Supporting Information

McNerney et al. 10.1073/pnas.1017298108

1. SI Introduction

Each curve in Fig. 1 was rescaled and shifted to show all curves on a single plot. The x- and y-coordinates of each series were transformed via

$$\log x \to a + b \log x \qquad \log y \to c + d \log y.$$

The values of the constants a, b, c, and d for each series are shown in Table S1.

2. SI The Model

2.1. Relation to the Production Recipe Model. The original production recipe (PR) model of Auerswald et al. (1) contained 6 parameters. We made the following simplifications of their model:

- Implementations per component $s \to \infty$. In the PR model, each component could be implemented in s possible ways. By setting $s = \infty$, we postulate that new implementations for a component can always be found. We consider this limit because: (*i*) many technologies perpetually benefit over time from scientific and engineering advances, and (*ii*) the main effect of finite s is that costs plateau once the global minimum or an inescapable local minimum is achieved [as Auerswald et al. already noted (1)]. Because our goal is to understand the dynamics of cost reduction, we set $s = \infty$.
- Length of production run $T \to \infty$. Because we are interested in the shape of the cost evolution, which is unaffected by T, we set no constraint on the number of recipe changes.
- Recipe changes per cost measurement $\tau \to 1$. The original model allowed for τ cost changes between observations. Alternatively, the inverse of τ is the frequency at which costs are observed, and has no effect on the underlying evolution. Setting $\tau = 1$ assumes a cost measurement is made for every recipe change.
- Search distance $\delta \to 1$. In the PR model, up to δ components are allowed to change each time a new recipe is tried. For simplicity, we consider modifying components to change one at a time.

Thus we retain only two parameters of the PR model: n, the number of components, and d, the number of dependencies per component.

2.2. Simulation Algorithm. The simplest way to simulate the model is as follows:

Simple_sim
for *each time step* **do**
  pick a component i
  generate new component cost c_j' for each $j \in \mathscr{A}_i$
  if $a_i' < a$ **then**
    $c_j \leftarrow c_j'$ for each $j \in \mathscr{A}_i$
  end
end

Although this algorithm is straightforward to implement, it is extremely inefficient. When improvements become rare, a program using the above algorithm will spend most iterations rejecting new costs.

A substantially more efficient procedure would be to only simulate successful steps, where cost variables are updated. This idea leads to the following algorithm:

Fast_sim
for *each cost update* **do**
  pick a component i
  generate new component cost c_j' for each $j \in \mathscr{A}_i$
    such that $a_i' < a_i$
  generate the time-taken to achieve the update
end

The challenge is to use the correct distributions for the random variables in each of the three steps of the fast algorithm.

Before describing the three distributions needed, recall that

$$p(+|i) = \frac{(na_i)^{d_i}}{d_i!}$$

is the probability of accepting a change, given that i has been picked. We can use these probabilities to write down the correct distributions for each of the three steps of the fast algorithm.

When $a_i > 1/n$, the target region is not simply the volume under a simplex. This problem is circumvented easily, by running the simulation using the straightforward algorithm initially, until all the target volumes have shrunk sufficiently. This happens when

$$a_i \leq \frac{1}{n}$$

is satisfied for all i. In practice, this happens very quickly, so little time is lost by using the straightforward algorithm initially.

Step 1: Pick a component i. We should pick i with frequency equal to the probability that i is the component responsible for the next update. This probability, which we will call $q(i)$, is proportional to $p(+|i)$, the probability that i yields an update once picked. Because the $q(i)$ should sum to 1 (some component is definitely responsible for the next update), to compute $q(i)$, we should normalize by the sum over $p(+|i)$:

$$q(i) \equiv \frac{p(+|i)}{\sum_{j=1}^{n} p(+|j)}.$$

Step 2: Generate new component cost c_j' for each $j \in \mathscr{A}_i$ such that $a_i' < a_i$. This step is composed of two substeps. First we generate $a_i' < a_i$, the new total cost of the members of $\mathscr{A}_i$. Second we partition a_i' among the members of $\mathscr{A}_i$.

To obtain the appropriate distribution from which to draw a_i', let the members of $\mathscr{A}_i$ be indexed 1 through d_i. The probability that the members of $\mathscr{A}_i$ have total cost a_i' is

$$p(a_i'|+,i) = \frac{\int_0^{a_i'} dc_1' \int_0^{a_i'-c_1'} dc_2' \ldots \int_0^{a_i'-\sum_{k=1}^{d_i} c_k'} dc_{d_i}' p(c_1') p(c_2') \ldots p(c_{d_i}') \delta(a_i' - \sum_{k=1}^{d_i} c_k')}{p(+|i)}.$$

The numerator computes the probability of drawing costs $(c_1, \cdots, c_{d_i})$ such that a_i' is their sum. Because we integrated only over the range corresponding to $a_i' < a_i$ [i.e., a subset of the range of $(c_1, \cdots, c_{d_i})$], we must normalize by the total probability of that subset, $p(+|j)$. Evaluating the integral, we have

$$p(a_i'|+,i) = \binom{d_i}{a_i'^{d_i}} a_i'^{d_i-1} \qquad a_i' \in [0,a_i].$$

A random variable with this distribution can be easily generated, using standard numerical techniques. See, for example, ref. 2.

Step 3: Generate the time-taken to achieve the update. The probability of achieving an update via component i is $p(i)p(+|i) = \frac{1}{n}p(+|i)$. The probability p_{any} of achieving an update via any component is the sum of $\frac{1}{n}p(+|i)$ over all components:

$$p_{\text{any}} = \frac{1}{n}\sum_{i=1}^{n} p(+|i).$$

The probability that T steps are taken to achieve an update is simply the geometric distribution with parameter p_{any},

$$p(T) = (1-p_{\text{any}})^{T-1} p_{\text{any}};$$

i.e., the probability that no update occurs $T-1$ times and then occurs on the Tth step.

