Role of design complexity in technology improvement

James McNerney[a,b], J. Doyne Farmer[a,c], Sidney Redner[a,b,d], and Jessika E. Trancik[a,e,1]

[a]Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501; [b]Department of Physics, Boston University, Boston, MA 02215; [c]Libera Universitá Internazionale degli Studi Sociali, Guido Carli, Viale Pola 12, 00198 Rome, Italy; [d]Center for Polymer Studies, Boston University, Boston, MA 02215; and [e]Engineering Systems Division, Massachusetts Institute of Technology, Cambridge, MA 02139

Edited by Hans-Joachim Schellnhuber, Potsdam Institute for Climate Impact Research (PIK), Potsdam, Germany, and approved March 8, 2011 (received for review November 22, 2010)

We study a simple model for the evolution of the cost (or more generally the performance) of a technology or production process. The technology can be decomposed into *n* components, each of which interacts with a cluster of *d* − 1 other components. Innovation occurs through a series of trial-and-error events, each of which consists of randomly changing the cost of each component in a cluster, and accepting the changes only if the total cost of the cluster is lowered. We show that the relationship between the cost of the whole technology and the number of innovation attempts is asymptotically a power law, matching the functional form often observed for empirical data. The exponent α of the power law depends on the intrinsic difficulty of finding better components, and on what we term the design complexity: the more complex the design, the slower the rate of improvement. Letting *d* as defined above be the connectivity, in the special case in which the connectivity is constant, the design complexity is simply the connectivity. When the connectivity varies, bottlenecks can arise in which a few components limit progress. In this case the design complexity depends on the details of the design. The number of bottlenecks also determines whether progress is steady, or whether there are periods of stasis punctuated by occasional large changes. Our model connects the engineering properties of a design to historical studies of technology improvement.

design structure matrix | experience curve | learning curve | performance curve

The relation between a technology's cost c and the cumulative amount produced y is often empirically observed to be a power law of the form

$$c(y) \propto y^{-\alpha}, \qquad [1]$$

where the exponent α characterizes the rate of improvement. This rate is commonly termed the progress ratio $2^{-\alpha}$, which is the factor by which costs decrease with each doubling of cumulative production. A typical reported value (1) is 0.8 (corresponding to $\alpha \approx .32$), which implies that the cost of the 200th item is 80% that of the 100th item. Power laws have been observed (or at least assumed to hold), for a wide variety of technologies (1–3), although other functional forms have also been suggested and in some cases provide plausible fits to the data*. We give examples of historical performance curves for several different technologies in Fig. 1.

The relationship between cost and cumulative production goes under several different names, including the "experience curve," the "learning curve," or the "progress function." The terms are used interchangeably by some, whereas others assign distinct meanings (1, 4). We use the general term performance curve to denote a plot of any performance measure (such as cost) against any experience measure (such as cumulative production), regardless of the context. Performance curve studies first appeared in the 19th century (5, 6), but their application to manufacturing and technology originates from the 1936 study by Wright on aircraft production costs (7). The large literature on this subject spans engineering (8), economics (4, 9), management science (1), organizational learning (16), and public policy (17). Performance curves have been constructed for individuals, production processes, firms, and industries (1).

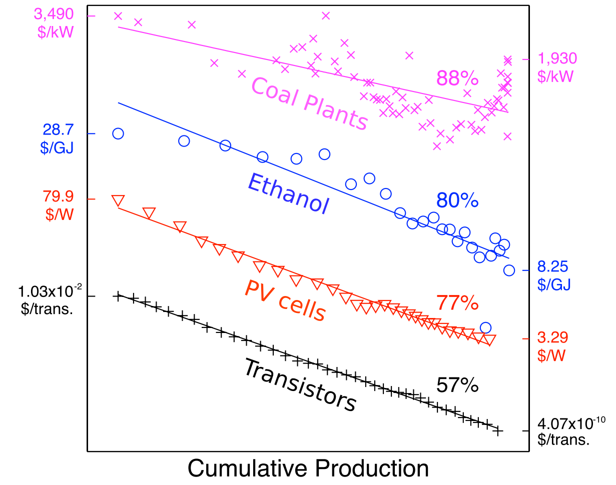


Fig. 1. Four empirical performance curves. Each curve was rescaled and shifted to aid comparison with a power law. The *x*- and *y*- coordinates of each series *i* were transformed via $\log x \to a_i + b_i \log x$, $\log y \to c_i + d_i \log y$. The constants a_i, b_i, c_i, and d_i were chosen to yield series with approximately the same slope and range, and are given in *SI Text*. Tick marks and labels on the left vertical axis show the first price (in real 2000 dollars) of the corresponding time series, and those of the right vertical axis show the last price. Lines are least-squares fits to a power law. Percentages are the progress ratios of the fitted power laws. Source: coal plants (10), ethanol (11), photovoltaic cells (12, 13, 14), transistors (15).

