
3. The result of step 2 is the dependence structure and defines the production chain.

Agents

4. Create F agents.
5. Assign the first N agents each to one component.
6. The other agents are randomly assigned one component.
7. Assign each agent a random configuration string that is feasible (i.e., each necessary input component is either made by the focal agent or is made by at least one other agent in the designated state AND not used by that other agent itself in all downstream components).[14]

Fitness landscape

8. For each of N components, generate 2^[dependences of that component]-1 random numbers from the uniform random distribution (-1,1).
9. The result is a N x 2^K-1 matrix of numbers. Each row is a B vector. The full matrix is a B matrix.
10. Assign a copy of the generated B matrix to each of F agents.

Downstream demand

11. Generate a downstream demand of value F / 2N for each component state of components that are fully downstream.[15]

Model iterations (baseline):

[13] A cycle of dependence exists if a component is dependent on other components that (directly or indirectly) are dependent on it. In other words, we require the possibility to arrange our dependence structure in an interaction matrix without interactions above the diagonal.
[14] Note that an input component can be (and often is) used for multiple downstream components. If an agent making the input component also makes all downstream components that use it, this input component is not available for other firms; “designated” here means that we only evaluate whether another agent makes, e.g., component 6 in state [0] if the string calls for component 6 in state [0].
[15] “Fully downstream” means that no other component is dependent on this component. “Component state” means, e.g., component 6 in state [0]

In each time period, select a random order in which to execute search, integration, disintegration, and execute them in that order.

Search:

Line up agents in random order, and for each agent do the following:

1. Calculate current performance.
2. Generate a random different configuration string.
3. Check if the string is feasible (i.e., each necessary input component is either made by the agent or is made by at least one agent in the designated state AND not used by that agent itself in all downstream components). If not, go back to 2.
4. Calculate performance of the new string. If the found performance is higher than current performance, the agent adopts the new string. If not, there is no change.

Integration:

Line up agents in random order, and for each agent do the following:

1. Calculate current performance.
2. Make a list of all components that are adjacent to the components that are made by the agent, and select one at random to add to the components made by the agent.
3. Generate a random configuration string.
4. Check if the string is feasible (i.e., each necessary input component is either made by the agent or is made by at least one agent in the designated state AND not used by that agent itself in all downstream components). If not, go back to 3.
5. Calculate the performance of the new string given the scope of the agent as determined in 2. If the found performance is higher than current performance, the agent adopts the new string and new made component set. If not, there is no change.

Disintegration:

Line up agents in random order, and for each agent do the following if its scope is larger than 1:

1. Calculate current performance.
2. Select a made component at random to drop.
3. Check if the reduced set of components is connected[16]. If not, go back to 2.
4. Generate a random configuration string.
5. Check if the string is feasible (i.e., each necessary input component is either made by the focal agent or is made by at least one other agent in the designated state AND not used by that other agent itself in all downstream components). If not, go back to 4.
6. Calculate the performance of the new string given the scope of the agent as determined in 2. If the found performance is higher than current performance, the agent adopts the new string and new made component set. If not, there is no change.

The model extension of redesign:

Redesign is enabled from time period 101 to 200. When it is enabled, the mechanism is enabled at the end of every time period, after the set of search, integration, and disintegration mechanisms.

Line up agents in random order, and for each agent do the following if it has not had a successful adaptation in the current period or the last:

1. Calculate current performance. This is the baseline performance.

[16] E.g., if a value chain consists of components 1 → 2 → 3 and an agent is fully integrated, it can only drop components 1 and 3, not component 2.

2. Identify all B matrix coefficients for all made components of the agent, and redraw 20% (rounded up) of this set of coefficients, chosen at random, from the uniform random distribution (-1,1). This is the redesigned architecture.
3. Calculate performance with the redesigned architecture. This is the redesign performance.
4. If the redesign performance is higher than the baseline performance, the agent adopts the redesigned architecture, and the baseline performance takes the value of the redesign performance.
5. Regardless of the outcome of step 4, line up the mechanisms search, integration, and disintegration in random order, and execute them as described above with the following notes:
    a. In these mechanisms, performance is calculated with the redesigned architecture, and performance is compared to the baseline performance.
    b. If any mechanism leads to improved performance, the agent adopts the change including the redesigned architecture, and the baseline performance takes the value of the found improved performance.[17]

Computation of performance:

- If an agent produces multiple components, the performance is the average of the performance generated per component.
- The performance of each component is based on two numbers: 1) the fitness value, which follows directly from the mapping of configurations to the fitness landscape (i.e. our B vector); 2) the ratios of upstream and downstream competitors and suppliers resp. buyers associated with the inputs necessary (upstream) and component made (downstream). Performance is computed by multiplying (1) with the factor log2(x+1), where x is the ratio found in (2).
- If a component has both an upstream and downstream ratio (i.e., the agent needs to source inputs for it AND it is passed on to a downstream market), the ratio used is the product of the upstream and downstream ratios (for examples, see the computation example below).
- If for a component multiple inputs are necessary, the upstream ratio is computed as follows:[18]
    1. Calculate the upstream ratio for each input. Inputs made by the agent itself are counted as 1.
    2. Take the natural logarithm of each input ratio.
    3. Take the average of the values obtained in (2). This is the log-ratio.
    4. The upstream ratio is exp(log-ratio).

[17] Note that we code as “landscape redesign” those events where agents find an improvement in step 4 AND NOT in step 5. We code as “ecosystem redesigns” those events where agents find an improvement in step 5 AND NOT in step 4. We code as “combination redesigns” those events where agents find an improvement in step 4 AND in step 5.

[18] Note that the computational example does not feature this computation step, as components never require more than one input in our model. However, industries with higher K will employ this computation algorithm. The reasoning for the log-transformation over taking the average or product of input ratios is that it combines the attractive features of both. It averages out values such that extreme values are mitigated by average values. Additionally, it evens out ratios as taking the product would do (e.g., ratios of ½ and 2 average out to 1).