3. SI Independent Components

3.1 Background. ***3.1.1. Useful Definitions and Nomenclature.*** We define the outset of component i, $\mathscr{A}_i$, as the set of components that depend on component i. Because the elements of the design structure matrix are given by $D_{ji} = 1$ if j depends on i and $D_{ji} = 0$ otherwise, $\mathscr{A}_i$ formally is

$$\mathscr{A}_i \equiv \{j: D_{ji} = 1\}. \qquad \textbf{[S1]}$$

It is also useful to define the inset of components $\mathscr{Z}_i$ that i depends on as

$$\mathscr{Z}_i \equiv \{j: D_{ij} = 1\}. \qquad \textbf{[S2]}$$

Viewing the relationships between components as a graph, the out-degree d_i of component i is the number of components in $\mathscr{A}_i$,

$$d_i = \sum_j D_{ji}, \qquad \textbf{[S3]}$$

whereas the in-degree of i is the number of components in $\mathscr{Z}_i$. For a given design structure matrix D, the outset $\mathscr{A}_i$ corresponds to the nonzero rows of column i and the inset $\mathscr{Z}_i$ corresponds to the nonzero columns of row i.

Let c_j be the cost of an individual component j and let a_i be the cost of cluster $\mathscr{A}_i$:

$$a_i \equiv \sum_{j\in\mathscr{A}_i} c_j = \sum_{j=1}^{n} D_{ji}c_j. \qquad \textbf{[S4]}$$

Component costs are updated by the following rule: A component i is picked at random, and new costs c_j' for $j \in \mathscr{A}_i$ are generated. If the resulting cluster cost satisfies $a_i' < a_i$, all components in $\mathscr{A}_i$ are updated with their new values.

This rule implies that a component i can be updated in two ways:

1. Component i is chosen and the components in $\mathscr{A}_i$ updated.
2. Component $j \in \mathscr{Z}_i$ $(j \neq i)$ is chosen and the components in $\mathscr{A}_j$ updated.

When the out-degree is different for each node, components may improve at different rates. An important characterization is the minimum out-degree of the inset of i

$$d_i^{\min} = \min_{j\in\mathscr{Z}_i} d_j.$$

For reasons that will become apparent, we term the inverse of this quantity the improvement rate of component i. We also define the design complexity as

$$d^* = \max_i \{d_i^{\min}\}.$$

3.1.2. Statistical Equivalence, Independence, and Bottlenecks. For the purpose of this document, we use the notation $f(x) \sim g(x)$ to mean there exists a constant $C \neq 0$ such that in the limit $x \to \infty$, $f(x)/g(x) \to C$. We will say that f and g are asymptotically statistically equivalent.

Components i and j are asymptotically statistically equivalent if the expectation value of their costs scale with time in the same way; i.e.,

$$E(c_i(t)) \sim E(c_j(t)).$$

The average is performed over different realizations of random trials of new component costs. Similarly, clusters i and j are asymptotically statistically equivalent if $E(a_i(t)) \sim E(a_j(t))$.

Components can be grouped into equivalence classes according to their improvement rate $d_i^{\min}$. We will argue that clusters that have the same $d_i^{\min}$ are statistically equivalent. If the underlying network of components has diverse values of $d_i^{\min}$, we call any component with $d_i^{\min} = d^*$ a bottleneck. As we will show, bottleneck components improve the slowest, so that they asymptotically dominate the cost of the entire technology.

Another important property is that of independence, in which individual clusters or components do not depend on each other. Two components are independent if $p(c_i,c_j) = p(c_i)p(c_j)$; here $p(c_i,c_j)$ denotes the joint probability that components i and j have respective costs c_i and c_j, whereas $p(c_i)$ is the probability that component i has cost c_i.

3.2. Average Cost Versus Time. Let the total cost of a technology be $\kappa(t) = \sum_{i=1}^{n} c_i$. Its expectation value at time t is

$$E(\kappa(t)) \equiv \int \kappa p(\kappa,t)d\kappa = \sum_{i=1}^{n}\int c_i p(c_1,\ldots,c_n,t)dc_1\ldots dc_n$$
$$= \sum_{i=1}^{n} E(c_i(t)), \qquad \textbf{[S5]}$$

where $E(c_i(t))$ is the expected cost of component i after t innovation attempts.

Whereas we are unable to compute the distribution $p(c_1,\ldots,c_n,t)$ exactly, we can find effective approximations under certain circumstances, and we can compute the asymptotic scaling. We use extreme value theory for this purpose in sections 4.1 and 4.2, whereas in 4.3 and 4.4 we use a continuous-time approximation and cast the problem in terms of a differential equation for the average cost.

We now derive the cost evolution for $d = 1$ by extreme value theory. Let t_i be the number of times component i is chosen after t time steps, where $\sum_i t_i = t$. The expected cost of component i can be written

$$E(c_i(t)) = \sum_{t_i=0}^{t_i=t} P(t_i|t) E(c_i|t_i), \qquad \text{[S6]}$$

where $P(t_i|t)$ is the probability that component i is chosen t_i times given t innovation steps, and $E(c_i|t_i)$ is the expected cost of component i given that it was chosen t_i times. Because each component is chosen with equal probability $1/n$, the conditional probability $P(t_i|t)$ is the binomial distribution

$$P(t_i|t) = \binom{t}{t_i} \left(\frac{1}{n}\right)^{t_i} \left(1-\frac{1}{n}\right)^{t-t_i}. \qquad \text{[S7]}$$

The expected cost $E(c_i|t_i)$, in contrast, requires stronger assumptions to be computable.

