Paper to be presented at the DRUID 2012

on

June 19 to June 21

at

CBS, Copenhagen, Denmark,

Adder: a new model for simulating the evolution of technology, with observations on why perfectly knowledgeable agents cannot launch technological revolutions

Janne M. Korhonen
Aalto University
School of Economics
janne.m.korhonen@aalto.fi

Julia Kasmire
Delft University of Technology

j.kasmire@tudelft.nl

Abstract

Computer simulations are increasingly used to study the development, adoption, and evolution of technologies. However, existing models suffer from various drawbacks that may not be easily corrected, among them lack of internal structure in technologies, static environments and practical difficulties of introducing rational or semi-rational search for solutions. This paper discusses the theoretical background and rationale for an improved model, the Adder, and sketches out the model's main features. As an example of the model?s flexibility, we use it to provide insight into why uncertainty about performance of technologies and user needs may be an essential component in the evolution of technology.

Jelcodes:L00,N80

Adder: a new model for simulating the evolution of technology, with observations on why perfectly knowledgeable agents cannot launch technological revolutions

An earlier and shorter version of this paper was presented in the 26th European Conference on Modeling and Simulation 2012 and published in the Proceedings.

ABSTRACT

Computer simulations are increasingly used to study the development, adoption, and evolution of technologies. However, existing models suffer from various drawbacks that may not be easily corrected, among them lack of internal structure in technologies, static environments and practical difficulties of introducing rational or semi-rational search for solutions. This paper discusses the theoretical background and rationale for an improved model, the Adder, and sketches out the model's main features. As an example of the model's flexibility, we use it to provide insight into why uncertainty about performance of technologies and user needs may be an essential component in the evolution of technology.

INTRODUCTION

For the last two decades, researchers studying various aspects of change and evolution in organizations, strategy and technology have increasingly turned to computer simulations to better understand the dynamic and often complex interactions that are inherent in their fields of study. Much of this research has been undertaken using only few basic types of simulation models, resulting to the emergence of "dominant designs" that effectually serve as benchmarks against which novel contributions are evaluated. In particular, the NK and percolation frameworks have seen broad use in the social sciences, with a number of papers appearing in top journals over the last 15 years (e.g. Levinthal 1997, Frenken, 2001, 2006; Auerswald et al, 2000; Ethiraj and Levinthal, 2004; Rivkin 2000; Almirall and Casadesus-Masanell, 2010).

Although these model frameworks continue to be useful, they also suffer from certain limitations. These limitations are particularly visible when the goal is to model a co-evolutionary system that evolves on several different levels simultaneously, or a system whose complexity changes over time. Obviously, both types of systems are of interest to many researchers from fields such as organization science, studies of science and technology, or management of innovations, to name just a few. Developing model frameworks that answer better to their requirements remains a challenge for simulation community. Therefore, this paper first summarizes the primary limitations of NK and percolation models from a perspective of "purposeful systems" researcher, i.e. one that is interested in the technological or social systems and practices that have developed to fulfill a certain purpose (Arthur 2009).

Due to space restrictions, only the common NK, percolation and logic circuit models receive broader discussion in this paper. Desirable features of a "general purpose" model are then discussed. Finally, a suggestion for such a general purpose model - capable of being used in a variety of settings - is introduced.

THEORETICAL BACKGROUND

The notion that technologies "evolve" is not new. For example, Basalla's seminal work (1989) notes that the earliest attempts linking technology and evolution explicitly date from the 19th century. In addition to Basalla's work, the evolutionary processes have been explored by authors such as Frenken (2001, 2006) and Arthur (2009). All broadly agree that while technologies exhibit evolutionary features, strict 1:1 mapping of biological metaphors to technological evolution is not appropriate. Instead, technological evolution should be understood as an instance of "Universal Darwinism" (UD) as introduced by Dennett (1995). In principle, UD states anything that displays variation, selection and heredity will evolve through natural selection, whether or not it would be classified as "alive" in any traditional sense. This universal definition of evolution does not specify *how* variation, selection and heredity work; the sources of variation and the mechanisms of selection may result either from unconscious environmental pressures and random events, or from deliberate "tinkering" and rational selection. Similarly, heredity can operate through biological mechanisms such as DNA, or it can operate through information codified in drawings, patents, and operating manuals. In fact, all technologies are seen to descend in some way from technologies that preceded them (Arthur 2009:18).