Computational example of the model

Generating environment, NK structure, and firms

Steps 1–3

For simplicity, we construct a product system with N = 3, K = 1, F = 6, with interactions as follows. Note that an ‘x’ in the interaction structure signals a dependence of the row component on the column component.

	1	2	3
1	x	x	
2		x	x
3			x

This dependence structure leads to the following production chain, with component 1 as downstream and component 3 as upstream

3 → 2 → 1

Steps 4–7

Six agents are generated and divided as follows. Note that firms are assigned a full configuration string of length N for simplicity, but only a subset of this string is relevant for the performance of the agent. Relevant bits in the string are those on which any of the made components of the firm are dependent (following directly from the dependence structure):

1. for an agent making component 1, the configurations of components 1 and 2 are relevant
2. for an agent making component 2, the configurations of components 2 and 3 are relevant
3. for an agent making component 3, only the configuration of component 3 is relevant

Agent	Set of components made	Configuration
1	1	**00**1
2	2	1**00**
3	3	10**1**
4	3	10**0**
5	3	00**0**
6	1	**10**1

Steps 8–10

B matrix for all agents:

Component C1	C1 == 1	C2 == 1	C1 == 1 & C2 == 1
	0.3409	-0.7192	0.9365
Component C2	C2 == 1	C3 == 1	C2 == 1 & C3 == 1
	-0.1654	-0.6038	-0.3732
Component C3	C3 == 1		
	0.1174		

Computation of fitness:

Agent	Configuration	Makes	Needs (upstream)	Offers (downstream)
1	**00**1	C1[0]	C2[0]	C1[0]
2	1**00**	C2[0]	C3[0]	C2[0]
3	10**1**	C3[1]	-	C3[1]
4	10**0**	C3[0]	-	C3[0]
5	00**0**	C3[0]	-	C3[0]
6	**10**1	C1[1]	C2[0]	C1[1]

Agent	C	Fitness	Ratios buyers/suppliers	Performance
1	1	1 / (1 + exp(-(0))) = 0.5	Upstream ratio = 1/2: 1 agent offer C2 [0] 2 agents need C2 [0] Downstream ratio = 1: 1 agent offers C1[0] Demand is F/N/2 = 6/3/2 = 1	0.5 * log2(1/2*1 + 1) = 0.29
2	2	1 / (1 + exp(-(0))) = 0.5	Upstream ratio = 2: 2 agents offer C3[0] 1 agent needs C3[0] Downstream ratio = 2: 1 agent offers C2[0] 2 agents need C2[0]	0.5 * log2(2*2 + 1) = 1.16
3	3	1 / (1 + exp(-(0.1174))) = 0.53	Downstream ratio = 0: 1 agent offers C3 [1] 0 agents need C3 [1]	0.53 * log2(0 + 1) = 0
4	3	1 / (1 + exp (-(0))) = 0.5	Downstream ratio = 1/2: 2 agent offers C3[0] 1 agent needs C3[0]	0.5 * log2(1/2 + 1) = 0.29
5	3	1 / (1 + exp (-(0))) = 0.5	Downstream ratio = 1/2: 2 agent offers C3[0] 1 agent needs C3[0]	0.5 * log2(1/2 + 1) = 0.29
6	1	1 / (1 + exp(-(0.3409))) = 0.58	Upstream ratio = 1/2: 1 agent offer C2 [0] 2 agents need C2 [0] Downstream ratio = 1: 1 agent offers C1[1] Demand is F/N/2 = 6/3/2 = 1	0.58 * log2(1/2*1 + 1) = 0.34

Successful search:

1. agent 3 has performance 0
2. agent 3 finds random different configuration string 100
3. this string is feasible (component 3 is made by agent 3, no input components are needed)
4. the performance is 0.21, so agent 3 adopts the string

Agent	C	Fitness	Ratios buyers/suppliers	Performance
3	3	1 / (1 + exp(-(0))) = 0.5	Downstream ratio = 1/3: 3 agents offer C3[0] 1 agent needs C3[0]	0.5 * log2(1/3 + 1) = 0.21

Note: This successful search also changes the performance of agents 4 and 5 negatively (as they face increased competition), and the performance of agent 2 positively (as it faces an increase in supply)

Agent	C	Fitness	Ratios buyers/suppliers	Performance
1	1	1 / (1 + exp(-(0))) = 0.5	Upstream ratio = 1/2: 1 agent offer C2 [0] 2 agents need C2 [0] Downstream ratio = 1: 1 agent offers C1[0] Demand is F/N/2 = 6/3/2 = 1	0.5 * log2(1/2*1 + 1) = 0.29
2	2	1 / (1 + exp(-(0))) = 0.5	Upstream ratio = 3: 3 agents offer C3[0] 1 agent needs C3[0] Downstream ratio = 2: 1 agent offers C2[0] 2 agents need C2[0]	0.5 * log2(3*2 + 1) = 1.40
3	3	1 / (1 + exp(-(0))) = 0.5	Downstream ratio = 1/3: 3 agents offer C3[0] 1 agent needs C3[0]	0.5 * log2(1/3 + 1) = 0.21
4	3	1 / (1 + exp(-(0))) = 0.5	Downstream ratio = 1/3: 3 agents offer C3[0] 1 agent needs C3[0]	0.5 * log2(1/3 + 1) = 0.21
5	3	1 / (1 + exp(-(0))) = 0.5	Downstream ratio = 1/3: 3 agents offer C3[0] 1 agent needs C3[0]	0.5 * log2(1/3 + 1) = 0.21
6	1	1 / (1 + exp(-(0.3409))) = 0.58	Upstream ratio = 1/2: 1 agent offer C2 [0] 2 agents need C2 [0] Downstream ratio = 1: 1 agent offers C1[1] Demand is F/N/2 = 6/3/2 = 1	0.58* log2(1/2*1 + 1) = 0.34

Successful integration:

1. Agent 4 has performance 0.21
2. Agent 4 considers integrating into component 2
3. Agent 4 finds configuration string **100** (note: the same it had before)
4. The string is feasible (components 2 and 3 are made by agent 3)
5. The performance is 0.5, so the agent 4 adopts the change.