When the degree $d = 1$ for every node, each component is isolated and independent. This means that at any time t, $p(c_1,\ldots,c_n,t) = p(c_1,t)\ldots p(c_n,t)$, and we can compute $E(c_i|t_i)$ as follows:

$$E(c_i|t_i) = \int c_i p(c_1,\ldots,c_n,t) dc_1 \ldots dc_n = \int c_i p(c_i,t) dc_i. \qquad \text{[S8]}$$

The strategy of the calculation is to compute $p(c_i,t)$ for a single representative component with cost $c_i = c$, and then compute the final result using Eqs. **S5**–**S7**.

Under the assumptions of the model, each trial cost value is independently drawn from a continuous distribution $f(c)$, which is fixed in time. From extreme value theory (3), the probability that the minimum cost exceeds c after t_i such trials is $[1-F(c)]^{t_i}$, where $F(c) = \int_0^c f(c)dc$ is the cumulative distribution. The probability density $p(c,t_i)$ of the cost of a component after t_i improvement attempts is therefore

$$p(c,t_i) = -\frac{d}{dc}[1-F(c)]^{t_i} = t_i f(c)[1-F(c)]^{t_i-1}. \qquad \text{[S9]}$$

For the case where $f(c) = \gamma n^\gamma c^{\gamma-1}$ on $[0,1/n]$, then $F(c) = (nc)^\gamma$, and the expectation of the cost of component i after t_i improvement attempts is

$$E(c_i|t_i) = \int_0^{1/n} c_i p(c_i,t_i) dc_i = \gamma t_i \int_0^{1/n} (nc_i)^\gamma [1-(nc_i)^\gamma]^{t_i-1} dc_i$$
$$= \frac{t_i}{n} \int_0^1 x^{1/\gamma} (1-x)^{t_i-1} dx = \frac{t_i}{n} B\left(1+\frac{1}{\gamma}, t_i\right). \qquad \text{[S10]}$$

In the last step, we substitute $x = (nc_i)^\gamma$ to express the integral in terms of the beta function $B(a,b)$ (4). When $\gamma = 1$ Eq. **10** reduces to $E(c_i|t_i) = \frac{1}{n}\frac{1}{t_i+1}$, similar to the original result of Muth (5) (who solved the even simpler case of a technology with one component). To highlight the asymptotic dependence on t_i, we use the large-b approximation $B(a,b) \sim \Gamma(a) b^{-a}$, where $\Gamma(a)$ is Euler's gamma function, to give

$$E(c_i|t_i) \sim \frac{t_i^{-1/\gamma}}{n} \Gamma\left(1+\frac{1}{\gamma}\right). \qquad \text{[S11]}$$

We now average over t_i. Substituting Eq. **S7** and **[S11]** into Eq. **S6** gives, in the long-time limit

$$E(c_i(t)) \sim \frac{1}{n}\Gamma\left(1+\frac{1}{\gamma}\right) \sum_{t_i=0}^{t} t_i^{-1/\gamma} \binom{t}{t_i} \left(\frac{1}{n}\right)^{t_i} \left(1-\frac{1}{n}\right)^{t-t_i}$$
$$\equiv \frac{1}{n}\Gamma\left(1+\frac{1}{\gamma}\right) E[t_i^{-1/\gamma}],$$

where $E[t_i^{-1/\gamma}]$ is the noninteger moment of the binomial distribution. This quantity is known to have the asymptotic expansion (6)

$$E[t_i^{-1/\gamma}] = \left(\frac{t}{n}\right)^{-1/\gamma} \left[1 + \frac{\frac{1}{\gamma}\left(\frac{1}{\gamma}+1\right)\left(1-\frac{1}{n}\right)}{2t/n} + \cdots\right]. \qquad \text{[S12]}$$

The leading term corresponds to merely replacing t_i by its average value t/n so that

$$E(c_i) \sim \frac{1}{n}\Gamma\left(1+\frac{1}{\gamma}\right)\left(\frac{t}{n}\right)^{-1/\gamma} [1+O(n/t)], \qquad \text{[S13]}$$

where $O(n/t)$ indicates that the answer is accurate to order n/t. This agrees with the intuition that in the limit when t is large the binomial distribution becomes sharply peaked around its mean value. Summing over over all n components in Eq. **S5** gives $E(\kappa(t)) = nE(c_i(t))$.

4. SI Interacting Components, Fixed Out-Degree

4.1. Cost Evolution for Constant Out-Degree $d > 1$ by Extreme Value Theory. In this section we derive an asymptotic approximation for the cost when the out-degree d of each component is the same, but greater than one, so that components are no longer independent. The strategy of the calculation is to express the average cost as a sum over clusters rather than components, and to treat each cluster as independent.

We first express the total cost as a sum over clusters. Let the design structure matrix (DSM) D be invertible. (We can show numerically that our approximation also works when D is not invertible). Then we can write the component costs in terms of the cluster costs as $c_i = \sum_j D_{ji}^{-1} a_j$ and the total cost κ as

$$\kappa = \sum_{j=1}^n c_j = \sum_{ij} D_{ij}^{-1} a_i = \sum_i^n k_i a_i, \qquad \text{[S14]}$$

where $k_i \equiv \sum_j D_{ij}^{-1}$. For example, for a fully connected network, $k_i = 1/n\ \forall i$, whereas for a network where each component is isolated $k_i = 1\ \forall i$.