However, Arthur (2009) emphasizes the need to consider *combinatorial evolution,* where novel technologies can arise by combination of existing technologies, in addition to incremental change. He finds this particularly useful when explaining developments such as jet engines or radar, that appear to be radical departures from the existing technologies (Arthur 2009:17). Arthur proposes a mechanism of *combinatorial evolution* where novel technologies can arise by combination of existing technologies. Combinatorial evolution (CE) does not abandon incremental variation, selection and retention: it acknowledges that these, too, have an important role to play. Rather, CE proposes an additional mechanism to explain how radical departures from existing technology might happen.

The idea of technologies being recombinations of existing technologies has been long accepted by scholars such as Gilfillan (1935), Schumpeter (1939), and Usher (1954), among others, who describe production or inventions as combinations of materials and forces (Schumpeter) or "new combinations of prior art" (Gilfillan). In Arthur's formulation, a CE mechanism for the evolution of technology would work as follows (Arthur 2009:21-24):

1. Early technologies form using existing primitive technologies as components.

2. These new technologies in time become possible components, or building blocks, for the construction of further new technologies.

3. This implies that technologies have an internal structure, a hierarchy of subsystems and sub-subsystems.

4. The complex technologies form using simpler ones as components.

5. The overall collection of technologies bootstraps itself upward from the few to the many and from the simple to the complex.

Furthermore, it is notable that few modelers have tried to test alternative assumptions of rationality, preferring to assume relatively myopic search (taking into account only one decision at any given time) in the technology landscape. However, solution search by humans is usually assumed to have some overarching direction, even though it is clearly “boundedly rational” (e.g. Simon 1982). we therefore argue that any model of technological evolution should accommodate both essentially random events - "mutations" - and non-random, somewhat directed but boundedly rational search for new technologies. Thus,

6. The evolution of technology happens both as a result of essentially random events and boundedly rational search for and evaluation of new solutions.

It should be noted that most scholars of technology are in a broad agreement that "technologies" should be understood to mean not just material artefacts, but also nonmaterial methods, processes and devices that are means to fulfill a human purpose (e.g. Arthur 2009:28). This broader definition of technology thus includes fields such as management practices and strategies.

In the following, we will use the above formulation as a basis for evaluating three important computer models of technological evolution. Due to space restrictions, we only evaluate the most widely known models in use. We will briefly discuss their limitations and then introduce an improved model.

IMPORTANT COMPUTER MODELS OF TECHNOLOGICAL EVOLUTION: NK, PERCOLATION AND LOGIC CIRCUITS

The NK model is perhaps the most widely used modeling framework in management and innovation research today. Originally introduced by Anderson (1983) to model spin-glass interactions in physics, the model became famous after Kauffman's seminal book Origins of Order (1993). The model was first used to study technological evolution by Kauffman and Macready (1995) and has since been used in a large number of important publications in the fields of organization science and innovation research (e.g. Levinthal 1997, Frenken, 2001, 2006; Auerswald et al, 2000; Ethiraj and Levinthal, 2004; Rivkin 2000; Almirall and Casadesus-Masanell, 2010).

To very briefly summarize, the NK model consists of a solution space and agent(s) that search the space for "better" solutions. The parameters N and K characterize the solution space, and the search behavior of the agent is determined by the researcher.

In management and innovation literature, N typically represents the number of decisions that have to be made, for example among alternative components, alternative strategic options, and the like. Parameter K controls how interconnected these decisions are, i.e. how many other decisions are affected by any one single decision. Decisions are usually binary (0/1). Typical search strategies are random mutation, local myopic search by altering a single decision, and random "long jumps" where more than one decision is altered simultaneously.

