Article

Evolving user needs and late-mover advantage

Strategic Organization
2017, Vol. 15(1) 67–90

DOI: 10.1177/1476127016648498
journals.sagepub.com/home/soq
SAGE

Adrien Querbes
Carnegie Mellon University, USA

Koen Frenken
Utrecht University, The Netherlands

Abstract
We propose a generalized NK-model of late-mover advantage where late-mover firms leapfrog first-mover firms as user needs evolve over time. First movers face severe trade-offs between the provision of functionalities in which their products already excel and the additional functionalities requested by users later on. Late movers, by contrast, start searching when more functionalities are already known and typically come up with superior product designs. We also show that late-mover advantage is more probable for more complex technologies. Managerial implications follow.

Keywords
first-mover disadvantage, late-mover advantage, NK-model, product complexity

Introduction

Few firms that invent new technologies have been able to reap the benefits from their inventions. Often, firms whose inventions "colonize" a new niche market are not the ones that "consolidate" their inventions by transforming a niche into a mass product (Markides and Geroski, 2005). This suggests that there are first-mover disadvantages that allow late entrants to take over the industry leadership from the early entrants, as the product continues to evolve over time.

Traditionally, late-mover advantage is analyzed either from the perspective of appropriability conditions or complementary assets. Following Teece (1986), second movers are more likely to be successful if they can easily imitate the original invention or if they have complementary assets that they can leverage (such as marketing, manufacturing, or after-sales). In such cases, first movers have incurred the highest costs of research and development, while late movers are able to reap most of the returns.

Corresponding author:
Koen Frenken, Copernicus Institute of Sustainable Development, Utrecht University, PO Box 80115, Utrecht, NL-3508 TC, The Netherlands.
Email: k.frenken@uu.nl

An alternative (though not exclusive) explanation of late-mover advantage takes into account the evolution of the invention itself. Indeed, most products continue to evolve after their first introduction in the market. As most new products start out as niche products serving specialized users, the progressive diffusion of a new product generally involves the introduction of additional functionalities associated with user needs of a wider group of consumers (Christensen, 1997; Clark, 1985). A first-mover firm may well establish technological leadership in the niche market in question. However, this advantage can only carry over to the mass market when the firm is able to adapt its technology from a niche product serving specialized users to a dominant design attracting mass consumers (Lieberman, 2013). The question holds why in some contexts first movers have been successful in adapting their products to evolving user needs, while in other contexts they have not been able to do so, allowing late movers to take over industrial leadership.

We propose a model of late-mover advantage based on Altenberg's (1994) generalized version of Kauffman's (1993) "NK-model." Though the original NK-model has often been used in management science after it was first introduced by Levinthal (1997), the generalized version of the NK-model is preferable to the original model, in that it allows us to model the discovery of new functionalities within a given design space. In this way, we can compare first-mover and late-mover firms searching separately, and without imitation, in the exact same design space, but evaluating their searches against uneven number of functionalities. More specifically, we assert that the first mover starts searching for product designs once the first functionality is known, while a late mover only starts searching once all functionalities are known. Hence, the only type of learning taking place between first and late mover concerns the discovery of additional functionalities: after the introduction of a product by a first mover, additional user needs are being discovered, with these needs becoming apparent to both the first mover and the late mover.

By considering how first movers are being challenged by new functionalities emerging later in time, our model is focusing on the uncertainty created by the evolving user needs during new product development and on the technological constraints faced by first movers to deal with such needs. We choose not to model some of the other aspects that are known to affect first-mover advantage (e.g. imitation by late movers, learning curves and brand value). Hence, we do not claim to provide a comprehensive theory of late-mover advantage. Rather, our model is meant to develop a theoretical argument about how higher levels of product complexity provide more opportunities for successful entry by late movers as new user needs emerge over time. The empirical context is one in which a first mover successfully serves specialized users with specific user needs, while late movers target a mass market that demand additional functionalities. We will develop the case of BlackBerry as an empirical example, a company that pioneered the smartphone for specialized users but failed to conquer the mass market.

Using the generalized NK-model, we derive two propositions. *First*, we formally demonstrate that first movers find it hard to improve their product design in the face of the subsequent discovery of new product functionalities. By contrast, new functionalities open up a window of opportunity for late movers, who start designing "from scratch" and generally reach better product designs than first movers do. *Second*, we are able to show that first-mover disadvantage is contingent upon a product's complexity. Complex product technologies—in terms of interdependencies between their underlying component technologies—present more severe trade-offs between existing and newly discovered functionalities compared to more modular technologies. As a result, first movers will find it harder to improve, the more complex the product technology.

Late-mover advantage

Technological leadership is a key determinant of first-mover advantage in new industries (Lieberman and Montgomery, 1988, 1998). The first firms that enter a new market are able to

develop cost leadership by being the first to go down the learning curve. Though some of the economies due to learning-by-doing may spillover to later entrants, such knowledge spillovers are expected to be limited due to the tacit knowledge residing in skills and organizational routines (Nelson and Winter, 1982). First movers may also be able to protect their product technology—for example, through patenting some of its component technologies—raising even higher the barriers to entry for late movers. Patents are particularly effective in securing industry leadership in complex product industries, as technological substitutes are more difficult to integrate in complex products than in more modular products (Marengo et al., 2012). Finally, first movers also may benefit from economies of scale in doing R&D. As the firm size of first movers tends to exceed the size of late movers at the time the latter enters the market, first movers can spread the sunken investments in R&D over a larger number of items than late movers (Klepper, 1996).

Technology-based explanations of the competitive advantage of first movers are built on the implicit assumption that the structure of demand does not change fundamentally in the course of time. Indeed, without any fundamental change in demand, buyers' inertia only reinforces the advantages for first movers (Lieberman and Montgomery, 1988, 1998). However, during the early phase of product evolution in new markets, user needs are generally still in flux and coevolve with innovative activity (Clark, 1985; Von Hippel, 1988). Sociologists coined this process *domestication*, which refers to the way users incorporate a new technology into their practices (Lie and Sørensen, 1996; Silverstone and Hirsch, 1992). During domestication processes, products do not necessarily fully keep their intended functions as new ways of using are discovered over time, quite differently from designers' intents.

Examples of functionalities that generally become articulated much later than the initial introduction of a new product category include the following: (1) safety issues that only become apparent after extensive experimentation in usage, (2) ergonomic features that only become known only after early users have experienced physical complaints, and (3) interoperability issues regarding the joint use of the new product with existing devices. New functional requirements may also stem from government regulations that, by nature, considerably lag behind the creation of a new product category.

Due to uncertainties regarding user needs in the early stages of a new industry, firms cannot fully foresee *ex ante* how their products will be received by users. Once first movers have created niche applications for a new technology, demand becomes more articulated and the knowledge of user needs builds up. Only then, mass market applications can be envisaged, and resources to develop such mass products will be deemed legitimate. Oftentimes in this process, a dominant design emerges serving the needs of the mass market at a relatively low price (Abernathy and Utterback, 1978). Hence, late movers that are quick to adopt the dominant design may benefit from product standardization without having to incur the costs of experimentation and technology switching, which first movers face (Dowell and Swaminathan, 2006; Lieberman, 2013; Suarez et al., 2015). Often, such successful late entrants are spinoffs of earlier entrants that build on the parent's experience, while experimenting with new designs that otherwise would cannibalize the niche product of the parent firm (Klepper and Sleeper, 2005). In economic terms, late movers benefit from positive externalities generated by first movers as they "free ride" on the innovation efforts by pioneers—that is, apart from knowledge spillovers regarding technological knowledge. Knowledge spillovers also occur because user needs get better understood, reducing market uncertainty.

Our model thus resonates with the plea by Suarez and Lanzolla (2007) to incorporate "environmental dynamics" in first-mover advantage theory. They call for theoretical frameworks where environmental dynamics may render late entry more advantageous than early entry. They argue that "*technology evolution may render a firm's knowledge obsolete, destroy existing competences*" (Suarez and Lanzolla, 2007: 382–383). As an example, they mention "*product categories with high 'vintage effects'—that is, where product quality significantly improves over*

time." This environmental dynamics is indeed underlying our model, where the discovery of new functionalities as an environmental dynamic leads to new product designs with higher fitness with user needs (i.e. "product quality"). These new designs are generally not developed as small variations of the pre-existing design, since the new functionality cannot be integrated easily in a design that was optimized for a different set of functionalities. Rather, taking into account a more complete list of functionalities, late movers can start designing "from scratch," which leads them to come up with radically new design propositions. To be successful as a new entrant, a late mover does not necessarily need to match a first mover in terms of specific technological competencies that allowed the first mover to excel in some specific product functionality for specific users, but rather to come up with a product design that is optimized against all functionalities that have become known over time. As a result, first movers do not only suffer from an outdated design proposition, but they also have to devaluate the knowledge and competences built up in the past.

From a managerial perspective, the challenge holds to learn about evolving user needs. First movers will initially sell their pioneering product to specialized users. A potential disadvantage for such firms lies in the difficulty to gauge the needs of a potential mass market and to reorient their R&D investments accordingly. A parallel reasoning exists in the work on incumbents versus disrupting newcomers, where the latter can still be considered a late mover (sometimes even decades after the product was initially launched). Christensen (1997, 2003) discusses several industries in which incumbents tended to innovate myopically focusing on existing users, rather than focusing on developing new products that appeal to a wider group of users.[1] This strategy is reinforced by the sunk costs invested in specific technologies and branding, as well as by switching costs associated with developing and producing alternative products. As Christensen and Rosenbloom (1995) put it,

> [i]t is difficult for established firms to marshal resources behind innovations that do not address the needs of known, present and powerful customers. In these instances […] the essence of the attacker's advantage is in its differential ability to identify and make strategic commitments to attack and develop emerging market applications, or value networks. The issue, at its core, may be the relative abilities of successful incumbent firms vs. entrant firms to change strategies, not technologies. (pp. 255–256)

In this context, incumbents are first movers who become so successful in serving specialized users that they continue to focus on this user group, at the risk of losing out the emerging mass consumers who are characterized by different needs.