The power law assumption has been used by firm managers (18) and government policy makers (17) to forecast how costs will drop with cumulative production. However, the potential for exploiting performance curves has so far not been fully realized, in part because there is no good theory explaining the observed empirical relationships. Why do performance curves tend to look like power laws, as opposed to some other functional form? What factors determine the exponent α, which governs the long-term rate of improvement? Why are some performance curves steady and others erratic? By suggesting answers to these questions, the theory we develop here can potentially be used to guide investment policy for technological change.

Author contributions: J.M., J.D.F., S.R., and J.E.T. performed research and wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

Freely available online through the PNAS open access option.

*Koh and Magee (35, 36) claim an exponential function of time (Moore's law) predicts the performance of several different technologies. Goddard (34) claims costs follow a power law in production rate rather than cumulative production. Multivariate forms involving combinations of production rate, cumulative production, or time have been examined by Sinclair et al. (38) and Nordhaus (37).

[1]To whom correspondence should be addressed. E-mail: trancik@mit.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1017298108/-/DCSupplemental.

An example of the possible usefulness of such a theory is climate change mitigation. Good forecasts of future costs of low-carbon energy technologies could help guide research and development funding and climate policy. Our theory suggests that based on the design of a technology we might be able to better forecast its rate of improvement, and therefore make better investments and better estimates of the cost of achieving low-carbon energy conversion.

There have been several previous attempts to construct theories to explain the functional form of performance curves (19–21). Muth constructed a model of a single-component technology in which innovation happens by proposing new designs at random (21). Using extreme value theory he derived conditions under which the rate of improvement is a power law. An extension to multiple components, called the production recipe model, was proposed by Auerswald et al. (19). In their model each component interacts with other components, and if a given component is replaced, it affects the cost of the components with which it interacts. They simulated their model and found that under some circumstances the performance curves appeared to be power laws. Other models include Bendler and Schlesinger, who derive a power law based on the assumption that barriers to improvement are distributed fractally (22), and Huberman, who represents the design process as a random evolving graph (20). More recently Frenken has used the Auerswald model to interpret and address questions such as the efficacy of outsourcing (23, 24). Other related models that use random search to model technological progress (but which do not directly address performance curves) are those of Silverberg and Verspagen (25, 26) and Thurner et al. (27).

In this paper we both simplify and extend the production recipe model of Auerswald et al. (19). The simplifications allow us to derive the emergence of a power law, and most importantly, to derive its exponent α. We find that $\alpha = 1/(\gamma d^*)$, where γ measures the intrinsic difficulty of finding better components and d^* is what we call the design complexity. When the connectivity of the components is constant the design complexity d^* is equal to the connectivity. When connectivity is variable, the complexity can also depend on the detailed properties of the design, in ways that we make clear. We also show that when costs are spread uniformly across a large number of components, the whole technology undergoes steady improvement. In contrast, when costs are dominated by a few components, the total cost undergoes erratic improvement. Our theory thus potentially gives insight into how to design a technology so that it will improve more rapidly and more steadily.

We should emphasize that many factors besides design can affect costs—for example, the cost of input materials or fuels may change due to market dynamics rather than technology design (10). Furthermore, design is generally focused not just on reducing costs, but also on improving other properties such as environmental performance or reliability. The variable "cost" in the theory here can be interpreted as any property that depends on technology design.

The Model

The production design consists of n components, which can be thought of as the parts of a technology or the steps in an industrial process[†]. Each component i has a cost c_i. The total cost κ of the design is the sum of the component costs: $\kappa = c_1 + c_2 + \cdots + c_n$. A component's cost changes as new implementations for the component are found. For example, a component representing the step "move a box across a room" may initially be implemented by a forklift, which could later be replaced by a conveyor belt. Cost reductions occur through repeated changes to one or more components.

Components are not isolated from one another, but rather interact as parts of the overall design. Thus changing one component not only affects its cost, but also the costs of other dependent components. Components may be viewed as nodes in a directed network, with links from each component to those that depend on it. The relationship between the nodes and links can alternatively be characterized by an adjacency matrix. In systems engineering and management science this matrix is known as the design structure matrix (DSM) (28–30). A DSM is an $n \times n$ matrix with an entry in row i and column j if a change in component j affects component i (Fig. 2). The matrix is usually binary (31, 32); however, weighted interactions have also been considered (33). DSMs have been found to be useful in understanding and improving complex manufacturing and technology development processes.

The model is simulated as follows:

1. Pick a random component i.
2. Use the DSM to identify the set of components $\mathscr{A}_i = \{j\}$ whose costs depend on i (the outset of i).
3. Determine a new cost c_j' for each component $j \in \mathscr{A}_i$ from a fixed probability distribution f.
4. If the sum of the new costs, $a_i' = \sum_{j \in \mathscr{A}_i} c_j'$, is less than the current sum, a_i, then each c_j is changed to c_j'. Otherwise, the new cost set is rejected.