We can now express Eq. **5** in terms of cluster costs rather than component costs. Let $\tilde{p}(a_1,\ldots,a_n,t)$ be the joint probability density function for the clusters. The expected cost of the technology can then be written

$$E(\kappa(t)) = \int \left(\sum_{i=1}^n c_i\right) p(c_1,\ldots,c_n,t) dc_1 \ldots dc_n$$
$$= \int \left(\sum_{i=1}^n k_i a_i\right) \tilde{p}(a_1,\ldots,a_n,t) da_1 \ldots da_n = \sum_{i=1}^n k_i E(a_i(t)). \qquad \text{[S15]}$$

We now make the approximation that the clusters evolve independently, so that $\tilde{p}(a_1,\ldots,a_n,t) = \tilde{p}(a_1,t)\ldots p(a_n,t)$. This implies that

$$E(\kappa(t)) = \sum_{i=1}^n k_i E(a_i(t)) = \sum_{i=1}^n k_i \int a_i \tilde{p}(a_i,t) da_i. \qquad \text{[S16]}$$

Because the out-degree is constant we can also make the approximation that all clusters behave similarly, so that we can write the cost a_i of any particular cluster as that of a generic cluster; i.e., $a_i = a$. This implies

$$E(\kappa(t)) \approx n\bar{k}E(a(t)). \quad \textbf{[S17]}$$

In the constant out-degree case it is straightforward to show that

$$\bar{k} \equiv \frac{1}{n}\sum_i k_i = \frac{1}{d}. \quad \textbf{[S18]}$$

The problem is now reduced to computing $\tilde{p}(a,t)$ and performing the integral. Following analogous steps as those for the $d = 1$ case of isolated components, the distribution $\tilde{p}(a,t_i)$ is

$$\tilde{p}(a,t_i) = t_i \tilde{f}(a)[1-\tilde{F}(a)]^{t_i-1}, \quad \textbf{[S19]}$$

where $\tilde{F}$ is the cumulative parent distribution of cluster costs, $\tilde{f}$ is the corresponding probability density, and t_i is the number of times cluster i is chosen.

We now express the parent cost distribution for clusters, $\tilde{f}(a)$, in terms of those of individual components. As before, we use $f(c) = \gamma n^\gamma c^{\gamma-1}$ (so that $F(c) = (nc)^\gamma$) for small c, so that the cluster cost distribution $\tilde{f}$ is

$$\tilde{f}(a) = \int f(c_1)\cdots f(c_d)\delta(c_1+\cdots+c_d-a)dc_1\cdots dc_d.$$

For concreteness, first consider the case $d = 3$, after which the general case follows easily. The distribution $\tilde{f}(a)$ is

$$\begin{aligned}
\tilde{f}(a) &= (\gamma n^\gamma)^3 \int_0^a dc_1 \int_0^{a-c_1} dc_2 \int_0^{a-c_1-c_2} dc_3 c_1^{\gamma-1}c_2^{\gamma-1}c_3^{\gamma-1}\delta(c_1+c_2 \\
&\quad +c_3-a) \\
&= (\gamma n^\gamma)^3 \int_0^a dc_1 c_1^{\gamma-1} \int_0^{a-c_1} dc_2 c_2^{\gamma-1}[(a-c_1)-c_2]^{\gamma-1} \\
&= (\gamma n^\gamma)^3 B(\gamma,\gamma)\int_0^a dc_1 c_1^{\gamma-1}(a-c_1)^{2\gamma-1} \\
&= a^{3\gamma-1}(\gamma n^\gamma)^3 B(\gamma,\gamma)B(\gamma,2\gamma).
\end{aligned}$$

The corresponding result for general d is

$$\tilde{f}(a) = a^{\gamma d-1}(\gamma n^\gamma)^d \prod_{j=1}^{d-1} B(\gamma,j\gamma) = a^{\gamma d-1}(\gamma n^\gamma)^d \frac{\Gamma(\gamma)^d}{\Gamma(\gamma d)}. \quad \textbf{[S20]}$$

Using the latter form, we obtain for $\tilde{F}$:

$$\tilde{F}(a) \sim a^{\gamma d}\frac{(\gamma n^\gamma)^d}{\gamma d}\frac{\Gamma(\gamma)^d}{\Gamma(\gamma d)} \equiv a^{\gamma d}H(n,d,\gamma). \quad \textbf{[S21]}$$

Substituting Eq. **S20** and **[S21]**into Eq. **S19** gives

$$\tilde{p}(a,t_i) = \gamma d a^{\gamma d-1}t_i H[1-a^{\gamma d}H]^{t_i-1} \sim \gamma d a^{\gamma d-1}t_i H\exp[-a^{\gamma d}Ht_i]$$
$$t\to\infty, \quad a\to 0. \quad \textbf{[S22]}$$

We can now compute $E(c_i(t))$ as before using Eq. **S6**. The average over $P(t_i|t)$ is no longer tractable in closed form, but we assume that $t_i \approx (d/n)t$. As before, each component is chosen $1/n$ of the time; the factor d appears because components belong to d clusters on average, and are therefore updated d times as often.

$$\tilde{p}(a,t) \sim \gamma d a^{\gamma d-1}t/nH\exp[-a^{\gamma d}dHt/n] \quad t\to\infty, \quad a\to 0. \quad \textbf{[S23]}$$