The ability of firms to reorient their R&D toward new users also depends on the characteristics of a product's technology. Innovations focusing on new users will be especially hard to develop when many interdependencies exist between a product's components, that is, when the product's technology is complex (Ethiraj et al., 2012). Innovations appealing to new users may cause malfunctions in existing features, possibly offsetting existing users. Consequently, firms may be more reluctant to orient product innovations toward new users, the more complex the technology at hand.

A recent, telling example is the rise and fall of BlackBerry, the company named after its BlackBerry smartphone.[2] Until 2007, it was the market leader with competitors such as Palm and Nokia copying the BlackBerry's layout (West and Mace, 2010). With falling prices, the smartphone became a standard consumer product attracting industry giants like Apple and Google into the market. With the advent of the Apple iPhone in 2007 and Android-based smartphones little after, BlackBerry suddenly lost its leading position with its market share shrinking to less than 2% in 2012 (www.gartner.com).

Compared to other smartphones BlackBerry excelled in two functionalities: security and instant email delivery. These functionalities were particularly valued by its main users (large companies and the US government). However, with the advent of the iPhone, the use of mobile phones was redefined as smartphones with many more functionalities, email being just one of them. Furthermore, mass consumers cared less about security than the original lead users. BlackBerry, however, was slow to realize that it had to redefine its own product, as ordinary users valued security and instant email much less than their specialized users did. On the one hand, the company suffered from agreements between Microsoft, Apple, and Android on its core professional services as the spread of Microsoft Exchange on smartphones benefiting from a large installed base of users (Kenney and Pon, 2011). On the other hand, the network externalities that should have favored BlackBerry were effectively counterbalanced by the ability of new entrants to generate indirect network effects through complementary services, in particular, in the form of the digital contents available on the Internet including Apple's iTunes and Google's information services (West and Mace, 2010). This convergence between smartphones and the Internet content supported the evolution of smartphones' functionalities toward a multifunctional IT device, to the detriment of BlackBerry.

The example underlines two interrelated dynamics. First, technological evolution involves the addition of new functionalities to existing products, hereby transforming a specialized niche product into a mass product. Second, the addition of new functionalities opens up windows of opportunity for new entrants. Hence, technological evolution has a direct impact on industrial dynamics. Though examples of the transformation of niches into mass markets are well documented,[3] its implications for industrial dynamics have not been subject of explicit theorizing hitherto, which we aim to provide.

The main contribution of the model that follows is to show, in a formal sense, that the subsequent addition of functionalities indeed opens up opportunities for new entrants. While first movers find it hard to provide new functionalities by adjusting their current designs, new firms start searching "from scratch" and without the burden of previous solutions selected against an incomplete list of functional specifications. From the formal model, we are also able to show that adaptation by first movers is more difficult, when more component technologies are interdependent. Hence, *ceteris paribus*, we can expect that the more complex a technology, the more volatile the industrial dynamics.

The model

In order to probe the logic of the design of complex products and their effect on industry leadership, we introduce a model of complex systems based on the NK-model developed by Kauffman (1993) and generalized by Altenberg (1994). The NK-model has its roots in biology, where it is used to study the interaction between genes and traits in biological organisms. After Levinthal (1997) introduced the NK-model in management science, this model has been widely used to theorize about learning curves (Auerswald et al., 2000), modularity (Ethiraj and Levinthal, 2004; Marengo et al., 2000; Simon, 2002), imitation (Ethiraj et al., 2008; Rivkin, 2000), decentralized decision-making (Rivkin and Siggelkow, 2003; Siggelkow and Levinthal, 2003), technological evolution (Frenken and Nuvolari, 2004; Marengo et al., 2012), search strategies (Baumann and Siggelkow, 2013; Knudsen and Levinthal, 2007), industry shakeouts (Lenox et al., 2007), and entrepreneurship (Ganco, 2013).[4]

Here, our objective is to assess how evolving user needs have different effects on first and late movers. Our interest in new functionalities is the reason why we rely on the generalized NK-model by Altenberg (1994), which allows us to keep constant the number of components in a product,

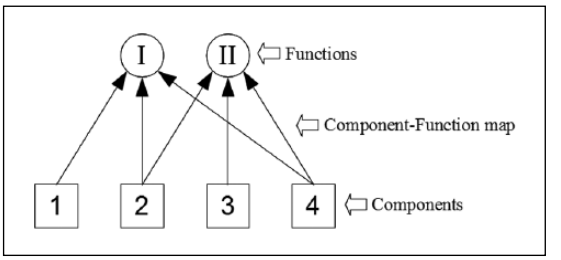

Figure 1. An example of component-function map with $N = 4$, $F = 2$, and $K = 3$.

with the number of functionalities increasing over time. We can then define first and late movers by the time they start searching the fitness landscape. First movers do so once the first functionality is known, while late movers only enter later once all functionalities are known. Hence, what differentiates first and late movers is the number of functionalities taken into account when they start their search process.

The generalized NK-model

In the generalized NK-model by Altenberg (1994),[5] a product is depicted by N component technologies interacting together to produce F functions, according to a *component-function map*. This map is more or less complex depending on the level of K, which sets up to the *polygeny* of functions, namely, the number of component technologies producing one function. Figure 1 shows an example of such a mapping with $N = 4$, $F = 2$, and $K = 3$. The generalized NK-model includes the original NK-model by Kauffman (1993) as a special case for which $F = N$ and, hence $K \in \{1;\ldots;N\}$.

Changes in components lead to changes in the performance (*fitness*) of functions according to the mapping between components and functions. For example, in Figure 1, a change in component 1 will only lead to a change in the performance of Function 1, but a change in component 2 would lead to changes in the performances of both Function 1 and Function 2. The number of functions influenced by a single component is called a component's *pleiotropy*. In the example, the first and third components have a pleiotropy of one, while the second and fourth components have a pleiotropy of two. It follows from the mapping of components onto functions that the sum of pleiotropy values of components must equal the sum of polygeny values of functions.

We make the assumption, without loss of generality, that each component has two possible states, allowing for 2^N possible combinations of components. The space of possible combinations is called a technology's "design space." Each binary string in the design space corresponds to a product technology x, with $x = \{x_1,\ldots,x_j,\ldots,x_N\} \in \{0,1\}^N$ as the binary string representing a particular combination of component technologies, which is part of a design space of size 2^N. The set of technologies and their corresponding fitness values constitutes a "fitness landscape" (Kauffman, 1993; Wright, 1932). Following Altenberg (1994), the fitness of a product technology x is given by

$$\varphi(x) = \frac{1}{F}\sum_{i=1}^{F}\varphi_i\left(x_{j_1(i)}, x_{j_2(i)},\ldots,x_{j_K}\right) \tag{1}$$

where $\varphi_i : \{0,1\}^K \to [0,1]$, based on uniform pseudo-random draws, $\{x_{j_1(i)}, x_{j_2(i)},\ldots,x_{j_K}\} \subset \{x_1,\ldots,x_j,\ldots,x_N\}$ as the set of K component technologies affecting the function i. Thus, the

fitness of one function (φ_i) depends on the state of the K components affecting this function while the global fitness (φ) is the average of the fitness of every function.

Search

We model the search efforts by firms as a *local* and *greedy* search algorithm. *Local* means that firms mutate only one component at the time. *Greedy* means that among all N potential one-component mutations, the firm chooses the mutation which offers the highest fitness improvement.[6] Search will stop once a local peak is found. Such product technologies correspond to local optima in the fitness landscape, which are products that can no longer be improved by mutations in single components.

As we distinguish between first and late movers, firms differ in the time they enter the market, which in turn determines how they evaluate the fitness of mutations. The late mover looks at the system as a whole once all functions are known. Hence, it looks for the mutation that yields the highest fitness increase of the global fitness *irrespective* of the fitness level of individual functions. This search procedure is equivalent to standard (greedy) hill-climbing in NK-model (Kauffman, 1993; Levinthal, 1997). The first mover, by contrast, discovers each function gradually as time goes by. First movers, in first instance, will evaluate mutations only against the fitness of the first function (φ_1). Once it has reached a design that is locally optimal with regard to the fitness of the first function, the second function becomes known, and the first mover will continue searching by evaluating mutations taking into account both functions, and so on. Thus, we assume here that the time in between the discovery of two functions is long enough to allow the firm to carry out all the mutations required to find a new local peak. This procedure is repeated until all functions are known. We can thus express time by the sequence of discoveries of new functions. That is, the first function is discovered at time $t = 1$, the second function at time $t = 2$, and so on. It follows that the counter ends at $t = F$. It also follows that φ_t is the fitness of the function t discovered at time t.

Since first movers discover functions sequentially, they have to decide how to deal with functions that they already optimized in the past. There are two possible strategies. Once they start optimizing a newly discovered function, they may decide not to compromise the fitness achieved through previously discovered functions. We call this assumption the "functional inertia" assumption. Alternatively, first movers may allow the fitness of previously discovered function to decrease as long as the mean fitness over all t functions considered at time t, increases. We refer to the latter strategy as the "functional flexibility" assumption.