<table>
<tr><th>Agent</th><th>C</th><th>Fitness</th><th>Ratios buyers/suppliers</th><th>Performance</th></tr>
<tr><td rowspan="2">4</td><td>2</td><td>1 / (1 + exp(-(0)))
= 0.5</td><td>Upstream ratio: -
(it makes its own input)
Downstream ratio = 1:
2 agents offer C2[0]
2 agents need C2[0]</td><td rowspan="2">C2:
0.5 * log2(1 + 1) = 0.5
C3:
0.5 * 1 = 0.5
Average = 0.5</td></tr>
<tr><td>3</td><td>1 / (1 + exp(-(0)))
= 0.5</td><td>Downstream ratio = -
(it uses its own output)</td></tr>
</table>

Note: This successful integration also changes the performance of agent 2 negatively (as it faces a decrease in supply *and* an increase in competition), the performance of agents 1 and 6 positively (as they face an increase in supply) and the performance of agents 3 and 5 positively (as they face a decrease in competition)

Agent	Configuration	Makes	Needs (upstream)	Offers (downstream)
1	**00**1	C1[0]	C2[0]	C1[0]
2	1**00**	C2[0]	C3[0]	C2[0]
3	10**1**	C3[1]	-	C3[1]
4	1**00**	C2[0] C3[0]	-	C2[0]
5	00**0**	C3[0]	-	C3[0]
6	**10**1	C1[1]	C2[0]	C1[1]

Agent	C	Fitness	Buyer/supplier ratios	Performance
1	1	1 / (1 + exp(-(0))) = 0.5	Upstream ratio = 1: 2 agents offer C2 [0] 2 agents need C2 [0] Downstream ratio = 1: 1 agent offers C1[0] Demand is F/N/2 = 6/3/2 = 1	0.5 * log2(1*1 + 1) = 0.5
2	2	1 / (1 + exp(-(0))) = 0.5	Upstream ratio = 2: 2 agents offer C3[0] 1 agent needs C3[0] Downstream ratio = 1: 2 agent offers C2[0] 2 agents need C2[0]	0.5 * log2(2*1 + 1) = 0.79
3	3	1 / (1 + exp(-(0))) = 0.5	Downstream ratio = 1/2: 2 agents offer C3[0] 1 agent needs C3[0]	0.5 * log2(1/2 + 1) = 0.29
4	2	1 / (1 + exp(-(0))) = 0.5	Upstream ratio: - (it makes its own input) Downstream ratio = 1: 2 agents offer C2[0] 2 agents need C2[0]	C2: 0.5 * log2(1 + 1) = 0.5 C3: 0.5 * 1 = 0.5 Average = 0.5
	3	1 / (1 + exp(-(0))) = 0.5	Downstream ratio = - (it uses its own output)	
5	3	1 / (1 + exp(-(0))) = 0.5	Downstream ratio = 1/2: 2 agents offer C3[0] 1 agent needs C3[0]	0.5 * log2(1/2 + 1) = 0.29
6	1	1 / (1 + exp(- (0.3409))) = 0.58	Upstream ratio = 1: 2 agents offer C2 [0] 2 agents need C2 [0] Downstream ratio = 1: 1 agent offers C1[1] Demand is F/N/2 = 6/3/2 = 1	0.58 * log2(1*1 + 1) = 0.58

Successful disintegration:

From the following state of the environment:

Agent	Configuration	Makes	Needs (upstream)	Offers (downstream)
1	**00**1	C1[0]	C2[0]	C1[0]
2	**100**	C1[1] C2[0] C3[0]		C1[1]
3	**111**	C1[1] C2[1]	-	C1[1]

		C3[1]		
4	1**00**	C2[0] C3[0]	-	C2[0]
5	0**00**	C2[0] C3[0]	-	C2[0]
6	**10**1	C1[1]	C2[0]	C1[1]

Agent	C	Fitness	Buyer/supplier ratios	Performance
1	1	1 / (1 + exp(-(0))) = 0.5	Upstream ratio = 1: 2 agents offer C2 [0] 2 agents need C2 [0] Downstream ratio = 1: 1 agent offers C1[0] Demand is F/N/2 = 6/3/2 = 1	0.5 * log2(1*1 + 1) = 0.5
2	1	1 / (1 + exp(-(0.3409))) = 0.58	Upstream ratio = - (agent makes its own input) Downstream ratio 1/3: 3 agents offer C1[1] Demand is F/N/2 = 6/3/2 = 1	C1: 0.58 * log2(1/3 + 1) = 0.24 C2: 0.5 C3: 0.5 Average = 0.41
	2	1 / (1 + exp(-(0))) = 0.5	Upstream ratio = - (agent makes its own input) Downstream ratio = - (agent uses its own output)	
	3	1 / (1 + exp(-(0))) = 0.5	Downstream ratio = - (agent uses its own output)	
3	1	1 / (1 + exp(-(0.3409 – 0.7192 + 0.9365))) = 0.64	Upstream ratio = - (agent makes its own input) Downstream ratio 1/3: 3 agents offer C1[1] Demand is F/N/2 = 6/3/2 = 1	C1: 0.64 * log2(1/3 + 1) = 0.27 C2: 0.24 C3: 0.53 Average = 0.35
	2	1 / (1 + exp(-(-0.1654 – 0.6038 – 0.3732))) = 0.24	Upstream ratio = - (agent makes its own input) Downstream ratio = - (agent uses its own output)	
	3	1 / (1 + exp(-(0.1174))) = 0.53	Downstream ratio = - (agent uses its own output)	
4	2	1 / (1 + exp(-(0))) = 0.5	Upstream ratio: - (agent makes its own input) Downstream ratio = 1: 2 agents offer C2[0] 2 agents need C2[0]	C2: 0.5 * log2(1 + 1) = 0.5 C3: 0.5 * 1 = 0.5 Average = 0.5
	3	1 / (1 + exp(-(0))) = 0.5	Downstream ratio = - (agent uses its own output)	
5	2	1 / (1 + exp(-(0))) = 0.5	Upstream ratio: - (agent makes its own input) Downstream ratio = 1: 2 agents offer C2[0] 2 agents need C2[0]	C2: 0.5 * log2(1 + 1) = 0.5 C3: 0.5 * 1 = 0.5 Average = 0.5
	3	1 / (1 + exp(-(0))) = 0.5	Downstream ratio = - (agent uses its own output)	
6	1	1 / (1 + exp(- (0.3409))) = 0.58	Upstream ratio = 1: 2 agents offer C2 [0] 2 agents need C2 [0] Downstream ratio = 1/3: 3 agents offer C1[1] Demand is F/N/2 = 6/3/2 = 1	0.58 * log2(1/3 + 1) = 0.24