This expression has the form of a Weibull distribution:

$$p(x) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1}e^{-(x/\lambda)^k},$$

with $x = a$ and parameters $k = \gamma d$ and $\lambda = (dHt/n)^{-1/(\gamma d)}$. The average value $E(x)$ for a Weibull distribution in terms of these standard parameters is $\lambda\Gamma(1+\frac{1}{k})$, which implies that the average cluster cost in the limit as $t\to\infty$ is

$$E(a(t)) = \Gamma\left(1+\frac{1}{\gamma d}\right)(dHt/n)^{-1/(\gamma d)}. \quad \textbf{[S24]}$$

From **[S17]** the total cost of the technology is

$$E(\kappa(t)) = \frac{n}{d}\Gamma\left(1+\frac{1}{\gamma d}\right)(dHt/n)^{-1/(\gamma d)}. \quad \textbf{[S25]}$$

Note that this reduces to **[S13]** when $d = 1$. This functional form matches our numerical simulations. It gives the correct asymptotic scaling exponent, and the constant is accurate within a factor of 2.

4.2. Alternative Derivation of Average Cost Evolution in the Constant Out-Degree Case. We now present an alternative derivation for the case of constant $d > 1$ by writing and solving a differential equation for the average cluster cost. This second derivation requires stronger assumptions than the first, but we include this latter approach for three reasons: (*i*) Simulations show it gives better agreement for small t. (*ii*) It illustrates the cause of the power law behavior more clearly than the first derivation. (*iii*) Most importantly, it provides intuition and justification for the solution of the more general case where the out-degree is not constant in the next section.

We use the same decomposition as in the previous section, and use **[S17]** and Eq. **S18** to express the total technology cost as $E(\kappa(t)) \approx nE(a(t))/d$, thereby reducing the computation of the average cost of the whole technology in terms of that of a single cluster. As in our previous discussion, we are thus assuming that the clusters are independent and roughly identical.

We now compute the average rate at which clusters improve, and then solve the resulting differential equation. For brevity, let $a' = a(t+1)$, and $a = a(t)$. The expected change in the cost of a cluster, $\Delta a = a' - a$, given that its current cost is a, can be written as

$$\begin{aligned}
E[\Delta a|a] &= \int_0^a (a'-a)\tilde{f}(a')da' = \int_0^a a'\frac{d\tilde{F}}{da'}da' - a\tilde{F}(a) \\
&= \int_0^a \left[\frac{d}{da'}(a'\tilde{F}(a')) - \tilde{F}(a')\right]da' - a\tilde{F}(a) \\
&= [a'\tilde{F}(a')]_0^a - \int_0^a \tilde{F}(a')da' - a\tilde{F}(a) = -\int_0^a \tilde{F}(a')da'.
\end{aligned} \quad \textbf{[S26]}$$

Using **[S21]**, we obtain

$$E[\Delta a|a] \sim -\frac{H(n,d,\gamma)}{\gamma d+1}a^{\gamma d+1}. \quad \textbf{[S27]}$$

This calculation assumes that $a = a(t)$ is given. However, $a(t)$ is itself a random variable that depends on the previous $t-1$ steps. Therefore, one must also average over these previous steps to obtain the unconditional rate of change $E(\Delta a)$:

$$E(\Delta a) \sim -\frac{H(n,d,\gamma)}{\gamma d+1}E(a^{\gamma d+1}). \quad \textbf{[S28]}$$

We now make three assumptions. First, as in the previous section (and motivated by the calculation in section 4.1), we assume that each component is chosen t/n times, and we neglect indirect updates of individual components when other clusters are chosen. Second, we take the continuous-time limit by noting that the average change in a in a unit time step is $E(\Delta a)$, so that for long times (when $E(\Delta a)$ is small) we can write $E(\Delta a) \approx dE(a)/dt_i$. Third, we assume that

$$E(a^p) \sim E(a)^p. \qquad \textbf{[S29]}$$

The first two assumptions are clearly excellent approximations in the long-time limit, whereas **[S29]** implies that the pth power of the fluctuation about the average has the same time dependence as the average raised to the pth power. For sufficiently small p this assumption is numerically supported, as we illustrate in Fig. S1. For large p this assumption may break down, however, even though it is valid for all the examples that we have studied. (As p increases, extreme fluctuations of a will dominate the average $E(a^p)$. As long as the distribution of costs decays faster than a power law for large costs, the time dependence of $E(a^p)$ must be the same as that of $[E(a)]^p$ as $t \to \infty$; it is worth noting that the calculation of $E(a^p)$ in the previous section does not assume an upper bound on p, suggesting that $E(a^p) \sim E(a)^p$ is always valid).

Combining **[S28]** and **[S29]**, substituting $E(a) \approx E(\kappa)d/n$ from **[S17]**, and substituting $t_i = (d/n)t$ as before, gives the differential equation

$$\frac{dE(\kappa)}{dt_i} \sim -\left(\frac{d}{n}\right)^{\gamma d+1} \frac{H(n,d,\gamma)}{\gamma d+1} E(\kappa)^{\gamma d+1}, \qquad \textbf{[S30]}$$

whose solution is

$$E(\kappa(t)) = \left[\left(\frac{d}{n}\right)^{\gamma d+1} \frac{\gamma d}{1+\gamma d} H(n,d,\gamma) t + 1\right]^{-1/(\gamma d)}, \qquad \textbf{[S31]}$$

where we imposed the model's initial condition $E(\kappa(0)) = 1$.

At large times, this becomes

$$E(\kappa(t)) \sim \frac{n}{d}\left(\frac{\gamma d}{\gamma d+1}\right)^{-1/(\gamma d)} \left(\frac{dHt}{n}\right)^{-1/(\gamma d)}. \qquad \textbf{[S32]}$$

This result has the same asymptotic time scaling as Eq. **S25** of the previous section, though with a different prefactor. A comparison of these prefactors can be seen in Fig. S2. The prefactors never differ by more than a factor of 2, and quickly approach 1 for larger values of γd.