Strategy 1: functional inertia. The assumption of functional inertia in the search behavior of first movers implies that, each time t^* a new function is discovered, first movers will only mutate the components that influence the most recently discovered function (φ_{t^*}) and that do not influence functions discovered earlier (φ_t for $t < t^*$). This assumption means that a first-mover firm—excelling in providing particular functions in the past—would not be willing to give up this leading position with respect to these functions. Such instances of inertia by firms may reflect the case where a firm's brand is strongly associated with excellent performances in providing particular functionalities, or equally, the case where a firm remains focused on its installed base of users (Christensen and Rosenbloom, 1995). The advantage of functional inertia is that the relevant design space for mutations becomes progressively smaller as more functions are being discovered. A smaller design space, in turn, makes search more efficient as the chance of finding a fitness increasing mutation decreases, where efficiency is defined as the number of mutations required to reach a certain fitness level. At the same time, this strategy also implies that the components influencing early discovered functions will get fixed into a particular state ("locked-in") early on, as

each component is only optimized with reference to the first function that it influences.[7] Consequently, the early fixation of components in particular states constrains a firm's design space, possibly excluding some paths to more promising optima. Hence, the advantage of higher search efficiency comes at the cost of a lower expected fitness value of the final optimum that is found at time $t = F$.

Strategy 2: functional flexibility. The previous strategy, which holds that the first movers will not accept any decrease in existing functions when confronted with a newly discovered function, can be criticized as being too restrictive. From a behavioral perspective, this assumption would mean that a first-mover firm—excelling in providing particular functions in the past—would under no circumstances be willing to give up its leading position with respect to these functions. Though such instances have been documented (Christensen, 1997, 2003), one can expect that many firms are willing to give up fitness in a particular functionality in situations where the overall quality of the product will increase. This "functional flexibility" strategy thus aims at optimizing all functions insofar functions are known to the firm. That is, at any time t, the first-mover firm will evaluate mutations with regard to the mean fitness of the t functions known at this time. Hence, in contrast to firms following the functional inertia strategy, all components are candidates for mutation at all times. This means that, as a search strategy, the functional flexibility strategy will be less efficient, but given its wider search scope, it is expected to find product designs with higher fitness compared to the functional inertia strategy. Consequently, this strategy applies best to industries where R&D is relatively inexpensive in comparison with the benefits of raising product quality.

Simulation

We perform two sets of simulations: one set of simulations compares a first mover with a late mover assuming that the first mover follows the "functional inertia" strategy, and a second set of simulations compares a first mover with a late mover assuming that the first mover follows the "functional flexibility" strategy. For both the first-mover search strategies and the late-mover search strategy, we performed 5000 simulations for each combination of settings of the parameter values *N, F*, and *K*. Every simulation starts by creating the landscape given the combination of parameter values *N, F*, and *K*. These values are $N = 2^n \left(n \in \{1,\ldots,5\}\right)$, $F = 2^f \left(f \in \{0,\ldots,5\}\right)$, and $K = 2^k \left(k \in \{0,\ldots,5\}\right)$, provided that the values of N, F, K satisfy $F \times K \geqslant N$, meaning that each component influences at least one function (otherwise, its inclusion in the representation of a product technology would be redundant). The simulation starts by randomly selecting a string from which the search will start.[8] The simulation ends when there are no more possible fitness improvements. At that time, the firm has reached a peak on the fitness landscape and any mutation would produce a fitness decrease.

Once the simulation is over, we save the *fitness value* φ of the string that is found at the end of the simulation, and the *number of mutations* preceding that discovery. Fitness is our main variable of interest since it reflects the quality of the product that a firm is able to offer and, hence, directly influences its competitive advantage. The number of mutations is also of interest, since it reflects the amount of resources spent on search: the higher the number of mutations required, the less efficient is a search strategy. To compare the number of mutations for products of different size N, we show a normalized number of mutations, namely $\textit{number of mutations}/(N/2)$ given that $N/2$ is the average Hamming distance between two randomly chosen strings in the fitness landscape. Hence, the expected distance between the randomly chosen string at which search starts and any optimum in the landscape at which search will end also equals $N/2$.

We also record the exact string x in each simulation, as to assess to what extent the product designs of first movers differ from those of late movers. We are interested in *product similarity* in the design space because the gains from licensing proprietary technologies held by the first mover to the late mover will be greater, the more similar the products of first and late movers are in terms of the choice of components, following the idea that patents apply to component technologies (Marengo et al., 2012). We refrain from modeling patenting explicitly but rather use product similarity as a first proxy of first-mover advantage over late-mover advantage in product contexts where patenting is relevant. That is, the more similar are the product designs of the first mover and late mover, the more likely licensing fees would be claimed by the first mover.

Product similarity between the two strings is given by counting how many components in the two designs are in the same state (e.g. 1110 and 1010 have a similarity score of three). To compare the similarity scores across simulations for products of different size N, we express similarity as a percentage of the maximum possible similarity score, which is N in the case that two strings are identical (following the example above, the similarity score between 1110 and 1010 is 75%). The higher this percentage, the more similar the product design of first and late movers, the more likely first movers can benefit from proprietary technology. On the lower side, the similarity will not decrease below $N/2$, which is the average similarity between two random products defined by two binary strings of equal length N. Hence, to make results comparable across simulations with different values for N, we scale the similarity value by dividing it by its minimum value $N/2$.

As an example, consider again the case given in Figure 1 of a product technology with $N = 4$ components, producing $F = 2$ functions, each of them is influenced by $K = 3$ of components. The global fitness of each design x is given by

$$\varphi(x) = \frac{1}{2}\left(\varphi_I\left(x_1, x_2, x_4\right) + \varphi_{II}\left(x_2, x_3, x_4\right)\right) \tag{2}$$

In Appendix 1, we present a simulation of fitness values of all possible designs. Using these values, we can illustrate the search strategies of late movers, first movers following the functional inertia strategy, and first movers following the functional flexibility strategy. The three strategies can be described as follows:

- *A late mover* starts searching at time $t = F$, that is, once all functions are known. The search is greedy-local and it continues as long as a mutation can improve the global fitness. Search stops when the firm reaches a local optimum.
- *A first mover following the functional inertia strategy* starts searching at time $t = 1$ when the first function becomes known. It optimizes φ_1 by greedy-local search, looking for fitness increasing mutations in the subset of K component technologies that influence φ_1. In the example of Figure 1, this subset consists of components 1, 2, and 4. It stops searching when a local peak is reached with regard to φ_1. Then, as time moves from $t = 1$ to $t = 2$, the first-mover firm attempts to improve φ_2 by mutating components that influence φ_2 but that do not influence φ_1. Following the example of Figure 1, only a mutation in component 3 is allowed, since mutations in any other components would always negatively influence φ_1. Search ends when a peak is found, as being locally optimal with regard to φ_1 and φ_2, given its inertia to leave component affecting φ_1 untouched. Hence, this peak is not necessarily locally optimal with respect to the global fitness φ.
- *A first mover following the functional flexibility strategy* first optimizes φ_1 by mutating components 1, 2, or 4 by greedy-local search, since these are the only components influencing φ_1.

When a peak is reached for this function, the second function is discovered, and the firm now attempts to improve the mean fitness of φ_1 and φ_2, which equals φ. Search now involves mutations in all four components, since mutations in all components are potentially increasing the global fitness φ.

Results

We present the simulation results in Figure 2, comparing late movers with first movers following the functional inertia strategy, and in Figure 3, comparing the late movers with first movers following the functional flexibility strategy. In Appendices 2 and 3, we report on the Wilcoxon signed-rank test for the results reported in Figure 2 and 3, respectively. These tests indicate for which parameter settings the values of first and late movers are significantly different. For completeness, we also study in Appendix 4 all the intermediate sequences of functionality discovery for the subset of parameters $N = F = 16$ and $K = \{2^0, 2^1, 2^2, 2^3, 2^4\}$. That is to say, apart from only looking at the two extreme cases of a first mover who discovers functionalities one by one ($\{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1\}$) and the late mover who discovers all 16 functions jointly ($\{16\}$), we also look at all intermediate cases of discovery as a robustness check (for instance $\{4,2,8,2\}$, $\{15,1\}$, and so on). As further explained in Appendix 4, the results that follow for the two extreme cases of functionality discovery remain robust for any alternative specification of discovery. For example, we looked at the results assuming that a first mover would discover not just one but several new functions at each time step. Furthermore, we analyzed how results differ if the first mover starts off the design process knowing not just about one but about several functions already. In all these cases, the advantages and disadvantages of being a late mover still hold true as discussed below.

Case 1: functional inertia

Figure 2 shows the simulation results comparing first movers with late movers following a functional inertia strategy for every combination of parameters $N, F,$ and K. It reports the average fitness values of the final local optimum discovered, the number of preceding mutations, as well as the similarity between the locally optimal strings found by the first mover compared to the late mover. White dots refer to first movers and gray dots to late movers.

From Figure 2, two general observations can be made that apply both to first movers and late movers. First, as long as $F = 2^0 = 1$, the first and late-mover strategies lead to equivalent results. This follows from the definition of first and late movers: first movers start searching when the first function is known while late movers start searching when all functions are known. If the total number of functions is just one, the two strategies must be equivalent. However, since search starts from a random string, first and late movers may still end up in different local optima. Yet, averaged over 5000 runs, the fitness of these optima are necessarily equivalent. This means that for products with a low number of functions F and low complexity K, first and second movers are equally likely to win out; that is, firm performance is mostly a matter of luck. It also means that for products with higher values of F and K, which arguably holds for most products empirically, the differences in performance are significant.