The solution space's topology ranges from simple and smooth "fully correlated" landscapes with a single optimum solution to roughly correlated (similar solutions generally have similar performance, but not always) to uncorrelated (no correlation between similarity of two solutions and their performance).

In its original formulation, the NK model implies that agents solve exogenous problems in a fixed solution space and in a static environment. While these simplifications allow for computationally tractable and transparent models, they do pose difficulties for modeling endogenous situations, i.e. where attributes of solution space change over time (Ganco and Hoetker 2009). In particular, changing interdependencies and complexity remains difficult to model. Solutions to the problem have been proposed (e.g. Altenberg 1994, 1997, Valente 2008, Ganco and Hoetker 2009) and used (Frenken 2006). However, as Ganco and Hoetker (2009) state, even the endogenous NK model is a poor fit for modeling conscientious decisions of human actors.

Another problem the NK model has when modeling combinatorial evolution is that it ignores the internal structure of technologies. Although many authors using NK models (such as Murmann and Frenken 2006 and Frenken 2006) do specifically argue that technologies have a hierarchical structure and consist of systems, subsystems and sub-subsystems (etc.), the model itself does not really allow for this. Thus, individual technologies are independent of each other: the NK model does not really model the evolution and interrelationships of technologies, rather it models a fixed environment where all the possible components are given, and the task is simply to search the landscape for the optimum combination of those components. Furthermore, the lack of internal structure makes the improvement of existing technologies through improvements in their components - a feature visible in any case study of technology - clearly impossible. While one could conceive potential solutions to these problems (e.g. "nested" NK models, where each individual component is a result of individual landscape search), such solutions become cumbersome and computationally intensive very quickly.

Even with modifications, a major problem with NK models is introducing simulated rationality to the “design” process. As it stands, most studies that utilize the NK model use myopic search heuristics, basically looking only one “move” ahead at each given point of time. The most common search heuristic in management, organizational and innovation literature is the “greedy” search, where agents evaluate all the possible one-decision changes and pick the one that provides the greatest improvement in fitness. Even more myopic search strategies have been used.

While these are often justified simplifications and help capture the problems of path dependency very nicely, with low N they result to overly abstract models, where a single 0/1 decision covers a great variety of nuanced details, or – with high N – to too myopic search, where the decisions taken at any one time are unrealistically small. It would be interesting to test how the results would differ, given more foresighted agents. Unfortunately, adding foresight is not easy. The computational complexity of determining improvement pathways in more complex situations makes the simple “brute force” method impractical, while advanced AI heuristics often remain beyond the skills of a researcher without computer science background.

Percolation Models

Percolation models have been used to model the dynamics of adoption (Stauffer and Aharony 1994; Grebel 2004) and technological innovation (Silverberg and Verspagen 2005). For the purposes of this paper, Silverberg and Verspagen's percolation-in-a-lattice model is the most relevant. In the model, sites in the lattice represent potential technologies. These ideas can be either impossible to realize, possible but not yet discovered, discovered but not yet viable, or discovered and viable. A site turns from discovered but not yet viable to a viable technology if and only if there exists a contiguous path of discovered technologies connecting the site to the "baseline." In this model, the dynamics of innovation processes can be modeled as a local search process on the *n*-dimensional lattice (2-dimensional lattices being most common). The difficulty of the search can be tuned by adjusting the share of sites that are impossible to realize.

The percolation model usefully reproduces several "stylized facts" about technological innovation: innovation “avalanches” in time and in clustering of technology space. This is due to "keystone" technologies (sites) that cause chain reactions of other technologies becoming viable. The model also overcomes a problem with NK models by clearly having the discoveries to depend upon previous discoveries.

However, the model is still not ideal from a point of view of researcher interested in modeling the evolution of technological systems. Just as in the NK model, the environment is still essentially static, and extending the model to include co-evolutionary features poses certain difficulties. Again, the complexity of technologies themselves is ignored, as the technologies do not have any internal structure: they simply are feasible, or they are not. Furthermore, as the technologies do not consist of components, an incremental improvement of technologies through e.g. component cost reductions is ignored as well. Although ignoring this network of elements that make up real-life technologies is a justifiable abstraction in many settings, improved models would be desirable.