5. SI Interacting Components, Variable Out-Degree

When components have variable out-degree, different components can exhibit different scaling dependences of cost versus time and the assumptions of the previous sections break down. Our strategy for finding a solution in this case is to identify the dominant components (those whose costs decrease the slowest) and neglect everything else (for reasons explained below). Unlike earlier calculations, we can only perform this calculation for $\gamma = 1$. (When $\gamma \neq 1$ the integrals involved lack closed-form solutions).

To calculate the evolution of the cost, we study $\Delta c_i = c_i' - c_i$, the change in c_i at a given time step. To compute the expected value $E(\Delta c_i)$ it is useful to define the following three quantities:

- $p(+|j)$ is the probability that component j is improved, given that a cluster that contains it is chosen.
- $p(c_i'|+j)$. Given that component j is chosen and that it improves, the probability that component i has cost c_i' is $p(c_i'|+j)$.
- $E(\Delta c_i|+j)$. Given that component j is chosen and it improves, the expected change in its cost is given by

$$E(\Delta c_i|+j) = \int (c_i' - c_i) p(c_i'|+j) dc_i'.$$

As before, we assume that in the long-time limit each component is chosen t/n times. The average change of c_i can then be written as

$$E(\Delta c_i) = \frac{1}{n} \sum_{j \in \mathcal{I}_i} E(p(+|j) E(\Delta c_i|+j)). \qquad \textbf{[S33]}$$

Note that the sum is over all the components j in the inset of i; i.e., all components that could potentially cause i to be updated. Thus the prefactor $1/n$ is the probability that component j is picked. Notice the existence of two averages. First, there is an average over the single step that takes the cost from c to c' to compute $E(\Delta c_i|+j)$ given that the system is in the state $\vec{c} = (c_1,\ldots,c_n)$. Second, there is an average over all states $\vec{c}$.

As already mentioned, we assume the parent distribution of individual component costs F is the uniform distribution on $[0,1/n]$. Let the members of $\mathcal{A}_j$ be indexed from 1 through d_j. We now compute each of the components on the right side of Eq. **S33**:

The first term is

$$\begin{aligned} p(+|j) &= \int_{<} p(c_1') p(c_2') \ldots p(c_{d_j}') dc_1' dc_2' \ldots dc_{d_j}' \\ &= \int_0^{a_j} dc_1' \int_0^{a_j - c_1'} dc_2' \ldots \int_0^{a_j - \sum_{k=1}^{d_j} c_k'} dc_{d_j}' p(c_1') p(c_2') \ldots p(c_{d_j}') \\ &= \frac{(na_j)^{d_j}}{d_j!}, \end{aligned} \qquad \textbf{[S34]}$$

where the subscript $<$ denotes integration over the range $a_i' < a_i \ \forall i$.

For the second term we must first compute the probability

$$p(c_i'|+j) = \frac{\int_0^{a_j} dc_1 \int_0^{a_j - c_1} dc_2 \ldots \int_0^{a_j - \sum_{k=1}^{n-1} c_k} dc_n p(c_1) p(c_2) \ldots p(c_n) \delta(c_i - c_i')}{p(+|j)}.$$

The numerator gives the probability of sampling the costs $(c_1,\cdots,c_n)$ such that c_i takes on the particular value c_i'. Because we integrate only over the range corresponding to $a_i' < a_i$ [i.e., a subset of the range of $(c_1,\cdots,c_n)$], we must normalize by the total probability of that subset, $p(+|j)$. Evaluating the integral gives

$$p(c_i'|+j) = \frac{n^{d_j}(a_j - c_1)^{d_j - 1}/(d_j - 1)!}{(na_j)^{d_j}/d_j!} = d_j \frac{(a_j - c_i')^{d_j - 1}}{a_j^{d_j}}. \qquad \textbf{[S35]}$$

The expected conditional change in the cost of component i is therefore

$$E(\Delta c_i|+j) = \int_0^{a_j} (c_i' - c_i) d_j \frac{(a_j - c_i')^{d_j - 1}}{a_j^{d_j}} dc_i' = \frac{a_j}{d_j + 1} - c_i. \qquad \textbf{[S36]}$$

Using the results of Eqs. **S34** and **S36**, Eq. **S33** becomes

$$E(\Delta c_i) = -\frac{1}{n} \sum_{j \in \mathcal{I}_i} \frac{n^{d_j}}{(d_j + 1)!} [(d_j + 1) E(a_j^{d_j} c_i) - E(a_j^{d_j + 1})]. \qquad \textbf{[S37]}$$

Because the change in the cost of a single component involves sums of terms raised to powers, an analytic solution of [**S37**] is not possible. Nevertheless, it is possible to derive a good approximation of the total cost in the long-time limit by identifying the components whose costs dominate, and neglecting everything else. (As we will show numerically, this is a reasonable approximation because the dominant component costs rapidly become orders of magnitude larger than all others.)