1. Agent 2 has performance 0.41
2. Agent 2 considers dropping component 1
3. The reduced set is still connected (3 is an input to 2)
4. Agent 2 finds configuration string 1**00** (note: the same it had before)
5. The string is feasible (components 2 and 3 are made by agent 2)
6. The new performance 0.43, so agent 2 adopts the change

Agent	C	Fitness	Buyer/supplier ratios	Performance
2	2	1 / (1 + exp(-(0))) = 0.5	Upstream ratio = - (agent makes its own input) Downstream ratio = 2/3 3 agents offer C2[0] 2 agents need C2[0]	C2: 0.5 * log2(2/3 + 1) = 0.37 C3: 0.5 Average = 0.43
	3	1 / (1 + exp(-(0))) = 0.5	Downstream ratio = - (agent uses its own output)	

Note: This successful integration also changes the performance of agent 1 positively (as it faces an increase in supply), the performance of agent 6 positively (as it faces an increase in supply *and* a decrease in competition), the performance of agent 3 positively (as it faces a decrease in competition), the performance of agents 4 and 5 negatively (as they face an increase in competition)

Agent	Configuration	Makes	Needs (upstream)	Offers (downstream)
1	**00**1	C1[0]	C2[0]	C1[0]
2	1**00**	C2[0] C3[0]		C2[0]
3	**111**	C1[1] C2[1] C3[1]	-	C1[1]
4	1**00**	C2[0] C3[0]	-	C2[0]
5	0**00**	C2[0] C3[0]	-	C2[0]
6	**10**1	C1[1]	C2[0]	C1[1]

Agent	C	Fitness	Buyer/supplier ratios	Performance
1	1	1 / (1 + exp(-(0))) = 0.5	Upstream ratio = 3/2: 3 agents offer C2 [0] 2 agents need C2 [0] Downstream ratio = 1: 1 agent offers C1[0] Demand is F/N/2 = 6/3/2 = 1	0.5 * log2(3/2*1 + 1) = 0.66
2	2	1 / (1 + exp(-(0))) = 0.5	Upstream ratio = - (agent makes its own input) Downstream ratio = 2/3 3 agents offer C2[0] 2 agents need C2[0]	C2: 0.5 * log(2/3 + 1) = 0.37 C3: 0.53 Average = 0.43
	3	1 / (1 + exp(-(0))) = 0.5	Downstream ratio = - (agent uses its own output)	
3	1	1 / (1 + exp(-(0.3409 – 0.7192 + 0.9365))) = 0.64	Upstream ratio = - (agent makes its own input) Downstream ratio 1/2: 2 agents offer C1[1] Demand is F/N/2 = 6/3/2 = 1	C1: 0.64 * log2(1/2 + 1) = 0.37 C2: 0.24 C3: 0.53 Average = 0.38
	2	1 / (1 + exp(-(-0.1654 – 0.6038 – 0.3732))) = 0.24	Upstream ratio = - (agent makes its own input) Downstream ratio = -	

			(agent uses its own output)	
	3	1 / (1 + exp(-(0.1174))) = 0.53	Downstream ratio = - (agent uses its own output)	
4	2	1 / (1 + exp(-(0))) = 0.5	Upstream ratio: - (agent makes its own input) Downstream ratio = 2/3: 3 agents offer C2[0] 2 agents need C2[0]	C2: 0.5 * log2(2/3 + 1) = 0.37 C3: 0.5 * 1 = 0.5 Average = 0.43
	3	1 / (1 + exp(-(0))) = 0.5	Downstream ratio = - (agent uses its own output)	
5	2	1 / (1 + exp(-(0))) = 0.5	Upstream ratio: - (agent makes its own input) Downstream ratio = 1: 3 agents offer C2[0] 2 agents need C2[0]	C2: 0.5 * log2(2/3 + 1) = 0.37 C3: 0.5 * 1 = 0.5 Average = 0.43
	3	1 / (1 + exp(-(0))) = 0.5	Downstream ratio = - (agent uses its own output)	
6	1	1 / (1 + exp(- (0.3409))) = 0.58	Upstream ratio = 1: 3 agents offer C2 [0] 2 agents need C2 [0] Downstream ratio = 1/3: 3 agents offer C1[1] Demand is F/N/2 = 6/3/2 = 1	0.58 * log2(3/2 * 1/3 + 1) = 0.47

Successful landscape redesign

From the situation as above, agent 3 finds a successful landscape redesign:

1. Current performance is 0.38. This is the baseline performance.
2. Agent 3 makes all components, so it deals with 7 B matrix components. 2 are redrawn as follows. This is the redesigned architecture.