Finally, the percolation model also suffers from the difficulties in implementing more rational search. Although AI algorithms capable of navigating the lattice certainly exist, implementing them may be beyond the skills of many researchers.

Logic Circuit Model

Another candidate for a general purpose model of technological evolution was originally introduced by Arthur and Polak (2006) and latter used by Arthur (2007, 2009) to illustrate his formulation of the CE framework, as well as by scholars from other fields (e.g. White 2008). In their model, technological build-out begins from simple "primitive" technologies. These technologies are randomly combined to result to more complex technologies, which themselves are then potential components in future technologies. The system includes concrete needs instead of abstract fitness values, and technologies that better satisfy these needs or have fewer components - i.e. are cheaper - than their alternatives supersede older technologies.

The model is implemented using simple logic circuits (NAND gates in the most common version) as primitive technologies. Needs include simple logical functions, such as 2-bit adders. New technologies are evaluated against how closely they fulfill a need's truth table, and how few primitive components they use in doing so.

This model successfully captures the interlinked build-out of technologies from simpler components, and replicates stylized facts such as avalanches of "creative destruction" when a significant innovation suddenly makes obsolete many old technologies. However, its implementation creates some problems, as well.

The first problem is that the model is relatively fixed. Unlike in NK and percolation models, which allow essentially unlimited variety in e.g. the form of the landscape, there are only so many simple logical needs (e.g. NOT, IMPLY, *n*-way-or, 2-bit adder, 8-bit adder and so on) that can serve as targets for the simulation to reach. Although it is possible to reduce the list of targets, increasing the list of targets while avoiding functional duplicates is more difficult. This limitation creates difficulties for studying settings where more or less every technology might fulfill some need.

The second problem is that the logic circuit model is somewhat challenging to implement and debug, and relatively demanding in computational power. While these difficulties can definitely be overcome, a simpler model would be desirable from many standpoints.

The third and perhaps the most important problem is the same that plagues the NK and percolation models: introducing bounded rationality is difficult. As a result, even Arthur and Polak (2006) are reduced to arguing that their model of random combinations *can* be representative of search through solution space, given certain assumptions. While this may be true, one would very much like to test how the assumption affects this otherwise excellent model.

TOWARDS A BETTER MODEL: THE “ADDER” SUGGESTION

The Adder model simplifies Arthur and Polak’s model by replacing logic circuits and Boolean arithmetic with real numbers and arithmetical expressions. Each experiment starts with elementary components (primitives) and arithmetical operators, usually with number 1 and operators plus and minus. During each time step of the experiment, developer agent(s) alter existing “technologies” by adding or removing components, hence the name of the model. The resultant technologies are then evaluated against goals and added to the repertoire of possible components. The objective of the system is to satisfy a certain set of needs or goals, expressed as real numbers. These needs can be thought to represent the needs that drive technology evolution, and as simplifications of logical operator needs used by Arthur and Polak (2006). The numbers can be either drawn randomly or by using some distribution scheme. Compared to a relatively fixed set of goals in the original model, this allows for certain flexibility in studying different technological landscapes (some landscapes might have feasible technologies more clustered in design space than others, for example).

As an example, let us assume that one of the goals is "10," that this is the first step and therefore the component "1" and operators plus and minus, and that the selected method of alteration is random draw of 0 to 12 components and operators. A possible draw could be

1+1-1+1+1-1+1+1,

"producing" the value "4." Although this technology did not fulfil the goal in itself, it is now added to the repertoire of possible components and is therefore available for use in the next step. Suppose that the next draw gets the components

-4-1-1+4+4+4+4,

that produce "10" and thus satisfy the first goal. The process continues until desired set of conditions is reached, for example, when all the goals are satisfied or the simulation has progressed for a predetermined time.

It should be noted that this completely random approach may not be the best representation of technological development. Another possible variant would add one element at a time, closing on the target.