A second observation holds that, for given values of F and K, increasing N leads to higher fitness. This can be understood from the fact that the more N exceeds F and K, the lower the pleiotropy of components (which is the number of functions influenced by each component), the fewer trade-offs exist, and more scope exists for firms to improve each function without creating malfunctions in the other components.

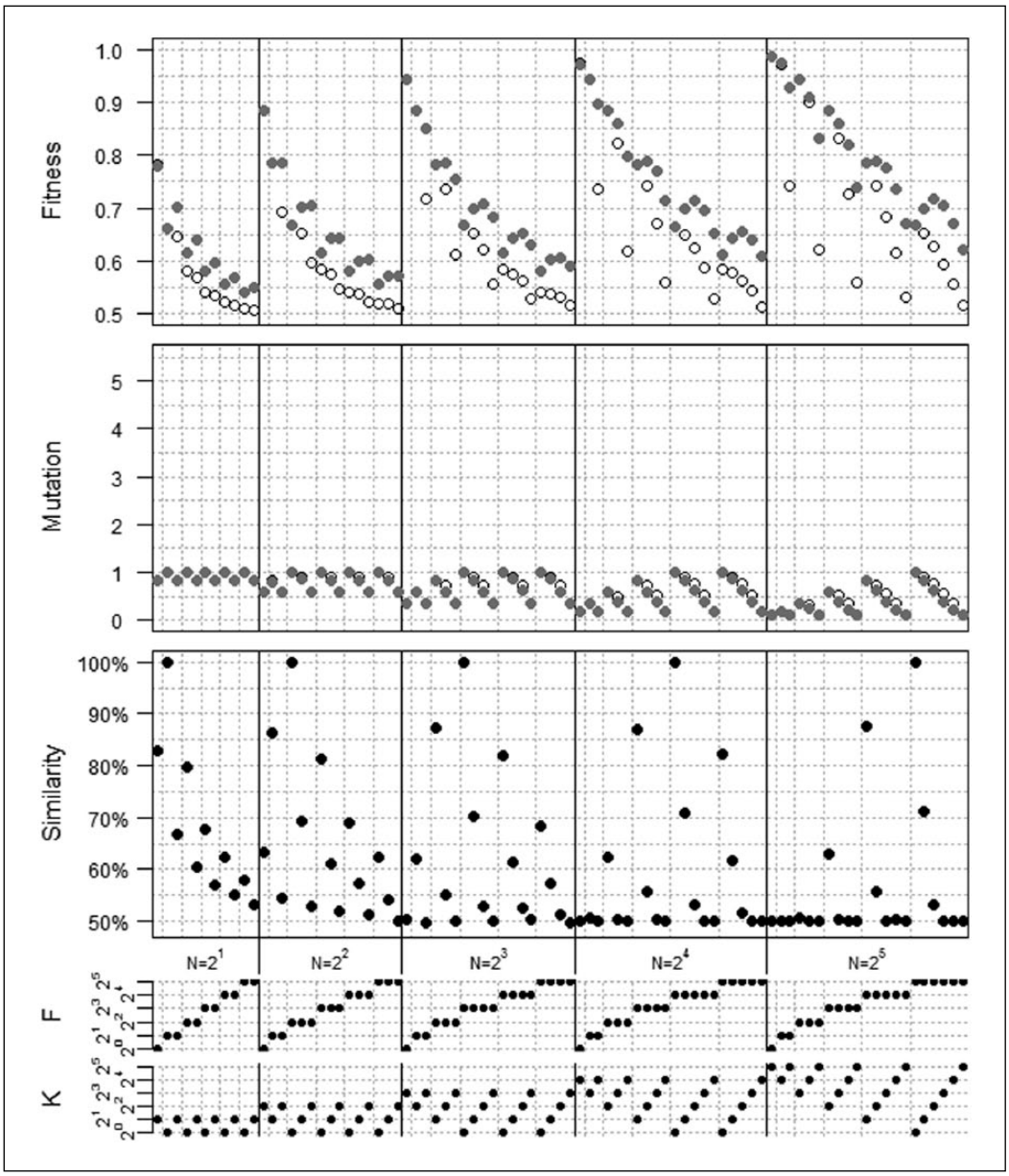

Figure 2. Average fitness, average normalized number of mutations, and product similarity (for the case of functional inertia). White dots (○) refer to first movers following a "functional inertia" strategy and gray dots (•) to late movers.

Turning now to the comparison between first and late movers, the first result holds that late movers achieve higher fitness than first movers for most parameter settings, and this fitness difference—given a set of values of F and N—increases with the product's complexity K. Hence, product complexity is indeed a major obstacle for first movers to adapt their design as more product functionalities become known over time (Ethiraj et al., 2012).

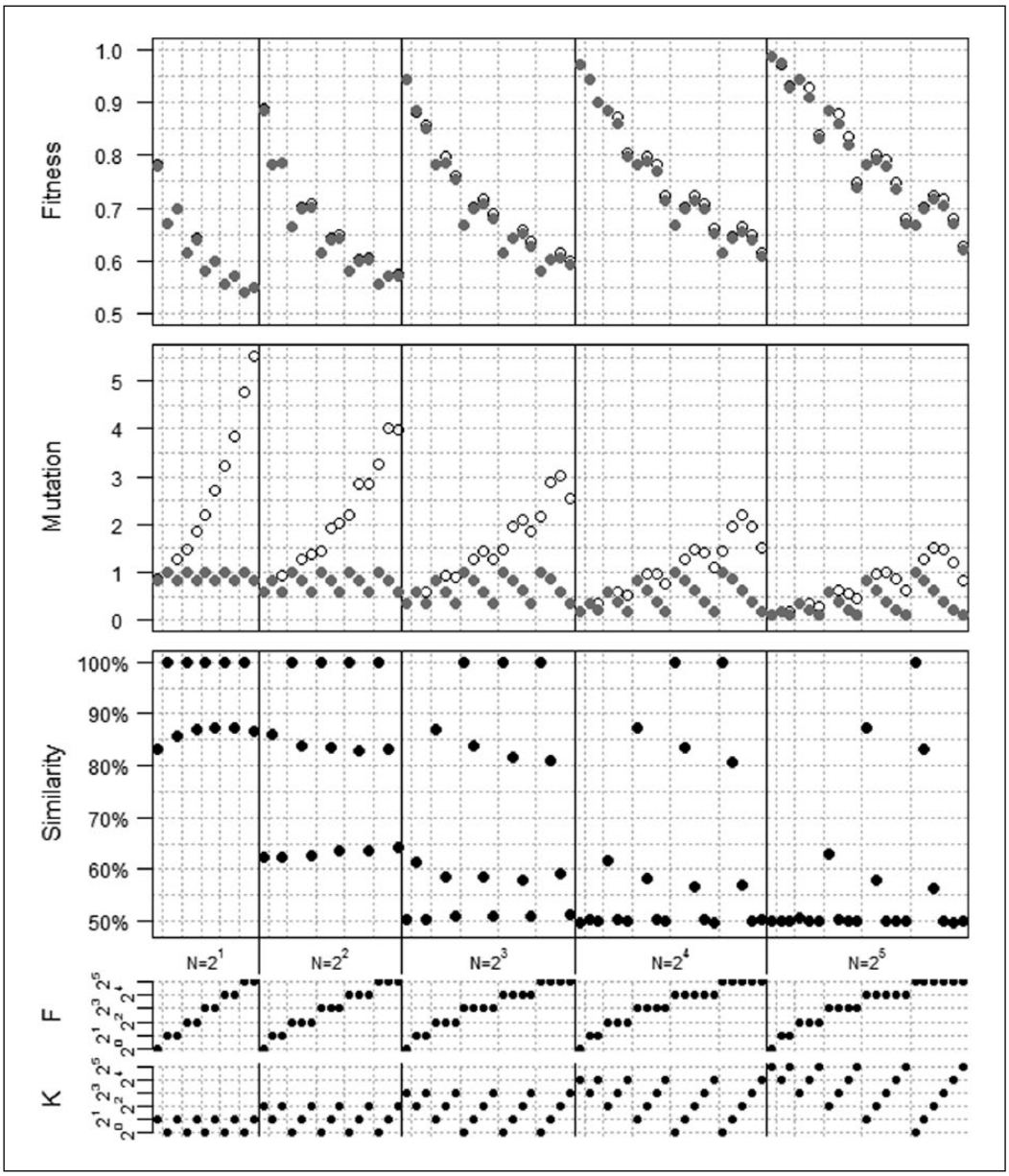

Figure 3. Average fitness, average normalized number of mutations, and product similarity (for the case of functional flexibility). White dots (○) refer to first movers following a "functional flexibility" strategy and gray dots (●) to late movers.

Looking at the total number of mutations preceding the discovery of the final local optimum, as an indicator of search inefficiency, we also observe a small but significant advantage of late movers over first movers for most parameter settings. Late movers put less search efforts than first movers, since late movers only engage in search during the final period once all functions are known, while first movers continuously extend their search efforts each time a new function is discovered.

Finally, we can observe that the similarity in the final product design found by first and late movers goes down with increasing values for K. This is in line with earlier findings that the number of local optima increases with K, given the values for F and N. That is, the "ruggedness" of a fitness landscape increases with the complexity of the system in question (Kauffman, 1993). In competitive terms, this result indicates that the advantage that first movers may enjoy by appropriating component designs before late movers enter the market becomes smaller for more complex products.

Case 2: functional flexibility

We repeated the simulation analysis for first movers following the functional flexibility strategy instead of the functional inertia strategy. Now, unlike before, first movers consider all components as candidates for mutation at any time. Figure 3 shows again the results for the average fitness values of the local optimum discovered, the normalized number of preceding mutations, and the similarity between the strings found by first and late movers. White dots refer to first movers and gray dots to late movers.