This process is repeated for t steps. The costs are defined on [0,1]. We assume a probability density function that for small values of c_i has the form $f(c_i) \propto c_i^{\gamma-1}$; i.e., the cumulative distribution $F(c_i) = \int_0^{c_i} f(c)dc \propto c_i^{\gamma}$. The exponent γ specifies the difficulty of reducing costs of individual components, with higher γ corresponding to higher difficulty. This functional form is fairly general in that it covers any distribution with a power-series expansion at $c = 0$.

Independent Components

We first consider the simple but unrealistic case of a technology with n independent components. This generalizes the one component case originally studied by Muth (21). The cost of a given component at time t is equivalent to the minimum of t independent, identically distributed random variables. In *SI Text*, we use extreme value theory to show that to first order in n/t the expected cost $E[\kappa(t)]$ is

$$E[\kappa(t)] = \Gamma\left(1 + \frac{1}{\gamma}\right)\left(\frac{t}{n}\right)^{-1/\gamma}, \qquad [2]$$

where $\Gamma(a)$ is Euler's gamma function.

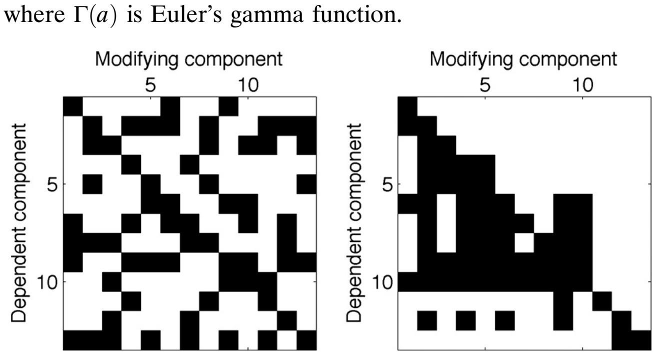


Fig. 2. Example design structure matrices (DSMs) with $n = 13$ components. Black squares represent links. The DSM on the left was randomly generated to have fixed out-degree for each component. The DSM on the right represents the design of an automobile brake system (31). All diagonal elements are present because a component always affects its own cost.

[†]The original production recipe model (19) contained 6 parameters. We eliminated four of them as follows: length of production run $T \to \infty$; output-per-attempted-recipe-change $\hat{B} \to 1$; available implementations per component $s \to \infty$; search distance $\delta \to 1$.

To understand intuitively why the expected cost decreases as a power law in time, consider the simple example where $\gamma = 1$. In this case at each innovation attempt a new cost κ' is drawn uniformly from [0,1], and a successful reduction occurs if κ' is less than the current cost κ. Because the distribution of new costs is uniform on [0,1] the probability $\mathrm{Prob}(\kappa' < \kappa)$ that κ' represents a reduction simply equals κ. When a reduction does occur, the average value of κ' is $\kappa/2$. Making the approximation that $E[\kappa^2] = E[\kappa]^2$, in continuous time the rate of change of the average component cost is

$$\frac{dE[\kappa]}{dt} \sim -\left(\frac{E[\kappa]}{2}\right) \times \mathrm{Prob}(\kappa' < \kappa) = -\frac{1}{2}E[\kappa]^2. \qquad [3]$$

The solution to Eq. **3** gives the correct scaling $\kappa(t) \sim 1/t$ as $t \to \infty$. The cost reductions are proportional to the cost itself, leading to an exponential decrease in cost with each reduction; however, each reduction takes exponentially longer to achieve as the cost decreases. The competition between these two exponentials yields a power law.

Interacting Components, Fixed Out-Degree

Now consider an n-component process with fixed out-degree, where each component affects exactly $d - 1$ other components, in addition to affecting itself. Whether or not a given component will improve in a given trial strongly depends on the other components in its cluster. Consequently, the costs are no longer independent. If the design structure matrix D_{ij} is invertible the total cost κ can be decomposed as

$$\kappa = \sum_{j=1}^{n} c_j = \sum_{i,j} D_{ij}^{-1} a_i = \sum_{i}^{n} k_i a_i, \qquad [4]$$

where $a_i = \sum_{j \in \mathscr{A}_i} c_j$ is the cost of cluster i and $k_i \equiv \sum_j D_{ij}^{-1}$. Because the interaction of components inside the same cluster is much stronger than that of components in different clusters, we can make the approximation that clusters evolve independently. In *SI Text*, we derive the approximate behavior using two different methods, one based on extreme value theory and the other based on a differential equation for $E[\kappa]$. In the latter case we find

$$E[\kappa(t)] = \left[\left(\frac{d}{n}\right)^{\gamma d+1} \frac{\gamma d}{1+\gamma d} H(n,d,\gamma)t + 1\right]^{-1/(\gamma d)}, \qquad [5]$$

where

$$H(n,d,\gamma) \equiv \frac{(\gamma n^{\gamma})^d}{\gamma d} \frac{\Gamma(\gamma)^d}{\Gamma(\gamma d)}. \qquad [6]$$

We compare our prediction in Eq. **5** to simulations in Fig. 3. Initially each component cost c_i is set to $1/n$, so that the total initial cost $c(0) = 1$, and we choose $\gamma = 1$ for simplicity. Eq. **5** correctly predicts the asymptotic power law scaling of the simulated performance curves, as well as the deviation from power law behavior for short times. (As shown in *SI Text*, the extreme value method also predicts the correct asymptotic scaling.)