Evaluation by Cost and Fitness

The "goodness" or "fitness" of these technologies can be evaluated in a variety of ways, depending on the requirement of particular experiment.

The primary important evaluation criteria is the "cost" of the technology. In its simplest, the cost is determined by counting the number of primitive elements required for the technology. To continue the above example, the primitive components (1's) have a cost of 1. Component technologies, such as technology "4" above, have a cost equal to the cost of primitives within it. Thus, the above technology "4" would cost 8.

It is evident that such costing schemes neatly capture one of the important mechanisms in the evolution of technologies: that many technologies, when first developed, are very expensive, but become cheaper as R&D efforts are made towards improving the efficiency and manufacturing processes, for example. The "technologies" encapsulated in the model can be thought of as simplified idealisations of production recipes or assembly instructions, subject to improvements as more streamlined processes are found.

For example, the technology "4" in the model could be superseded by several generations of more efficient technologies, with the ultimate limit of efficiency being

1+1+1+1,

with a cost of 4. It should be noted that in contrast to Arthur and Polak's original model with its somewhat difficult-to-analyze circuit designs, determining the efficiency limits and the most efficient technology possible is always trivial (i.e. when Cost = Product).

The secondary evaluation criteria is the fitness-for-purpose, that is, how close the technology gets to the target. A default setting of the model accepts only those new technologies that are either i) closer to the target value than existing technologies or ii) cheaper than existing technologies. Other evaluation criteria are possible.

Obsoleted Technologies

The Adder models the obsolescence of technologies through basic mechanisms described above. If a new technology proves to be either fitter or cheaper than existing technologies, it takes their place in the repertoire of technologies that are used as a pool of possible components in the future. If the new technology is simply a cheaper version of already existing technology, the technologies currently using the old version are updated. As their costs are updated in turn, it is possible that a development of a new component triggers an avalanche of replacements. The size-frequency distribution of these avalanches (see Fig. 1) shows hints of power law distribution, indicative of self-organized criticality (Pak and Wiesenfeld 1988)

This obsolescence does not, however, necessarily obsolete other technologies that are already using the now-obsoleted technology. This allows for "legacy" technologies, where otherwise obsoleted components remain in use as parts of older systems. As an example, suppose that a target "10" is reached, while there exist technologies that use the tech 9 as a component. Although future technologies will not use tech 9 any longer, any technologies that have tech 9 as a component will retain it. It is even possible that tech 9 is incrementally developed towards a cheaper version, resulting to decreased costs for technologies that use it.

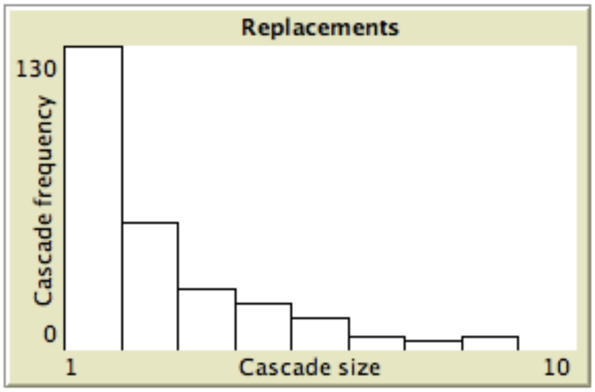

Figure 1: Sample Distribution of "Replacement Cascades"

Bounded Rationality

It is easy to see that implementing even fully rational agents is nearly trivial in this model. For example, it's easy to have the agents determine the optimum components they'd need in order to reach a certain target. One can easily go further and implement bounded rationality (Simon 1982) by e.g. introducing random uncertainty into the calculation. Thus, studying the effects of bounded rationality in the product development process becomes possible.

Tuneable Difficulty of Search

The description above assumes that all technologies – all real numbers - are possible. This, however, is hardly the case in reality: one can imagine products such as chocolate coffee pots that may be feasible, but unviable.