Changing the assumption about how first movers search, leads to different results than those obtained before. First, looking at the fitness achieved by first and late movers, the clear advantage of late movers as reported in Figure 2, now turns into a small, and mostly significant, advantage for first movers in Figure 3. This result can be understood by looking at the functional flexibility strategy of first movers in closer detail. Recall that in the case of functional flexibility, both first movers and late movers are allowed to mutate all components at all times. The only difference between the two search strategies is the actual trajectory they follow through the design space. Late movers only start searching after all F functions are known and immediately move toward a local optimum following their greedy search algorithm. By contrast, first movers look for a new local optimum each time a new function is discovered. Hence, the final local optimum is not found by starting from a randomly chosen string, as for late movers, but from a string that was locally optimal with respect to fewer functions, which was in turn discovered from a string that was locally optimal with respect to even fewer functions and so on.[9]

The fitness advantage of first movers, however, comes at a cost of search inefficiency. Looking at the number of mutations preceding the discovery of the final local optimum, we clearly see that first movers spend much more effort than late movers. This difference goes up when K increases, with given values for F and N. It also happens when F increases, with given values of N and K. For large systems, the difference can be fivefold, with late movers generally requiring only two or three mutations, while first movers often require 10 mutations or more. It is exactly the functional flexibility that renders the first-mover strategy so costly compared to first movers whose search behavior follows the functional inertia. Hence, to the extent that mutations are costly, the small fitness advantage that first movers enjoy over late movers is likely to be offset by the higher cost of R&D.

Turning to the results on the similarity in product designs found by first and late movers, we observe again that the similarity score goes down with increasing product complexity K. Hence, even if first movers are functionally flexible, late movers still may well end up in different local optima in designing complex products. Again, we can conclude that, the higher the complexity of a product, first movers profit less from appropriating component designs before late movers enter the market.

Discussion

We have shown that the late movers may enjoy advantages over the first movers in product technologies where new functionalities are discovered over time. This is particularly the case for

complex products where technological interdependencies create difficulties for first movers to adapt their product design in order to include more functionalities. The more complex a product technology is in terms of interdependencies between its component technologies, the greater the challenge for first movers to integrate new functionalities given the technology choices made early on, and the less likely they will be able to sustain their first-mover advantage. Complex products also lead late movers to explore more dissimilar designs than first movers, thus largely circumventing proprietary component technologies that first movers may hold.

In conclusion, our proposition holds that the more new functions become known as user needs evolve, the more likely a late mover will take over the industry leadership of the first mover. Our model provides one explanation why first movers are often unable to seize the opportunities of adding new functionalities and hereby expanding their user segments. It also provides a clear and testable hypothesis for the mixed empirical results on first-mover advantage (Lieberman, 2013) by using product complexity as a moderating variable: the higher a product's complexity, the less likely first-mover advantages can be sustained in the face of evolving user needs, *ceteris paribus*. This role of technological interdependencies in industry dynamics is in line with an evolutionary model of industry shakeouts, which showed that in industries with more complex production processes, the rates of entry and exit remain high over longer time periods, with decreasing survival rates for incumbents (Lenox et al., 2007).

Our model complements the work by Adner and Levinthal (2001) on demand heterogeneity and technological speciation. In our model, user needs evolve because new product features are being discovered over time. Heterogeneity, then, only exists in a longitudinal sense in that the fitness function includes progressively more functionalities. Adner and Levinthal (2001), by contrast, reason from demand heterogeneity in a cross-sectional sense, that is, from the coexistence of different user groups with only partially overlapping preference sets. The *explananda* in the two theories are also different. The model by Adner and Levinthal (2001) focuses on the evolutionary dynamics underlying the emergence of a whole new product category when an existing technology finds a new application domain (see also, Adner, 2002; Levinthal, 1998). By contrast, we have looked at technological evolution within a single product category and with special interest in explaining late-mover advantages.

The managerial implications of our model of late-mover advantage are multiple. One managerial implication holds that early entrants in complex product markets run the risk of relying too much on their proprietary component technologies as a source of competitive advantage. Obviously, such a competitive advantage only exists to the extent that potential entrants require such components to develop competing products. However, with evolving user needs, windows of opportunity for new entrants continue to exist, by coming up with alternative product designs taking into account the new functionalities demanded by users. This conclusion from our theoretical model is in line with the related work on disruptive innovation comparing incumbents with newcomers, where newcomers were able to create a mass market while incumbents remained too focused on serving the needs of their existing customers (Christensen, 1997, 2003).

A second managerial implication holds that firms need to continuously monitor evolving user needs and experiment with the integration of new functionalities into their existing product design. In some cases, such new functionalities can be discovered by companies themselves through imagination, experimentation, and consumer surveys. In other cases, new functionalities are discovered by users themselves through innovative user practices. In such contexts, an effective user–producer interaction is crucial to guide the innovation process of a firm (Von Hippel, 1988) as well as the strategy process vis-à-vis the market segments a firm aims to target (Christensen, 1997). Thus, even though users are generally not the source of new technological solutions, their evolving practices signal new needs that are posing serious challenges to incumbents firms.

The more specific managerial lesson, then, holds that firms should avoid restricting R&D resources to the optimization of only a few functionalities demanded by their existing customers, in technologies whose potential uses are not yet well understood. Put differently, as emphasized by Winter et al. (2007), when there is a notion of some global direction of product evolution (here, when new functionalities will be added), local optimization should not undermine this preferred direction of evolution. Furthermore, to the extent that customers are few and large, the monopsony power exerted by them does not only bring down prices but also forces a firm to innovate in a focused direction optimizing the functions demanded by specialized users (Christensen and Bower, 1996). Thus, while specialized users are often critical for the introduction of a new product technology in the first place, a firm would benefit from quickly diversifying its user base as to gain a better and richer understanding of the possible uses of the product technology in question.

These managerial implications, however, are of lesser importance to first movers that still hold a critical resource that renders successful entry of late movers less likely. For example, first movers may have built up complementary assets or proprietary standards with network externalities for users, which may be difficult to build up by late movers. And, although late movers enter with an alternative product design, their new product will typically not be fully different from a first mover's product design, as our model suggested as well. Hence, some of the critical components in a late-mover design may be patented by first movers, rendering successful entry more difficult or costly (Gans et al., 2002).

A final remark on the limitations of our model, our objective was to show that we can formalize the notion that late-mover advantage can stem from evolving user needs. Doing so, we were also able to derive that product complexity renders late-mover advantages in contexts of evolving user needs even bigger. We explicitly abstracted away from some of the other likely mechanisms and conditions that affect firms' competitive advantage, as to be able to show theoretically that evolving user needs alone can already be responsible for late-mover advantages.

Limitations, however, remain. First, there are some minor limitations of a technical nature. For example, we assumed that the time in between the discovery of two functions was long enough to allow the firm to carry out all the mutations required to find a new local peak. In this way, we could conveniently define time in the model solely in terms of the number of functions already discovered. Future modeling may choose to drop this assumption. We also assumed that while the number of functions increases over time, the number of product components remains fixed. Following Altenberg (1994), this assumption can in principle be dropped within a generalized NK-model.

More substantially, the model can be extended by taking into account more mechanisms that are at play in industrial dynamics. Indeed, since this model is an early experiment to model evolving user needs in the first-mover advantage framework, ample room is left for future extensions. For instance, introducing heterogeneity in the demand for functions will provide an alternative framework where more strategic decisions are available for the firms. Such decisions include offering products focused on specific bundles of functions or delaying the release of functions for technical priorities. Second, one can introduce a profit function that feeds back into the level of innovative activity. This would lead one to a different assessment of the relative success of first and late movers. While we now solely focused on the number of mutations as the cost of innovation and the final fitness of the product design as the success of innovation, the explicit introduction of a profit function would also take into account that first movers can reap temporary monopoly profits, which in turn can be invested in innovative activity, while late movers are arguably more constrained financially.

A final limitation has been that we estimated the cost of search to be equal to the number of mutations carried out by a firm in design space, independently of whether another firm already carried out the same mutations before. Late movers do not profit from imitating parts of the designs that first movers already explored before them, that is, we assumed that knowledge spillovers are fully absent. We also did not explicitly model the possibility for first movers to patent parts of their

designs or develop product standards with network externalities, which would severely raise the costs of late movers to enter. As a future extension of our model, one could study how late movers may try to imitate product design from first movers (Rivkin, 2000) and how a first mover's patenting strategy or product standards may influence the opportunities and benefits for imitation for late movers (Marengo et al., 2012).