B matrix for agent 3 (changes in bold):

Component C1	C1 == 1	C2 == 1	C1 == 1 & C2 == 1
	0.3409	-0.7192	0.9365
Component C2	C2 == 1	C3 == 1	C2 == 1& C3 == 1
	-0.1654	**-0.4430**	-0.3732
Component C3	C3 == 1		
	0. 3525		

3. The redesign performance is 0.41.
4. Agent 3 adopts the redesigned architecture.

Agent	C	Fitness	Buyer/supplier ratios	Performance
3	1	1 / (1 + exp(-(0.3409 – 0.7192 + 0.9365))) = 0.64	Upstream ratio = - (agent makes its own input) Downstream ratio 1/2: 2 agents offer C1[1] Demand is F/N/2 = 6/3/2 = 1	C1: 0.64 * log2(1/2 + 1) = 0.37 C2: 0.27 C3: 0.59 Average = 0.41
	2	1 / (1 + exp(-(-0.1654 – **0.4430** – 0.3732))) = 0.27	Upstream ratio = - (agent makes its own input) Downstream ratio = - (agent uses its own output)	
	3	1 / (1 + exp(-(**0.3525**))) = 0.59	Downstream ratio = - (agent uses its own output)	

Successful ecosystem redesign:

Consider the following situation:

Agent	Configuration	Makes	Needs (upstream)	Offers (downstream)
1	**00**1	C1[0]	C2[0]	C1[0]
2	**100**	C1[1] C2[0]	C3[0]	C1[1]
3	10**0**	C3[0]	-	C3[0]
4	1**00**	C2[0] C3[0]	-	C2[0]
5	0**00**	C2[0] C3[0]	-	C2[0]
6	**10**1	C1[1]	C2[0]	C1[1]

<table>
<tr><th>Agent</th><th>C</th><th>Fitness</th><th>Buyer/supplier ratios</th><th>Performance</th></tr>
<tr><td>1</td><td>1</td><td>1 / (1 + exp(-(0)))
= 0.5</td><td>Upstream ratio = 1:
2 agents offer C2 [0]
2 agents need C2 [0]
Downstream ratio = 1:
1 agent offers C1[0]
Demand is F/N/2 = 6/3/2 = 1</td><td>0.5 * log2(1*1 + 1) = 0.5</td></tr>
<tr><td rowspan="2">2</td><td>1</td><td>1 / (1 + exp(-(0.3409)))
= 0.58</td><td>Upstream ratio = -
(agent makes its own input)
Downstream ratio 1/2:
2 agents offer C1[1]
Demand is F/N/2 = 6/3/2 = 1</td><td rowspan="2">C1: 0.58 * log2(1/2 + 1) =
0.34
C2: 0.5* log2(1*1 + 1) = 0.5
Average = 0.42</td></tr>
<tr><td>2</td><td>1 / (1 + exp(-(0)))
= 0.5</td><td>Upstream ratio = 1
1 agent offers C3 [0]
1 agent needs C3 [0]
Downstream ratio = -
(agent uses its own output)</td></tr>
<tr><td>3</td><td>3</td><td>1 / (1 + exp(-(0)))
= 0.5</td><td>Downstream ratio = 1
1 agent offers C3 [0]
1 agent needs C3 [0]</td><td>0.5 * log2(1*1 + 1) = 0.5</td></tr>
<tr><td rowspan="2">4</td><td>2</td><td>1 / (1 + exp(-(0)))
= 0.5</td><td>Upstream ratio: -
(agent makes its own input)
Downstream ratio = 1:
2 agents offer C2[0]
2 agents need C2[0]</td><td rowspan="2">C2: 0.5 * log2(1 + 1) = 0.5
C3: 0.5 * 1 = 0.5
Average = 0.5</td></tr>
<tr><td>3</td><td>1 / (1 + exp(-(0)))
= 0.5</td><td>Downstream ratio = -
(agent uses its own output)</td></tr>
<tr><td rowspan="2">5</td><td>2</td><td>1 / (1 + exp(-(0)))
= 0.5</td><td>Upstream ratio: -
(agent makes its own input)
Downstream ratio = 1:
2 agents offer C2[0]
2 agents need C2[0]</td><td rowspan="2">C2: 0.5 * log2(1 + 1) = 0.5
C3: 0.5 * 1 = 0.5
Average = 0.5</td></tr>
<tr><td>3</td><td>1 / (1 + exp(-(0)))
= 0.5</td><td>Downstream ratio = -
(agent uses its own output)</td></tr>
<tr><td>6</td><td>1</td><td>1 / (1 + exp(-
(0.3409)))
= 0.58</td><td>Upstream ratio = 1:
2 agents offer C2 [0]
2 agents need C2 [0]
Downstream ratio = 1/2:
2 agents offer C1[1]
Demand is F/N/2 = 6/3/2 = 1</td><td>0.58 * log2(1/2 + 1) = 0.34</td></tr>
</table>

Here, agent 2 finds an ecosystem redesign:

1. Current performance is 0.42. This is the baseline performance.

2. Agent 2 makes components 1 and 2, so it deals with 6 B matrix components. 2 are redrawn as follows. This is the redesigned architecture.

B matrix for agent 2 (changes in bold):

<table>
<tr><td rowspan="2">Component C1</td><td>C1 == 1</td><td>C2 == 1</td><td>C1 == 1 & C2 == 1</td></tr>
<tr><td>-0.0551</td><td>-0.7192</td><td>0.9365</td></tr>
<tr><td rowspan="2">Component C2</td><td>C2 == 1</td><td>C3 == 1</td><td>C2 == 1 & C3 == 1</td></tr>
<tr><td>-0.1654</td><td>0.4927</td><td>-0.3732</td></tr>
<tr><td rowspan="2">Component C3</td><td>C3 == 1</td><td></td><td></td></tr>
<tr><td>0.1174</td><td></td><td></td></tr>
</table>

3. The redesign performance is 0.39, a decrease in performance.
4. Agent 2 does not adopt the redesigned architecture yet.
5. Agent 2 performs *integration*:
    a. It considers integrating into component 3, with configuration 101.
    b. With the redesigned architecture, this leads to performance 0.48.
    c. Agent 2 adopts the new architecture and the integration change.