The salient result is that the exponent $\alpha = 1/(\gamma d)$ of the performance curve is directly and simply related to the out-degree d, which can be viewed as a measure of the complexity of the design, and γ, which characterizes the difficulty of improving individual components in the limit as the cost goes to zero. If $\gamma d = 1$ then $\alpha = 1$ and the progress ratio $2^{-1/(\gamma d)}$ is 50%. If $\gamma d = 3$ then $\alpha = 1/3$ and the progress ratio is approximately 80%, a common value observed in empirical performance curves.

This d dependence has a simple geometric explanation. Consider the case where $\gamma = 1$. Drawing d new costs independently is equivalent to picking a point with uniform probability in a

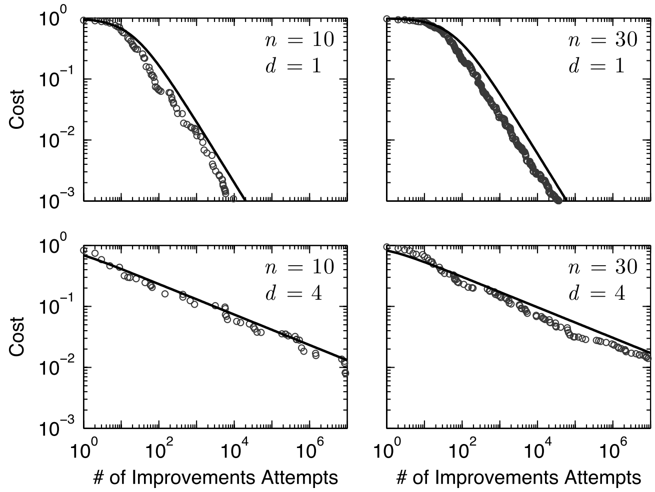


Fig. 3. Comparison of the cost for a simulation of a single realization of the production recipe model (circles) to the predicted expected cost $E[\kappa(t)]$ from Eq. **5** (solid curve), for the case of constant out-degree. n is the number of components and d is the connectivity.

d-dimensional hypercube. The combinations of component costs that reduce the total cost lie within the simplex defined by $\sum_{k \in \mathscr{A}_i} c'_k < a_i$, where c'_k are the new costs. The probability of reducing the cost is therefore the ratio of the simplex volume to the hypercube volume,

$$p = \frac{(a_i^d/d!)}{(1/n)^d} = \frac{(na_i)^d}{d!}, \qquad [7]$$

which is a decreasing function of d. Thus a component with higher out-degree (greater connectivity) is less likely to be improved when chosen.

Interacting Components, Variable Out-Degree

When the out-degree of each component is variable the situation is more interesting and more realistic because components may differ in their rate of improvement (31). Slowly improving components can create bottlenecks that hinder the overall rate of improvement. In this case it is no longer a good approximation to treat clusters as evolving independently.

As illustrated in Fig. 4, there are two ways to reduce the cost of a given component i:

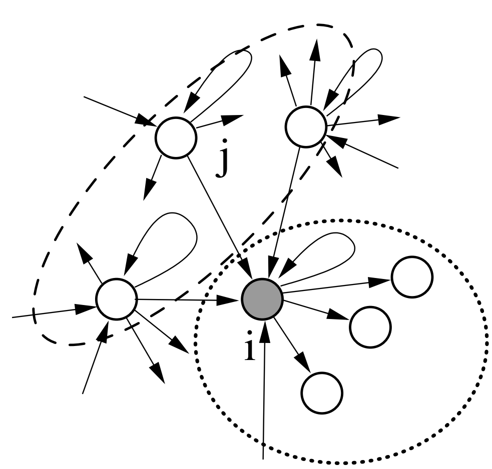


Fig. 4. A component i (shaded circle), together with the components $\mathscr{A}_i$ that are affected by i (dotted ellipse) and the components that affect i (dashed ellipse). The arrow from j to i indicates that a change in cost of component j affects the cost of i.

1. Pick i and improve cluster $\mathscr{A}_i$.
2. Pick component j in the inset of i and improve cluster $\mathscr{A}_j$.