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was made possible by a grant awarded by the Netherlands Organization of Scientific Research (NWO), under the Complexity program, no. 645.000.007

Notes

1. The evidence presented by Christensen (1997, 2003) is not undisputed. See, in particular, King and Tucci (2002) and King and Baatartogtokh (2015).
2. Initially named RIM (Research In Motion), the company has officially adopted the name of BlackBerry on 9 July 2013.
3. A classic example concerns the bike technology that initially occupied a niche as a racing vehicle for young men and could become a mass product only when safety became a primary functionality (Pinch and Bijker, 1984). For other examples, see Geels (2002).
4. For a review, see Ganco and Hoetker (2009).
5. Alternberg's (1994) generalized model is a generalization of the original NK-model with respect to the number of components and functions that describe a system. Other limitations of the original NK-model still carry over to the generalized NK-model, though, including the assumption that the structure of interdependencies is given and known, and unchangeable by innovative efforts of firms (Bonaccorsi, 2011). Furthermore, although our model can incorporate the discovery of additional functionalities of an existing product over time, it cannot model the emergence of complete new uses of existing artifacts and a corresponding new list of functionalities, which may only partially overlap with the functionalities of previous use contexts. As argued by Felin et al. (2014), such more fundamental innovations in biological and technological evolution alike—sometimes referred to as "speciation," "preadaptation," or "exaptation"—cannot be represented and understood with fitness landscape models such as NK-models.
6. In order to test the robustness of our results, we have compared the results of a *greedy* search with the results of a *random* search, where firms mutate one component randomly and accept it when fitness increases, instead of looking exhaustively for the one mutation that increases the fitness the most as in greedy search. The *greedy* search produces slightly faster and higher fitness improvement than *random* search, but (dis)advantage for first and late movers are the same for both search algorithms.
7. The chances of a component to get fixed early on depend on its pleiotropy: the higher the pleiotropy of a component, the more likely it will influence functions early discovered, the quicker it will get fixed into a particular state.
8. Hence, we let first movers and late movers start their search processes from different, randomly chosen strings. This follows from the idea that late movers face another set of functions than first movers and, hence, start their search "from scratch."
9. The more general conclusion that multi-stage search processes may lead to higher fitness than single-stage search processes was also reached in other NK-model exercises. Siggelkow and Levinthal (2003) showed that a search process that alternates decentralized and centralized search generally leads to higher fitness in the face of environmental shocks. And, Baumann and Siggelkow (2013) showed that "chunky" search focusing on different parts of a complex system in sequential stages, often outperforms the single-stage search strategy.

References

Abernathy WJ and Utterback JM (1978) Patterns of innovation in industry. *Technology Review* 80(7): 40–47.

Adner R (2002) When are technologies disruptive: A demand-based view of the emergence of competition. *Strategic Management Journal* 23(8): 667–688.

Adner R and Levinthal DA (2001) Demand heterogeneity and technology evolution: Implications for product and process innovation. *Management Science* 47(5): 611–628.

Altenberg L (1994) Evolving better representations through selective genome growth. In: *Proceedings of the 1st IEEE conference on evolutionary computation*, Orlando, FL, 27–29 June, pp. 182–187. New York: IEEE.

Auerswald P, Kauffman SA, Lobo J, et al. (2000) The production recipes approach to modeling technological innovation: An application to learning-by-doing. *Journal of Economic Dynamics and Control* 24(3): 389–450.

Baumann O and Siggelkow N (2013) Dealing with complexity: Integrated vs. chunky search processes. *Organization Science* 24(1): 116–132.

Bonaccorsi A (2011) A functional theory of technology and technological change. In: Antonelli C (ed.) *Handbook on the Economic Complexity of Technological Change*. Cheltenham: Edward Elgar, pp. 286–338.

Christensen CM (1997) *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*. Boston, MA: Harvard Business School Press.

Christensen CM (2003) *The Innovator's Solution: Creating and Sustaining Successful Growth*. Boston, MA: Harvard Business School Press.

Christensen CM and Bower JL (1996) Customer power, strategic investment, and the failure of leading firms. *Strategic Management Journal* 17(3): 197–218.

Christensen CM and Rosenbloom R (1995) Explaining the attacker advantage. *Research Policy* 24(2): 233–257.

Clark KB (1985) The interaction of design hierarchies and market concepts in technological evolution. *Research Policy* 14(5): 235–251.

Dowell G and Swaminathan A (2006) Entry timing, exploration, and firm survival in the early US bicycle industry. *Strategic Management Journal* 27(12): 1159–1182.

Ethiraj SK and Levinthal DA (2004) Modularity and innovation in complex systems. *Management Science* 50(2): 159–173.

Ethiraj SK, Levinthal DA and Roy RR (2008) The dual role of modularity: Innovation and imitation. *Management Science* 54(5): 939–955.

Ethiraj SK, Ramasubbu N and Krishnan MS (2012) Does complexity deter customer-focus? *Strategic Management Journal* 33(2): 137–161.

Felin T, Kauffman S, Koppl R, et al. (2014) Economic opportunity and evolution: Beyond landscapes and bounded rationality. *Strategic Entrepreneurship Journal* 8(4): 269–282.

Frenken K and Nuvolari A (2004) The early development of the steam engine: An evolutionary interpretation using complexity theory. *Industrial and Corporate Change* 13(2): 419–450.

Ganco M (2013) Cutting the Gordian knot: The effect of knowledge complexity on employee mobility and entrepreneurship. *Strategic Management Journal* 34(6): 666–686.

Ganco M and Hoetker G (2009) NK modeling methodology in the strategy literature: Bounded search on a rugged landscape. In: Bergh D and Ketchen D (ed.) *Research Methodology in Strategy and Management—Volume 5*. Bingley: Emerald Group Publishing, pp. 237–268.

Gans J, Hsu D and Stern S (2002) When does start-up innovation spur the gale of creative destruction? *RAND Journal of Economics* 33(4): 571–586.

Geels FW (2002) Technological transitions as evolutionary reconfiguration processes: A multi-level perspective and a case-study. *Research Policy* 31(8–9): 1257–1274.

Kauffman SA (1993) *The Origins of Order: Self-Organization and Selection in Evolution*. New York: Oxford University Press.

Kenney M and Pon B (2011) Structuring the smartphone industry: Is the mobile internet OS platform the key? *Journal of Industry, Competition and Trade* 11(3): 239–261.

King A and Baatartogtokh B (2015) How useful is the theory of disruptive innovation? *Sloan Management Review (Fall issue)*, 15 September. Available at: http://sloanreview.mit.edu/article/how-useful-is-the-theory-of-disruptive-innovation/

King A and Tucci C (2002) Incumbent entry into new market niches: The role of experience and managerial choice in the creation of dynamic capabilities. *Management Science* 48(2): 171–186.

Klepper S (1996) Entry, exit, growth, and innovation over the product cycle. *American Economic Review* 86(3): 562–583.

Klepper S and Sleeper S (2005) Entry by spinoffs. *Management Science* 51(8): 1291–1306.

Knudsen T and Levinthal DA (2007) Two faces of search: Alternative generation and alternative evaluation. *Organization Science* 18(1): 39–54.

Lenox M, Rockart S and Lewin A (2007) Interdependency, competition, and industry dynamics. *Management Science* 53(4): 599–615.

Levinthal DA (1997) Adaptation on rugged landscapes. *Management Science* 43(7): 934–950.

Levinthal DA (1998) The slow pace of rapid technological change: Gradualism and punctuation in technological change. *Industrial and Corporate Change* 7(2): 217–247.

Lie M and Sørensen KH (1996) *Making Technology Our Own? Domesticating Technology into Everyday Life*. Oslo: Scandinavian University Press.

Lieberman M (2013) First-mover advantage. *Palgrave Encyclopedia of Strategic Management*, 19 December. Available at: http://www.palgraveconnect.com/esm/doifinder/10.1057/9781137294678.0232

Lieberman MB and Montgomery DB (1988) First-mover advantages. *Strategic Management Journal* 9(1): 41–58.

Lieberman MB and Montgomery DB (1998) First-mover (dis)advantages: Retrospective and link with the resource-based view. *Strategic Management Journal* 19(12): 1111–1125.

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

Marengo L, Pasquali C, Valente M, et al. (2012) Appropriability, patents, and rates of innovation in complex products industries. *Economics of Innovation and New Technology* 21(8): 753–773.

Markides C and Geroski P (2005) *Fast Second: How Smart Companies Bypass Radical Innovation to Enter and Dominate New Markets*. San Francisco, CA: Jossey-Bass.

Nelson RR and Winter SG (1982) *An Evolutionary Theory of Economic Change*. Cambridge, MA: The Belknap Press.

Pinch TJ and Bijker WE (1984) The social construction of facts and artefacts: Or how the sociology of science and the sociology of technology might benefit each other. *Social Studies of Science* 14(3): 399–441.

Rivkin JW (2000) Imitation of complex strategies. *Management Science* 46(6): 824–844.

Rivkin JW and Siggelkow N (2003) Balancing search and stability: Interdependencies among elements of organizational design. *Management Science* 49(3): 290–311.

Siggelkow N and Levinthal DA (2003) Temporarily divide to conquer: Centralized, decentralized, and reintegrated organizational approaches to exploration and adaptation. *Organization Science* 14(6): 650–669.

Silverstone R and Hirsch E (1992) *Consuming Technologies: Media and Information in Domestic Spaces*. London; New York: Routledge.

Simon HA (2002) Near decomposability and the speed of evolution. *Industrial and Corporate Change* 11(3): 587–599.

Suarez FF and Lanzolla G (2007) The role of environmental dynamics in building a first-mover advantage theory. *Academy of Management Review* 32(2): 377–392.

Suarez FF, Grodal S, and Gotsopoulos A (2015) Perfect timing? Dominant category, dominant design, and the window of opportunity for firm entry. *Strategic Management Journal* 36(3): 437–448.

Teece DJ (1986) Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. *Research Policy* 15(6): 285–305.

Von Hippel E (1988) *The Sources of Innovation*. Oxford: Oxford University Press.

West J and Mace M (2010) Browsing as the killer app: Explaining the rapid success of Apple's iPhone. *Telecommunications Policy* 34(5): 270–286.

Winter SG, Cattani G and Dorsch A (2007) The value of moderate obsession: Insights from a new model of organizational search. *Organization Science* 18(3): 403–419.

Wright S (1932) The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In: *Proceedings of the sixth international congress on genetics*, Ithaca, New York, 24–31 August 1932, vol. 1, pp. 355–366. Brooklyn, New York: Brooklyn Botanic Garden.