We first determine those components whose costs asymptotically dominate, which we call the bottleneck components. We make this determination as follows: during the early part of the evolution all components have the same cost $c_i = g$. Assuming they remain the same, Eq. **S37** becomes

$$E(\Delta c_i) = -\frac{1}{n}\sum_{j\in\mathcal{I}_i}\frac{(nd_j)^{d_j}}{(d_j+1)!}E(g^{d_j+1}) = -\sum_{j\in\mathcal{I}_i}K_jE(g^{d_j+1}). \qquad \textbf{[S38]}$$

When g is small the prefactor K_j is unimportant—only the exponent of $E(g^{d_j+1})$ matters. Because $g < 1$, the largest term in the sum is given by the index j with the smallest exponent $d_j + 1$. Thus, the largest contribution to the change of the cost of component i is given by the member of i's inset with the smallest outdegree d_j; i.e.,

$$d_i^{\min} = \min_{j\in\mathcal{I}_i} d_j. \qquad \textbf{[S39]}$$

The key role of $d_i^{\min}$ in determining the asymptotic scaling is verified numerically in Fig. S3. We use the DSM from Fig. 5*B* of the main article. The simulation result for $E(c_i(t))$ is plotted for each component i. We see that all components that are characterized by the same value of $d_i^{\min}$ have the power-law decay with exponent $-1/d_i^{\min}$. In simulations with many different DSMs we invariably observe that $d_i^{\min}$ determines the asymptotic scaling of each component. (The top right corner of Fig. S2 shows the time evolution of two components in a single realization: a bottleneck (component 7) and a nonbottleneck (component 1) whose cost decreases more quickly. Note the prominent upticks in c_1 at three distinct times. These occur because component 1 is in cluster 7, so that when component 7 is successfully updated, the value of c_1 is changed to a new value that is of the order of c_7. These updates of component 7 occur rarely, as seen by the long flat steps in $c_7(t)$. After an uptick, however, component 1 is updated many times in succession and its cost rapidly returns close to its previous value.) The dominant components will be those that decrease the slowest; i.e., those with the largest value of $d_i^{\min}$. This maximal value is

$$d^* = \max d_i^{\min}. \qquad \textbf{[S40]}$$

To derive the asymptotic evolution of the cost for a general DSM, suppose that there are m components that are bottlenecks. We define the set of these bottleneck components as

$$\mathcal{B} = \{i | d_i^{\min} = d^*\}. \qquad \textbf{[S41]}$$

Consider the change in cost for bottleneck components. We make the assumption, that is justified numerically, that all bottlenecks have a common cost g that is much larger than that of all other components at long times. That is

$$c_j \approx \begin{cases} g & \text{if } j\in B \\ 0 & \text{otherwise.} \end{cases} \qquad \textbf{[S42]}$$

With this approximation Eq. **S37** becomes

$$E(\Delta g) \approx -\sum_{j\in\mathcal{B}}K_jE(g^{d_j+1}) \approx -mK^*E(g^{d^*+1}), \qquad \textbf{[S43]}$$

where we make use of the fact that there are m bottlenecks, each of which has $d_j = d^*$, and K^* is the value of K_j with $d_j = d^*$. Using approximation [**S29**] that a single scale accounts for all moments of the cost, we can write

$$E(\Delta g) = \frac{dE(g)}{dt} \approx -mK^*[E(g)]^{d^*+1}. \qquad \textbf{[S44]}$$

Solving, substituting for K^*, and recalling that because there are m bottlenecks, $E(\kappa(t)) \approx mE(g(t))$, the expected total cost of the technology becomes

$$E(\kappa(t)) \sim m\left(d^*\frac{m}{n}\frac{(nd^*)^{d^*}}{(d^*+1)!}t\right)^{-1/d^*}. \qquad \textbf{[S45]}$$

Fig. S4 compares the observed asymptotic slope of $E(\kappa(t))$ to the theoretically predicted slope $-1/d^*$ for DSMs with d^* values 1,2,…,9. For each value of d^*, 10 random DSMs were generated. For each DSM the cost reduction process was simulated 300 times, from which the value $E(\kappa(t))$ was computed. The asymptotic slope of this expected value was then measured by a least-squares fit, whereas the theoretical slope $-1/d^*$ was found directly from the DSM. The data lie close to the dashed diagonal line, in agreement the theoretical prediction. (The small deviation for the largest values of d^* arises because $E(\kappa(t))$ approaches its asymptotic behavior extremely slowly, thus giving a too-large value of the observed slope of $E(\kappa(t))$ in a finite-time simulation.)

1. 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.
2. Press W, Teukolsky S, Vetterling W, Flannery B (1999) *Numerical Recipes in C* (Cambridge University Press, Cambridge, UK).
3. Galambos J (1987) *The Asymptotic Theory of Extreme Order Statistics* (R.E. Krieger, Malabar, FL).
4. Abramowitz M, Stegun IE (1964) *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables* (Dover, New York).
5. Muth JF (1986) Search theory and the manufacturing progress function. *Manage Sci* 32:948–962.
6. Z ˇnidaric ˇ M (2009) Asymptotic expansion for inverse moments of binomial and Poisson distributions. *Open Stat Probab J* 1:7–10.

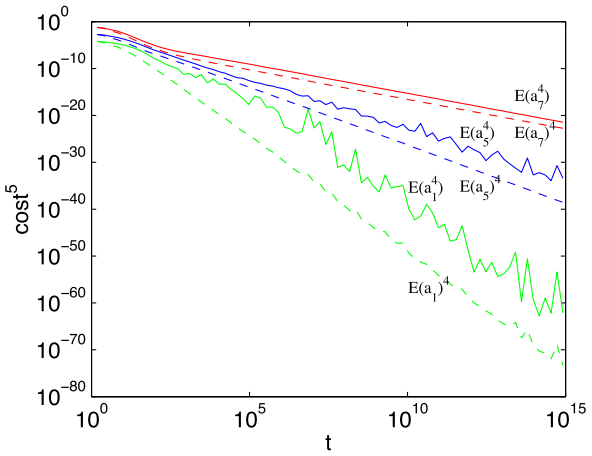


Fig. S1. Comparison of $E(a^p)$ and $E(a)^p$ for $p = 4$ on a double logarithmic scale for 10^4 realizations of the cost reduction process for the DSM in Fig. 5*B* of the main article for clusters $i = 1, 5, 7$. The quantities $E(a_i^p)$ and $E(a_i)^p$ have the same time dependence and differ only by a multiplicative constant.