<table>
<tr><th>Agent</th><th>C</th><th>Fitness</th><th>Buyer/supplier ratios</th><th>Performance</th></tr>
<tr><td rowspan="3">2</td><td>1</td><td>1 / (1 + exp(-(-0.0551)))
= 0.49</td><td>Upstream ratio = -
(agent makes its own input)
Downstream ratio 1/2:
2 agents offer C1[1]
Demand is F/N/2 = 6/3/2 = 1</td><td rowspan="3">C1: 0.49 * log2(1/2 + 1) = 0.29
C2: 0.62
C3: 0.53
Average = 0.48</td></tr>
<tr><td>2</td><td>1 / (1 + exp(-(0.4927)))
= 0.62</td><td>Upstream ratio = -
(agent makes its own input)
Downstream ratio = -
(agent uses its own output)</td></tr>
<tr><td>3</td><td>1 / (1 + exp(-(0.1174)))
= 0.53</td><td>Downstream ratio = -
(agent uses its own output)</td></tr>
</table>

Note that, similarly to previous scenarios, this change affects agent 3 negatively as it loses its only downstream user.

APPENDIX B: CHARACTERISTICS OF THE MODELED LANDSCAPE AND INTERACTION STRUCTURE

An essential part of our modeling effort is to turn the NK landscape into one where we can manipulate the interaction structure. We do so by randomly drawing the coefficients of a vector of interaction effects jointly determining fitness, rather than drawing random fitness values directly. Fitness of a component (conditional on the states of itself and K other components) is determined by the following formula:

$$fc_i(c_i, \dots, c_{K+1}) = \frac{1}{1 + e^{-(B \cdot X)}}$$

Here, X captures the configuration of all bits that determine the performance of c_i (i.e., c_i plus K other bits). It is a vector of length $2^N - 1$, for each of the unique configurations possible with N bits (except the configuration of all 0s). An element of the X vector takes value 1 if the corresponding bit is in state 1, and 0 otherwise. Most elements of the X vector capture interaction effects; in these cases, they take value 1 if all bits specified in the interaction are in state 1, and 0 otherwise. Table B-1 provides an example of the X vector of a component whose fitness value is determined by itself and two other components, for various configurations. It has $2^3 - 1 = 7$ values.

--- Table B-1 here ---

The B vector in our logistic curve is the vector of random coefficients, specifying the positive or negative effects of having a bit (or combination of bits) set to 1 in a configuration. It consists of uniformly distributed random numbers in the interval U(-1,1), and has the same length as the X vector. $B \cdot X$, then, is the dot product of the two vectors. This means that a coefficient is only "counted" if the corresponding state of the relevant bits is 1. Another way to write $B \cdot X$ is as $b_1x_1 + b_2x_2 + b_{12}x_1x_2 + \dots$, where the b coefficients make up the B vector, and x_i designates the state of component i (0 or 1). If a configuration consists only of 0s, $B \cdot X = 0$ (as all elements in X are 0) and the resulting fitness value is $1 / (1 + e^0) = ½$. Thus, switching bits to 1 generates (random) fitness changes away from 0.5, within the range (0,1).

Our fitness specification is a generalization of the standard NK landscape, as every configuration (every random number in the standard NK) is uniquely specified by a set of coefficients. Note, however, that the configuration of all 0s is the baseline in our fitness function. Thus, if in a standard NK landscape the configuration of 0s has a fitness value other than 0.5, our function will have an offset variable, where $(B \cdot X)$ is replaced by $(O + B \cdot X)$, where O is found by solving $1 / (1 + e^{-O})$ = [the fitness of a configuration made of all 0s]. To keep our fitness landscapes symmetric around 0.5, we assume the offset variable O to be 0, with the consequence of having a configuration of 0s always return fitness value 0.5.

Ruggedness of Fitness Landscapes

While our fitness specification is a generalization of standard NK, standard NK landscapes are extreme cases of our landscapes. Our landscapes are generally less rugged, with the number of peaks increasing only moderately (though still exponentially) as K increases. See also Rahmandad (2019), who finds similar landscape features with a similar specification (PN landscapes), and provides a theoretical interpretation and empirical justification. Here, we are mainly concerned with being able to manipulate the interaction structure. Table B-2 shows the average number of local maxima (peaks) in our landscapes, as well as the height of the global maximum (in brackets).

--- Table B-2 here ---

Resilience to Changes in Interactions

One feature of standard NK fitness redrawings is that they essentially re-randomize the landscape. In this type of model, when architectural change occurs (e.g., an interdependency disappears), a firm's current fitness value is randomly redrawn and the shape of the landscape is randomly redrawn as well. As a result, such a specification is unsuitable for modeling search processes in a malleable architecture. One particularly attractive feature of our functional specification, at least for the purposes of this study, is that the post-redesign fitness more closely resembles prior fitness. In other words, a firm that is on a local peak is unlikely to find itself in a valley after the interdependency structure of the system changes, and vice versa.

We run a small simulation to confirm this result. To compare apples to apples, we generate 10,000 NK landscapes with N = 8, K = 4 with standard NK fitness landscapes (i.e., with random fitness values per configuration), and 10,000 with our logistic fitness function (i.e., with random coefficients driving fitness). Here we follow established NK models and evaluate fitness as the simple mean of all (8) component fitness values associated with each 8-bit configuration (and end up with $2^8 = 256$ fitness values on the total landscape). We then change the interdependence structure by removing one random interdependence of one of the components and evaluate the change in fitness of the same configuration. The results are shown in Figure B-1, where the x axis displays the change in fitness (negative or positive) and y displays the frequency of this change on a logarithmic scale. Two findings are noteworthy. First, in both models, there is a peak at $x = 0$, meaning that the most common outcome is that there is no change at all. In standard NK models, this can be explained as follows. The architectural change essentially removes half of the possible fitness values for a component (here from 2^4 to 2^3). If the configuration falls in the unchanged portion of the component landscape, fitness is unaffected for that component. For the functionally specified NK landscape, there is the additional factor of the baseline around 0.5 discussed above. Thus, if a component's configuration consists of all 0s, no architectural change will change the fitness value (it is 0.5 regardless).

The second finding is the slope of the frequency distribution as it moves away from 0. As predicted, in the standard NK model the slope is relatively flat. This means that post-change fitness is mostly unrelated to pre-change fitness. In functionally specified NK landscapes, the slope is much steeper—larger fitness change is very unlikely compared to standard NK specification. Thus, post-change fitness is generally much more similar to pre-change fitness.