Author biographies

Adrien Querbes is a Fellow in the Department of Social and Decision Sciences at Carnegie Mellon University. He received a PhD in economics of innovation from the University of Bordeaux (France). His research studies innovation processes as the emergence of new ideas, as well as the dissemination/adoption of these new ideas. His approach includes the use of modeling methods based on evolutionary and complex systems, adapted to social sciences. His research has been presented at various computational and evolutionary social sciences conferences and published in *PLOS ONE* and *the Journal of Innovation Economics and Management*.

Koen Frenken is a Full Professor in Innovation Studies at the Copernicus Institute of Sustainable Development, Utrecht University. He received a joint PhD degree from the University of Grenoble (applied economics) and the University of Amsterdam (social sciences). He won personal grants in the Marie Curie, NWO VIDI and NWO VICI funding schemes and published in *Research Policy, Technovation, PLOS ONE, Economic Geography, Journal of Economic Geography, Regional Studies*, and other journals. He is also the author of *Innovation, Evolution and Complexity Theory* (2006, Edward Elgar) and the editor of *Applied Evolutionary Economics and Economic Geography* (2007, Edward Elgar). Theoretical interests include evolutionary economics, complexity theory, and network science. Empirically, he mainly works on innovation, diffusion, economic geography, and sharing economy. He also holds an interest in science, technology and innovation policy.

Appendix 1

Table 1. Example of fitness values for the product depicted in Figure 1.

	φ_1	φ_2	φ
0000	0.320	0.970	0.645
0001	0.260	0.720	0.490
0010	0.320	0.140	0.230
0011	0.260	0.040	0.150
0100	0.410	0.530	0.470
0101	0.470	0.860	0.665
0110	0.410	0.240	0.325
0111	0.470	0.980	0.725
1000	0.770	0.970	0.870
1001	0.820	0.720	0.770
1010	0.770	0.140	0.455
1011	0.820	0.040	0.430
1100	0.890	0.530	0.710
1101	0.420	0.860	0.640
1110	0.890	0.240	0.565
1111	0.420	0.980	0.700

Table 2.

Late mover procedure
$t = 2$
Agent chooses a random string to start search: 0011 with $\varphi = 0.150$. Greedy search leads to a mutation to 0111 with $\varphi = 0.725$, which is locally optimal with regard to φ.
Final product design: 0111 (**φ = 0.725)** Number of mutations required: 1
First mover procedure ("functional inertia")
$t = 1$
Agent chooses a random string to start search: 0010 with $\varphi_1 = 0.320$. Greedy search leads to a mutation to 1010 with $\varphi_1 = 0.770$. Greedy search leads to a mutation to 1110 with $\varphi_1 = 0.890$, which is locally optimal with regard to φ_1.
$t = 2$
Agent discovers the second function. For current technology 1110, $\varphi_2 = 0.240$. Only mutation in component 3 is allowed, since other components influence φ_1. Greedy search with regard to φ_2 leads to a mutation to 1100 with $\varphi_2 = 0.530$, which is locally optimal with regard to φ_2.
Final product design: 1100 (**φ = 0.710**) Number of mutations required: 3 Product similarity with late mover: 25%
First mover procedure ("functional flexibility")
$t = 1$
Agent chooses a random string to start search: 0010 with $\varphi_1 = 0.320$. Greedy search leads to a mutation to 1010 with $\varphi_1 = 0.770$. Greedy search leads to a mutation to 1110 with $\varphi_1 = 0.890$, which is locally optimal with regard to φ_1.
$t = 2$
Agent discovers the second function. Mutations are allowed in all components. Greedy search with regard to φ leads to a mutation to 1100 with $\varphi = 0.710$. Greedy search with regard to φ leads to a mutation to 1000 with $\varphi = 0.870$, which is locally optimal with regard to φ.
Final product design: 1000 (**φ = 0.870**) Number of mutations required: 4 Product similarity with late mover: 0%

Appendix 2

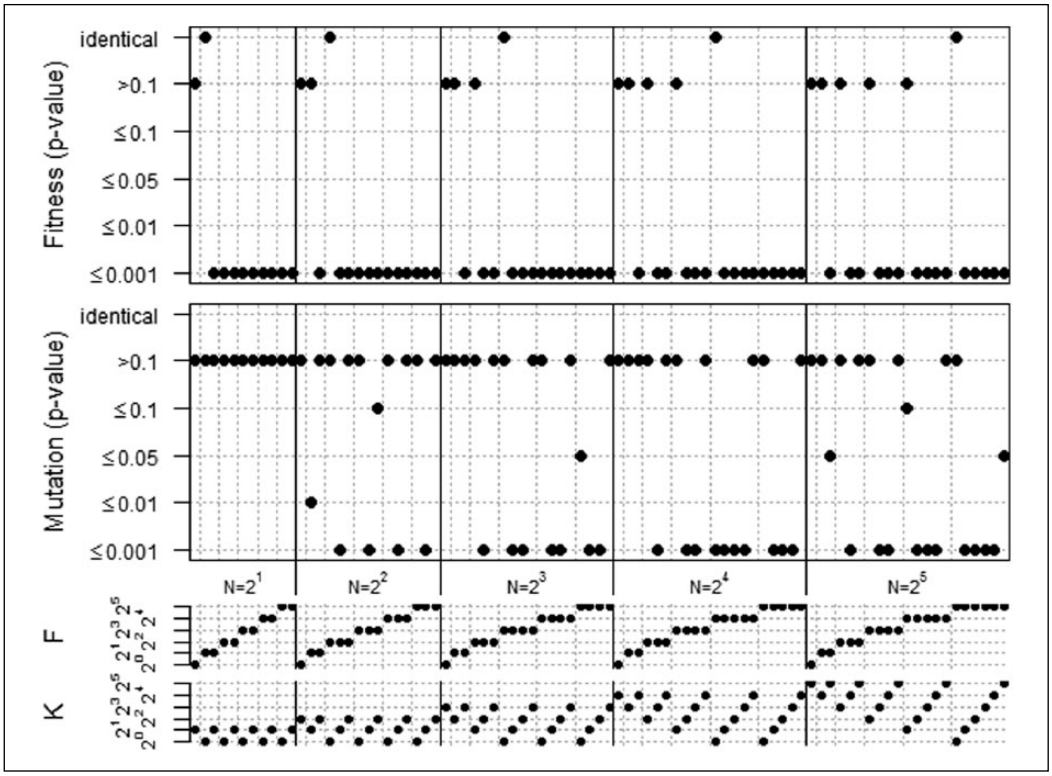

Figure 4. The p-values of the Wilcoxon signed-rank test on the paired difference of fitness and mutations, between first and late movers exploring the same landscape. The null hypothesis holds that the distribution of the differences is symmetric about 0 (i.e. no differences between first and late movers), the alternative is that they differ.

Appendix 3

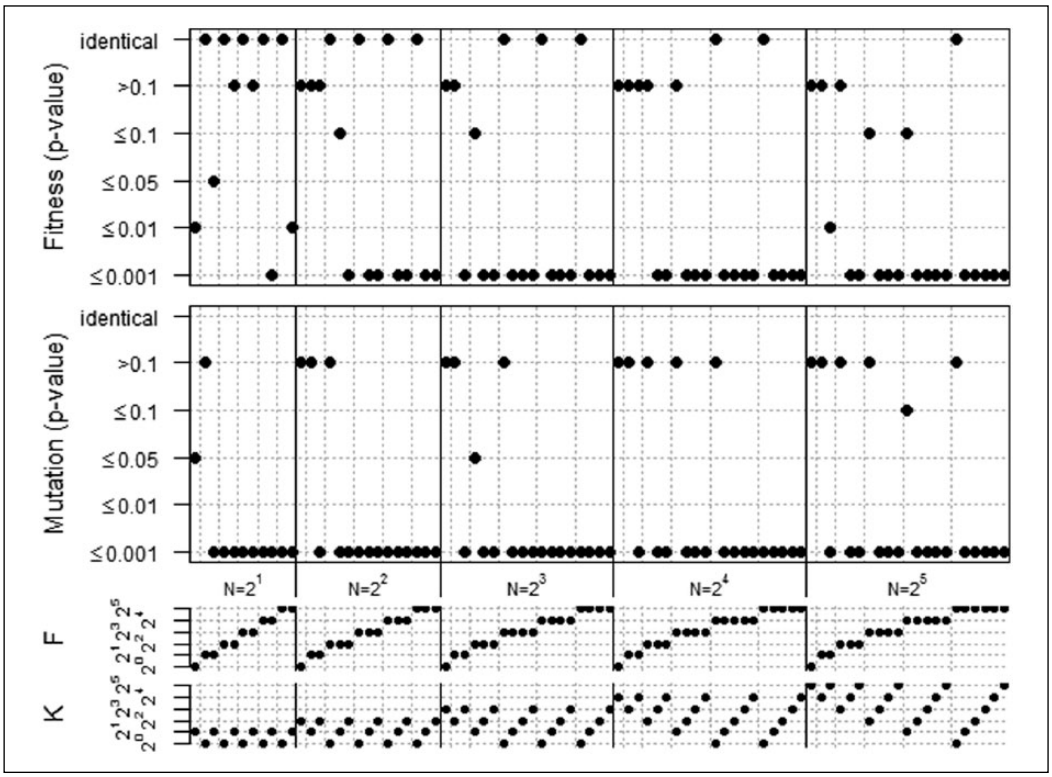

Figure 5. The p-values of the Wilcoxon signed-rank test on the paired difference of fitness and mutations, between first and late movers exploring the same landscape. The null hypothesis holds that the distribution of the differences is symmetric about 0 (i.e. no differences between first and late movers), the alternative is that they differ.