From Eq. **7**, if component i has a large out-degree d_i, it is relatively unlikely to be improved by process 1. Nonetheless, if j has low out-degree, then i will improve more rapidly via process 2. Let d_j^i be the out-degree of component j, which is in the inset of i. Then the overall improvement rate of component i is determined by $d_i^{\min} = \min_j \{d_j^i\}$; i.e., it is driven by the out-degree of the component j in its inset whose associated cluster $\mathscr{A}_j$ is most likely to improve. In *SI Text*, we demonstrate numerically that asymptotically $E[c_i] \sim t^{-1/d_i^{\min}}$. As t becomes large, the difference in component costs can become quite dramatic, with the components with the largest values of $d_i^{\min}$ dominating. The overall improvement rate for the whole technology is then determined by the slowest-improving components, governed by the design complexity

$$d^* = \max_i \{d_i^{\min}\}. \quad \textbf{[8]}$$

We call any component with $d_i^{\min} = d^*$ a bottleneck. When t is large one can neglect all but the bottleneck components, and as we show in *SI Text*, the average total cost scales as $E[\kappa] \sim t^{-1/d^*}$. Note that in the case of constant out-degree d Eq. **8** reduces to $d^* = d$.

To test this hypothesis we randomly generated 90 different DSMs with values of d^* ranging from 1 to 9 and $\gamma = 1$, simulated the model 300 times for each DSM, measured the corresponding average rate of improvement, and compared with that predicted from the theory. We find good agreement in every case, as demonstrated in *SI Text*.

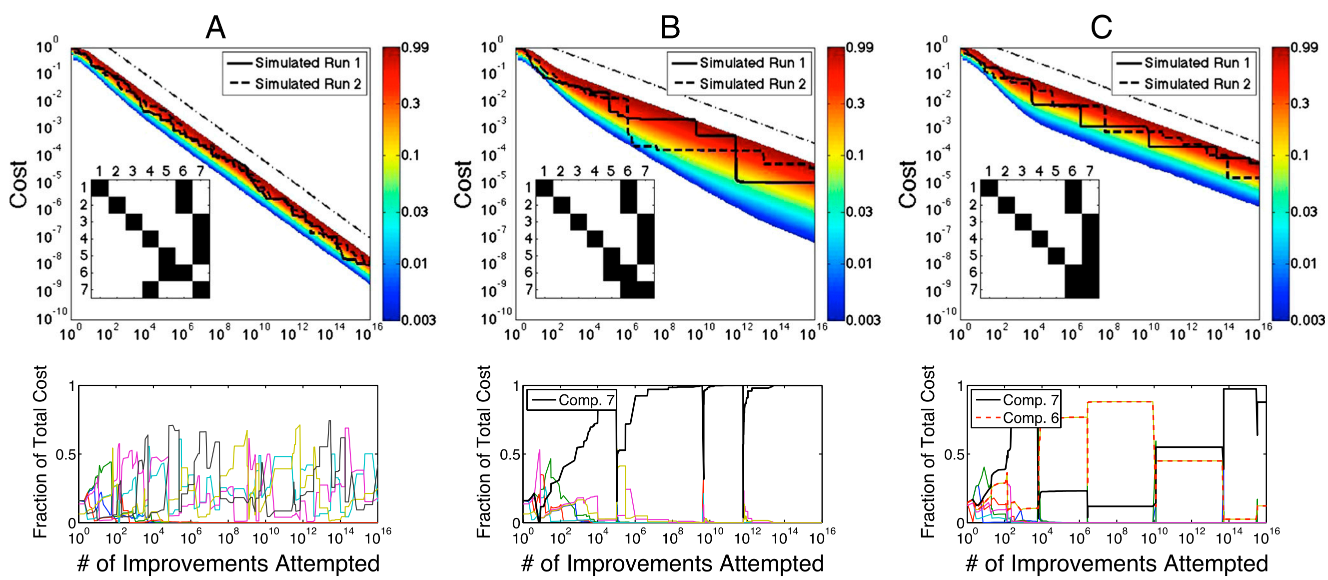
Fluctuations

The analysis we have given provides insight not only about the mean behavior, but also about fluctuations about the mean. These can be substantial, and depend on the properties of the DSM. In Fig. 5 we plot two individual trajectories of cost vs. time for each of three different DSMs. The trajectories fluctuate in every case, but the amplitude of fluctuations is highly variable. In Fig. 5 *Left* the amplitude of the fluctuations remains relatively small and is roughly constant in time when plotted on double logarithmic scale (indicating that the amplitude of the fluctuations is always proportional to the mean). For Fig. 5 *Center* and *Right*, in contrast, the individual trajectories show a random staircase behavior, and the amplitude of the fluctuations continues to grow for a longer time.

This behavior can be explained in terms of the improvement rates $d_i^{\min}$ for each component. The maximum value of $d_i^{\min}$ determines the slowest-improving components. In Fig. 5 *Left* the maximum value of $d_i^{\min} = 2$. This value occurs for four components. After a long time these four components dominate the overall cost. However, because they have the same values of $d_i^{\min}$ their contributions remain comparable, and the total cost is averaged over all four of them, keeping the fluctuations relatively small. (See Fig. 5 *Lower*.)