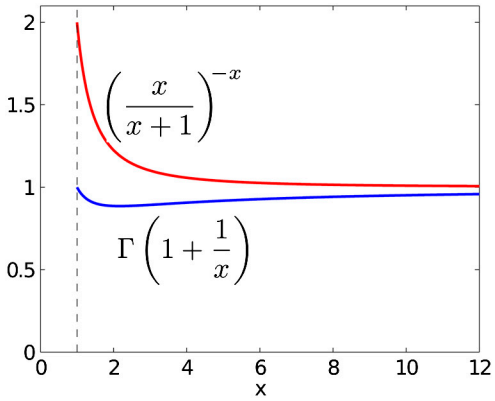
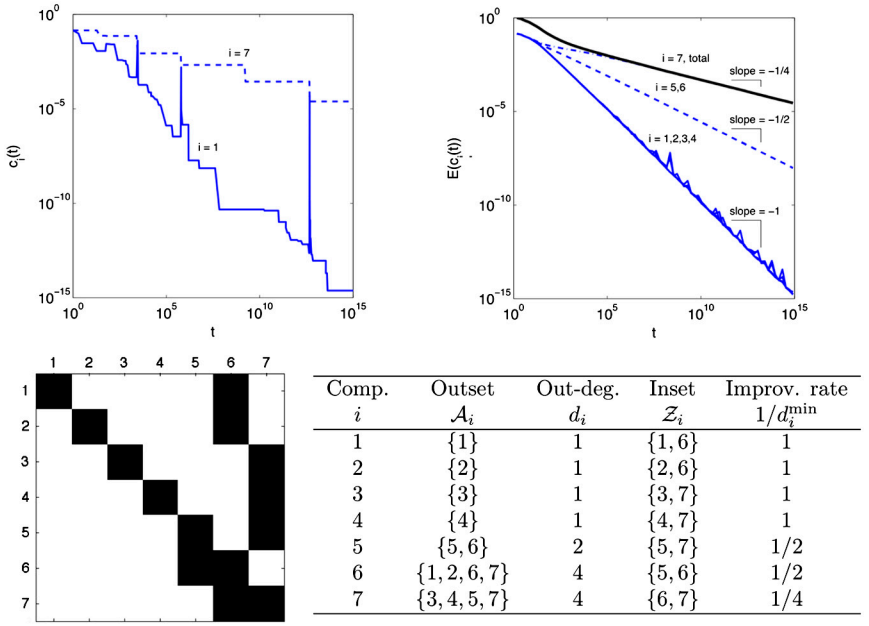


Fig. S2. Comparison of [**S32**] prefactor (top curve) and Eq. **S25** prefactor (bottom curve) as a function of $x = \gamma d$.



Comp. i	Outset $\mathcal{A}_i$	Out-deg. d_i	Inset $\mathcal{Z}_i$	Improv. rate $1/d_i^{\min}$
1	$\{1\}$	1	$\{1,6\}$	1
2	$\{2\}$	1	$\{2,6\}$	1
3	$\{3\}$	1	$\{3,7\}$	1
4	$\{4\}$	1	$\{4,7\}$	1
5	$\{5,6\}$	2	$\{5,7\}$	1/2
6	$\{1,2,6,7\}$	4	$\{5,6\}$	1/2
7	$\{3,4,5,7\}$	4	$\{6,7\}$	1/4

Fig. S3. (*Top Left*) Evolution of $c_1(t)$ and $c_7(t)$ during a single realization. (For a discussion of the cause of the spikes, see the text.) (*Top Right*) Evolution of $E(c_i(t))$ for $i = 1,2,\ldots,7$, averaged over 10^4 realizations. The paths of components 1–4 and those of components 5 and 6 are nearly coincident. (*Bottom Left*) The DSM from Fig. 5*B* of the main article. (*Bottom Right*) List of component outsets, insets, out-degrees, and improvement rates.

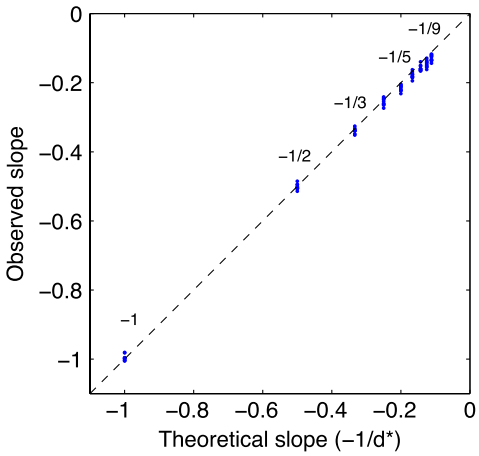


Fig. S4. The asymptotic slope of $E(\kappa(t))$ versus $-1/d^*$ on a double logarithmic scale. Each blue dot averages over 300 simulations of one DSM. 10 random DSMs were generated for each d^*.

Table S1. Values of transformation constants

Series	*a*	*b*	*c*	*d*
Coal plants	−9.0	1.4	−9.0	2.0
Ethanol	−16.2	2.0	−6.0	3.0
PV cells	0.62	1.0	−3.0	1.2
Transistors	−0.26	0.38	−0.64	0.2