Appendix 4

For completeness, we also study all the intermediate sequences of functionality discovery for the subset of parameters $N = F = 2^4$ and $K = \{2^0, 2^1, 2^2, 2^3, 2^4\}$. This includes the first and late movers as originally defined, whose sequences can be represented mathematically as $\{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1\}$ and $\{16\}$, respectively. Hence, these can be interpreted as the extremes of the sequence set. In total, for 16 functionalities, there are 228 sequences (including first and late mover), which are each simulated 5000 times. To compare their results, we characterize each sequence by the Simpson Index

$$\text{Simpson Index} = \frac{\sum_t f_t \times (f_t - 1)}{F \times (F - 1)} \tag{3}$$

Where f_t is the number of functions discovered at a given time step t. Hence, the Index takes the value 0 for the first mover, 1 for the late mover, and for instance 0.65 for $\{4,2,8,2\}$ or 0.875 for

{15,1}. It gives us the probability that two functions are discovered at the same time step, without replacement.

The results are presented in Figure 6 for the case of functional inertia of the first mover and Figure 7 for functional flexibility of the first mover. We observe in Figure 6 a nearly linear relationship between the Simpson Index and our three variables of interest. Hence, the late-mover advantage can be generalized as an advantage of any agents that improves a product while taking into account sets of functionalities larger than its competitors. And, in Figure 7, we also observe a clear correlation between the Simpson Index and our three variables of interest.

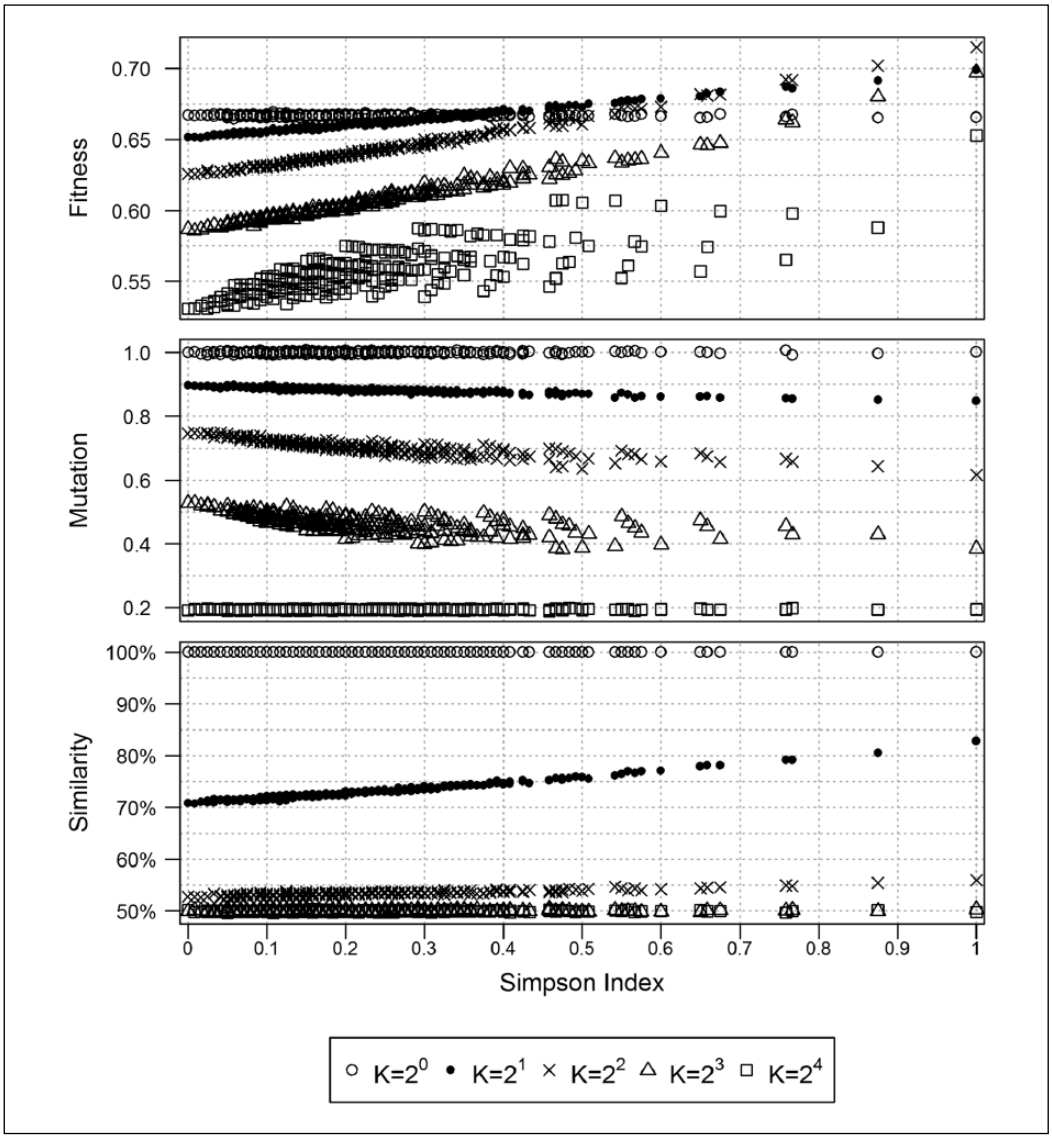

Figure 6. Average fitness, average normalized number of mutations, and product similarity. Each sequence of function discovery is represented by its Simpson Index, from first mover (Index = 0) to late mover (Index = 1). All follow a "functional inertia strategy."

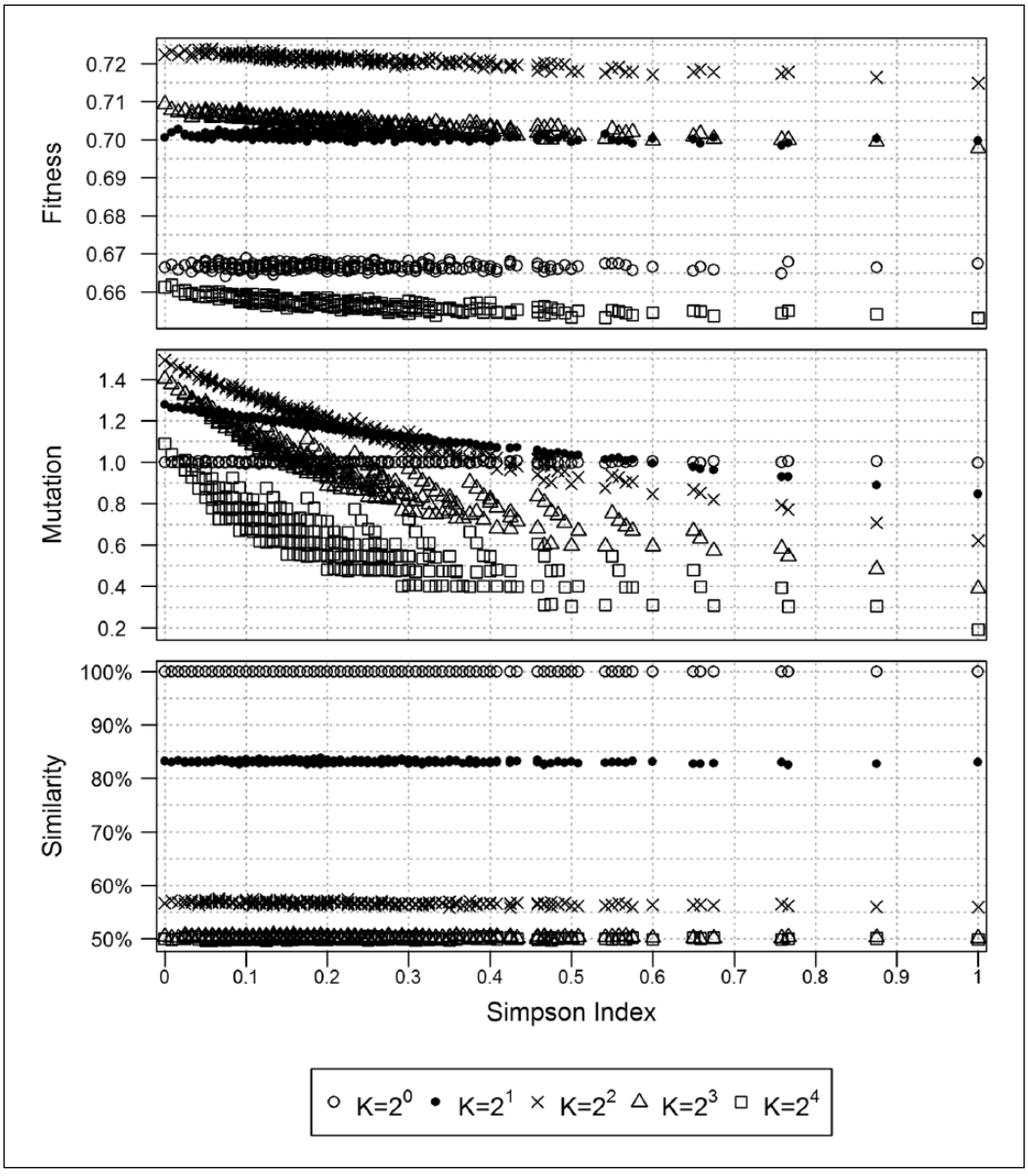

Figure 7. Average fitness, average normalized number of mutations, and product similarity. Each sequence of function discovery is represented by its Simpson Index, from first mover (Index = 0) to late mover (Index = 1). All follow a "functional flexibility strategy."