In contrast, in Fig. 5 *Center* we illustrate a DSM where the slowest-improving component (number 7) has $d_i^{\min} = 4$ and the next slowest-improving component (number 6) has $d_i^{\min} = 2$. With the passage of time component 7 comes to dominate the cost. This component is slow to improve because it is rarely chosen for improvement. But in the rare cases that component 7 is chosen the improvements can be dramatic, generating large downward steps in its trajectory. The right case illustrates an intermediate situation where two components are dominant.

Another striking feature of the distribution of trajectories is the difference between the top and bottom envelopes of the plot



component i	1	2	3	4	5	6	7
out-degree of i	1	1	1	2	2	3	4
$d_i^{\min}$	**1**	**1**	**1**	**2**	**2**	**2**	**2**

component i	1	2	3	4	5	6	7
out-degree of i	1	1	1	1	2	4	4
$d_i^{\min}$	**1**	**1**	**1**	**1**	**2**	**2**	**4**

component i	1	2	3	4	5	6	7
out-degree of i	1	1	1	1	1	4	5
$d_i^{\min}$	**1**	**1**	**1**	**1**	**1**	**4**	**4**

Fig. 5. Evolution of the distribution of costs. Each figure in the top row shows a simulated distribution of costs as a function of time using the DSM in the lower left corner of each plot. The upper dash-dot lines provides a reference with the predicted slope $\alpha = 1/(\gamma d^*)$, with $\gamma = 1$; from left to right the slopes are $-1/2$, $-1/4$, and $-1/4$. The data for each DSM are the result of 50,000 realizations, corresponding to different random number seeds. The distributions are color coded to correspond to constant quantiles; i.e., the fraction of costs less than a given value at a given time. The solid and dashed black curves inside the colored regions represent two sample trajectories of the total cost as a function of time. The DSMs are constructed so that in each case component 1 has the lowest out-degree and component 7 has the highest out-degree. Below each distribution we plot the fraction of the total cost contributed by each of the 7 components at any given time (corresponding to the first simulation run). The components in *B* and *C* with the biggest contribution to the cost in the limit $t \to \infty$ are highlighted. The box at the bottom gives the value of $d_i^{\min}$ for each component of the design.

of the distribution vs. time. In every case the top envelope follows a straight line throughout most of the time range. The behavior of the bottom envelope is more complicated; in many cases, such as Fig. 5 *Left*, this bottom envelope also follows a straight line, but in others (for example, Fig. 5 *Center*) the bottom envelope changes slope over a large time range. A more precise statement can be made by following the contour corresponding to a given quantile through time. All quantiles eventually approach a line with slope $1/d^*$. However, the upper quantiles converge to this line quickly, whereas in some cases the lower quantiles do so much later. This slower convergence stems from the difference in improvement rates of different components. Whenever there is a dramatic improvement in the slowest-improving component (or components), there is a period where the next slowest-improving component (or components) becomes important. During this time the lower $d_i^{\min}$ value of the second component temporarily influences the rate of improvement. After a long time the slowest-improving component becomes more and more dominant, large updates become progressively more rare, and the slope becomes constant.

The model therefore suggests that properties of the design determine whether a technology's improvement will be steady or erratic. Homogeneous designs (with constant out-degree) are more likely to show an inexorable trend of steady improvement. Heterogeneous designs (with larger variability in out-degree) are more likely to improve in fits and starts. High variability among individual trajectories can be interpreted as indicating that historical contingency plays an important role. By this we mean that the particular choice of random numbers, rather than the overall trend, dominates the behavior. In this case progress appears to come about through a series of punctuated equilibria.

To summarize, in this section we have shown that the asymptotic magnitude of the fluctuations is determined by the number of bottlenecks; i.e., the number with $d_i^{\min} = d^*$. The fluctuations decrease as the number of bottlenecks increases. In the constant out-degree case all of the components are equivalent, and this number is just n. In the variable out-degree case, however, this number depends on the details of the DSM, which influence $d_i^{\min}$.

Testing the Model Predictions

Our model makes the testable prediction that the rate of improvement of a technology depends on the design complexity, which can be determined from a design structure matrix. The use of DSMs to analyze designs is widespread in the systems engineering and management science literature. Thus, one could potentially examine the DSMs of different technologies, compute their corresponding design complexities, and compare to the value of α based on the technology's history[‡]. Thus we are able to make a quantitative prediction about learning curves. This is in contrast to previous work, which did not make testable predictions about α[§]. This test is complicated by the fact that α also depends on γ, which describes the inherent difficulty of improving individual components, which in turn depends on the inherent difficulty of the innovation problem as well as the collective effectiveness of inventors in generating improvements. The exponent γ is problematic to measure independently. Nonetheless, one could examine a collection of different technologies and either assume that γ is constant or that the variations in γ average out. Subject to these caveats, the model then predicts that the design complexity of the DSM should be inversely proportional to the estimated α of the historical trajectory. A byproduct of such a study is that it would yield an estimate of γ in different technologies.