--- Figure B-1 here ---

APPENDIX B TABLES AND FIGURE

Table B-1: illustration of the X vector in our logistic fitness specification

X vector element	Config. **000**	Config. **011**	Config. **101**	Config. **110**	Config. **111**
1 [ ] . .	0	0	1	1	1
2 . [ ] .	0	1	0	1	1
3 [ ] [ ] .	0	0	0	1	1
4 . . [ ]	0	1	1	0	1
5 [ ] . [ ]	0	0	1	0	1
6 . [ ] [ ]	0	1	0	0	1
7 [ ] [ ] [ ]	0	0	0	0	1

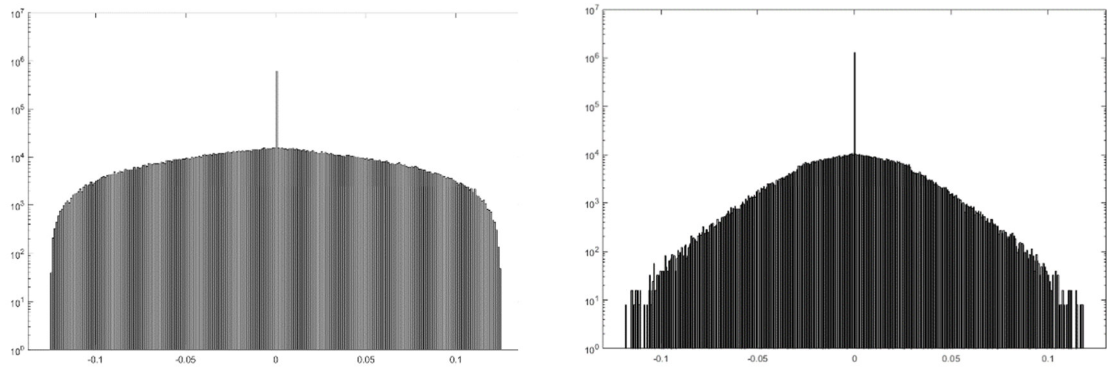
Note: in column 1, evaluated configuration "bits" for the vector element are indicated with square brackets [ ]

Table B-2: Characteristics of fitness landscapes with logistic specification

Number of peaks (height global max)	2	3	4	5	6	7	8	9	10
1	1,07 (0,59)	1,13 (0,59)	1,18 (0,59)	1,23 (0,59)	1,28 (0,59)	1,34 (0,59)	1,38 (0,59)	1,46 (0,59)	1,51 (0,59)
2		1,21 (0,60)	1,30 (0,61)	1,38 (0,60)	1,48 (0,60)	1,57 (0,60)	1,67 (0,61)	1,77 (0,61)	1,91 (0,60)
3			1,44 (0,62)	1,59 (0,62)	1,71 (0,62)	1,87 (0,62)	2,01 (0,62)	2,23 (0,62)	2,40 (0,62)
4				1,76 (0,64)	2,01 (0,64)	2,21 (0,64)	2,46 (0,64)	2,72 (0,64)	3,10 (0,64)
5					2,33 (0,66)	2,69 (0,66)	3,01 (0,66)	3,45 (0,66)	3,94 (0,66)
6						3,21 (0,68)	3,69 (0,68)	4,38 (0,68)	5,07 (0,68)
7							4,60 (0,70)	5,46 (0,70)	6,52 (0,69)
8								6,83 (0,71)	8,45 (0,71)
9									10,74 (0,73)

Note: columns indicate system size N, rows indicate complexity K

Figure B-1: Effects of architectural change on landscape fitness values



Notes: Y-axis has a logarithmic scale. The left panel shows the standard NK structure; the right panel our logistic function. N = 8, K = 4, 10,000 simulations (2,580,000 observations). Observed outcomes are differences in fitness values of the same configuration before and after removing one interaction from the NK interaction structure. For the logistic specification, removing an interaction means removing multiple coefficients from the B vector: for the main effect and all interaction effects including this component.

APPENDIX C: ADDITIONAL FINDINGS OF THE BASELINE INDUSTRY NK MODEL

The main text describes the evolution of firm scope and performance when varying the main parameters of the model. Figures C-1 and C-2 complement these findings by showing end-state vertical scope and performance for a more exhaustive set of parameter settings, plotted as averages and standard deviations. Note that the values in figures C-1 and C-2 correspond to the right-most data points of Figure 4 in the main text, organized along parameter settings. The standard deviations in these figures convey important additional information: our model essentially follows a population of firms that take different paths and end up in different positions. The standard deviations around average vertical scope and performance, then, show how different these positions are.