To compare the model predictions to real data one must relate the number of attempted improvements to something measurable. It is not straightforward to measure the effort that has gone into improving a technology, and to compare to real data one must use proxies. The most commonly used proxy is cumulative production, but other possibilities include cumulative investment, installed capacity, research and development expenditure, or even time. The best proxy for innovation effort is a subject of serious debate in the literature (34–38).

Possible Extensions to the Model

There are a variety of possible ways to extend the model to make it more realistic. For example, the model currently assumes that the design network described by the DSM is constant through time, but often improvements to a technology come about by modifying the design network. One can potentially extend the model by adding an evolutionary model for the generation of new DSMs.

The possibility that the design complexity d^* changes through time suggests another empirical prediction. According to our theory, when d^* changes, α changes as well. One can conceivably examine an historical sequence of design matrices, compute their values of d^*, and compare the predicted $\alpha \sim 1/d^*$ to the corresponding observed values of α in the corresponding periods in time. Our theory predicts that these should be positively correlated.

We have assumed a particular model of learning in which improvement attempts are made at random, with no regard to the history of previous improvements or knowledge of the technology. An intelligent designer should be able to do (as well or) better, drawing on his or her knowledge of science, engineering, and present and past designs. (We note that for particularly complex design problems, random search may be computationally the most efficient option.) The model we study here can be viewed as a worst case, which should be an indicator of the difficulty of design under any approach: A problem that is harder for a designer to solve under random search is also likely to be more difficult to solve with a more directed search.

Discussion

We have developed a model that both simplifies and generalizes the original model of Auerswald et al. (19), which predicts the improvement of cost as a function of the number of innovation attempts. Whereas we have formulated the model in terms of cost, one could equally well have used any performance measure of the technology that has the property of being additive across the components. Our analysis makes clear predictions about the trajectories of technology improvement. The mean behavior of the cost is described by a power law with exponent $\alpha = 1/(\gamma d^*)$, where d^* is the design complexity and γ describes the intrinsic difficulty of improving individual components. In the case of constant connectivity (out-degree) the design complexity is just the connectivity, but in general the complexity can depend on details of the design matrix, as spelled out in Eq. **8**. In addition, the range of variation in technological improvement trajectories depends on the number of bottlenecks. This number coincides with the total number of components n in the case of constant connectivity, but in general the number of bottlenecks is smaller, and depends on the detailed arrangement of the interactions in the design.

Many studies in the past have discussed effects that contribute to technological improvement, such as learning-by-doing, research and development, or capital investment. Our approach here is generic in the sense that it could apply to any of these mechanisms. As long as these mechanisms cause innovation attempts that can be modeled as a process of trial and error,

[‡]One problem that must be considered is that of resolution. As an approximation, a DSM can be constructed at the coarse level of entire systems (e.g., "electrical system," "fuel system") or it can be constructed more accurately at the microscopic level in terms of individual parts. In general these will give different design complexities.

[§]A possible exception is Huberman (20), who presents a theory in terms of an evolving graph, and gives a formula that predicts power law scaling as a function of "the number of new shortcuts." It is not clear, however, whether this could ever be measured.

any of them can potentially be described by the model we have developed.

Our analysis makes a unique contribution by connecting the literature on the historical analysis of performance curves to that on the engineering design properties of a technology. We make a prediction about how the features of a design influence its rate of improvement, focusing attention on the interactions of components as codified in the design structure matrix. Perhaps most importantly, we pose several falsifiable propositions. Our analysis illustrates how the same evolutionary process can display either historical contingency or steady change, depending on the design. Our theory suggests that it may be possible to influence the long-term rate of improvement of a technology by reducing the connectivity between the components. Such an understanding of how the design features of a technology affect its evolution could aid engineering design, as well as science and technology policy.

ACKNOWLEDGMENTS. We thank Yonathon Schwarzkopf and Aaron Clauset for helpful conversations and suggestions, and Sidharth Rupani and Daniel Whitney for useful correspondence. J.M., J.D.F., and J.E.T. gratefully acknowledge financial support from National Science Foundation (NSF) Grant SBE0738187; S.R. acknowledges support from NSF Grant DMR0535503.

1. Dutton JM, Thomas A (1984) Treating progress functions as a managerial opportunity. *Acad Manage Rev* 9:235–247.
2. Argote L, Epple D (1990) Learning curves in manufacturing. *Science* 247: 920–924.
3. McDonald A, Schrattenholzer L (2001) Learning rates for energy technologies. *Energ Policy* 29:255–261.
4. Thompson P (2010) Learning by doing. *Handbook of Economics of Technical Change* (Elsevier/North-Holland, New York).
5. Bryan WN, Harter N (1899) Studies in the telegraphic language: The acquisition of a heirarchy of habits. *Psychol Rev* 6:345–375.
6. Ebbinghaus H (1885) *Memory: A Contribution to Experimental Psychology* (Columbia University Press, New York).
7. Wright TP (1936) Factors affecting the cost of airplanes. *J Aeronaut Sci* 3:122–128.
8. Ostwald PF, Reisdorf JB (1979) Measurement of technology progress and capital cost for nuclear, coal-fired, and gas-fired power plants using the learning curve. *Eng Process Econ* 4:435–454.
9. Arrow KJ (1962) The economic implications of learning by doing. *Rev Econ Stud* 29:155–73.
10. McNerney J, Farmer JD, Trancik JE Historical costs of coal-fired electricity and implications for the future. *Energ Policy*, doi: 10.1016/j.enpol.2011.01.037.
11. Alberth S (2008) Forecasting technology costs via the experience curve—myth or magic? *Technol Forecast Soc* 75:952–983 September.
12. Maycock PD (2002) The world photovoltaic market. (PV Energy Systems, Williamsburg, VA) Technical Report.
13. Nemet GF (2006) Beyond the learning curve: Factors influencing cost reductions in photovoltaics. *Energ Policy* 34:3218–3232.
14. Strategies-Unlimited (2003) Photovoltaic five-year market forecast, 2002–2007. Technical Report PM-52.
15. Moore GE (1965) Cramming more components onto integrated circuits. *Electronics* 38:114–117.
16. Argote L (1999) *Organizational Learning: Creating, Retaining and Transferring Knowledge* (Kluwer, Boston).
17. Wene C (2000) *Experience Curves for Energy Technology Policy* (International Energy Agency, Paris).
18. Boston Consulting Group (1972) *Perspectives on Experience* (Boston Consulting Group, Boston, MA).
19. Auerswald P, Kauffman S, Lobo J, Shell K (2000) The production recipes approach to modeling technological innovation: An application to learning by doing. *J Econ Dyn Control* 24:389–450.
20. Huberman BA (2001) The dynamics of organizational learning. *Comput Math Organ Th* 7:145–153.
21. Muth JF (1986) Search theory and the manufacturing progress function. *Manage Sci* 32:948–962.
22. Bendler JT, Schlesinger MF (1991) Fractal clusters in the learning curve. *Physica A* 177:585–588.
23. Frenken K (2006) A fitness landscape approach to technological complexity, modularity, and vertical disintegration. *Struct Change Econ Dynam* 17:288–305.
24. Frenken K (2006) Technological innovation and complexity theory. *Econ Innovat New Tech* 15:137–155.
25. Silverberg G, Verspagen B (2005) A percolation model of innovation in complex technology spaces. *J Econ Dyn Control* 29:225–244.
26. Silverberg G, Verpagen B (2007) Self-organization of r&d search in complex technology spaces. *J Econ Interact Coord* 2:195–210.
27. Thurner S, Klimek P, Hanel R (2010) Schumpeterian economic dynamics as a quantifiable minimum model of evolution. *New J Phys* 12:075029.
28. Baldwin CY, Clark KB (2000) *Design Rules* (MIT Press, Cambridge, MA).
29. Smith RP, Eppinger SD, Whitney DE, Gebala DA (1994) A model-based method for organizing tasks in product development. *Res Eng Des* 6:1–13.
30. Steward DV (1981) The design structure-system: A method for managing the design of complex-systems. *IEEE T Eng Manage* 28:71–74.
31. Rivkin JW, Siggelkow N (2007) Patterned interactions in complex systems: Implications for exploration. *Manage Sci* 53:1068–1085.
32. Whitney DE, Dong Q, Judson J, Mascoli G (1999) Introducing knowledge-based engineering into an interconnected product development process. *Proceedings of the 1999 ASME Design Engineering Technical Conferences* (American Society for Mechanical Engineers, New York).
33. Chen S, Huang E (2007) A systematic approach for supply chain improvement using design structure matrix. *J Intell Manuf* 18:285–299 Apr.
34. Goddard C (1982) Debunking the learning curve. *IEEE T Compon Hybr* 5:328–335.
35. Koh H, Magee CL (2006) A functional approach for studying technological progress: Application to information technology. *Technol Forecast Soc* 73:1061–1083.
36. Koh H, Magee CL (2008) A functional approach for studying technological progress: Extension to energy technology. *Technol Forecast Soc* 75:735–758.
37. Nordhaus WD (2009) The perils of the learning model for modeling endogenous technological change. *SSRN eLibrary*, Available at http://ssrn.com/abstract=1322422.
38. Sinclair G, Klepper S, Cohen W (2000) What's experience got to do with it? Sources of cost reduction in a large specialty chemicals producer. *Manage Sci* 46:28–45.