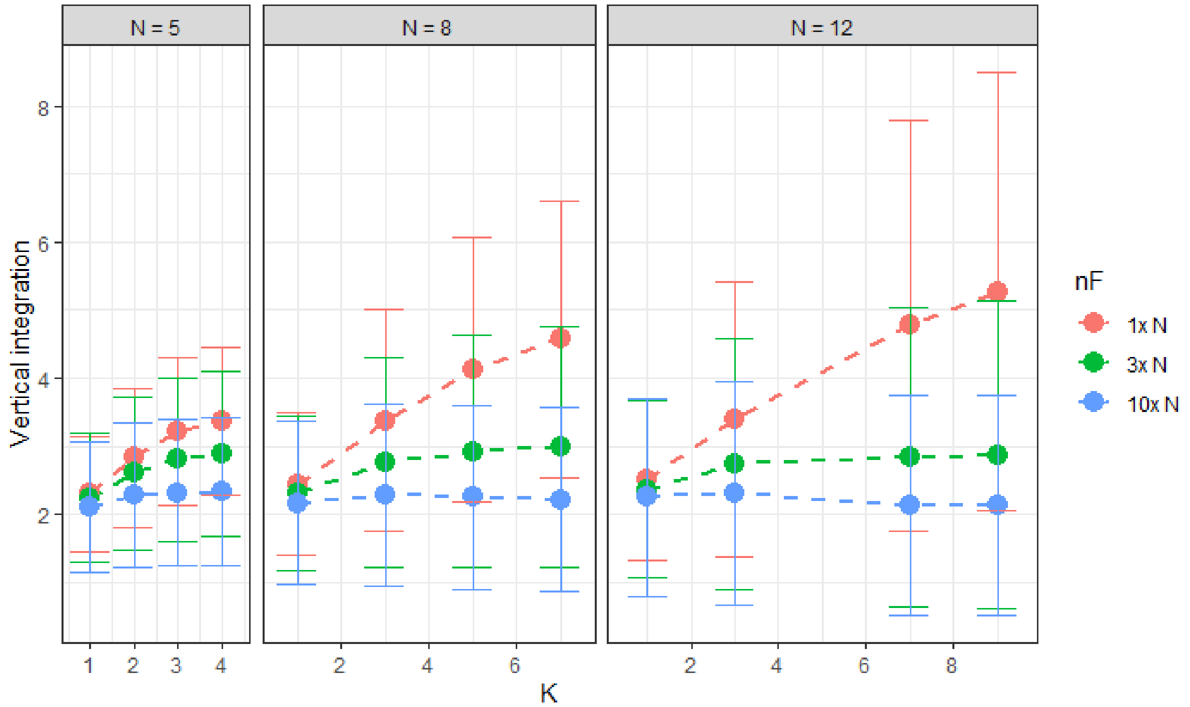
For vertical integration, these findings strengthen the picture emerging from the main results, highlighting the interaction between system size, complexity, and number of firms in driving average vertical scope. Scope increases with a decrease in firms, and this pattern is stronger with higher complexity. This means that, when there are many industry participants, firms find it easier to coordinate across vertical boundaries, whereas with fewer participants, firms find it more beneficial to grow in scope. For performance, the simulations show an important additional result in the distribution of performance over industry participants (shown by the bars indicating standard deviations). Industries with lower complexity and (especially) fewer firms are characterized by higher standard deviations, i.e., performance is unevenly distributed. This means that relatively many firms are trapped on a low local peak, while a few others are located in advantageous positions that enable them to achieve above-average performance (i.e., they occupy a "bottleneck").

Note that firms' performance is driven by the combination of "technical" fitness and interorganizational conditions, and their access to more attractive positions is likewise determined by both the shape of the technical fitness landscape and the choices and positioning of other firms in the industry. The uneven distribution of performance can be traced to these factors. In industries with low complexity K, the fitness landscape has fewer effective dimensions, and thus less space to explore for firms. The technical architecture drives the existence of bottlenecks. In industries with few firms F, firms have few options to choose from (e.g., it is likely that no potential suppliers exist for a certain upstream component, or potential buyers for a focal component). Here, the limitations of the organizational landscape drive the existence of bottlenecks.

--- Figures C-1 and C-2 here ---

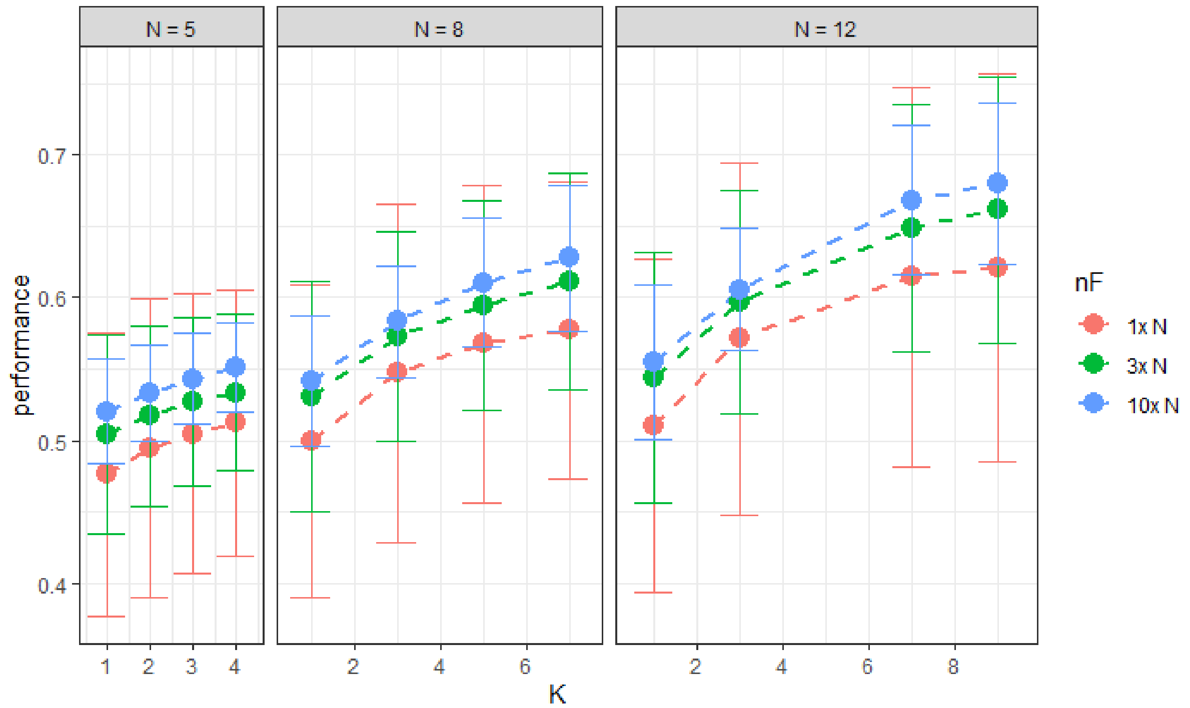
APPENDIX C FIGURES

Figure C-1: End-state vertical scope of baseline industry NK model



Note: Standard deviation is indicated by bars surrounding data points. Number of firms (nF) indicated as multiple of system component count N

Figure C-2: End-state performance of baseline industry NK model



Note: Standard deviation is indicated by bars surrounding data points. Number of firms (nF) indicated as multiple of system component count N