A simple way of tuning the difficulty of the search in this landscape is the addition of "anti-targets" or "valleys." These are simply numbers that are either not allowed or that incur some kind of a penalty. The density of these anti-targets can be easily adjusted, and thus different technological landscapes can be explored using the Adder model. It should be noted that these anti-targets correspond roughly to sites that are impossible to realize in the lattice percolation model described above.

An example of how the density of targets and anti-targets may be used to tune the difficulty of search for new technologies is shown below in Figure 2, where the y axis reports the highest technology reached at the moment of time.

Keystone technologies

Key variables of the model – the density and spread of targets and anti-targets – can be set to replicate an important finding of Arthur and Polak's model, namely, that without early low-level needs, more advanced needs are difficult or impossible to satisfy. In other words, certain technologies may serve as "keystone" technologies, enabling further technological development. However, as the Adder is far more tuneable than the original, we have also found that these results depend on the setting of the variables in question. If (nearly) all technologies are feasible (no anti-targets), the lack of early needs does not stop the progress towards more complex technologies.

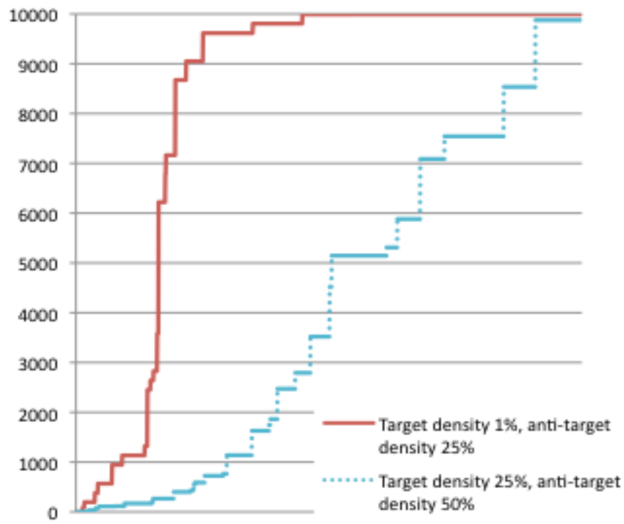

Figure 2: Differences in Performance With Two Different Target/Anti-Target Ratios'

Incremental versus Radical Innovation

The Adder can be used to model either incremental improvement of existing technologies or radical departures from the existing state-of-the-art, or both at the same time. Incremental improvement may take place by either randomly redrawing components and operators for existing technologies, or by more "rational" methods. Similarly, radical innovation can be modelled either as random draws or as rational attempts to reach targets (with or without uncertainty).

Multi-agent Industrial Ecosystems

The Adder can serve as a basis for a multi-agent simulation where agents have different search strategies. An industrial ecosystem can be modelled in this manner; for example, the Adder can model the division of labour between component producers and original design manufacturers. Features such as patents and knowledge sharing are possible to implement as well.

Extending the model to multiple dimensions

The model detailed above can be thought to abstract away significant complexity by treating goals and technologies as specific points along a one-dimensional line. Although we feel that this abstraction does not unduly affect the results, it is easy to extend the model to as many dimensions as is desired. As an example, two-dimensional model could have target coordinates (x, y) instead of single target. The technologies would therefore also have two "products," distinct from each other. The dimensions could be thought to correspond to various real-life needs and functionalities that actual technologies try to satisfy.

EXAMPLE CASE: WHY (ALMOST) PERFECT KNOWLEDGE MAY NOT BE GOOD FOR TECHNOLOGY DEVELOPERS

As an example demonstrating the Adder's basic flexibility, we have used the basic framework to develop a simulation of the effects of uncertainty on technological evolution. The goal of the following model is to study how the understanding of technological possibilities and user needs may affect technological development, and how agents with perfect knowledge of needs and technologies would develop new technologies. The results presented here are preliminary and should be considered a work in progress, as certain scenarios still need to be simulated.

In a review of evolutionary theories of technological change, Nelson (2005) notes that two key variables – the strength of technological understanding and the knowledge of user needs – seem to control the rate and direction of technological advance. If both are very strong, argues Nelson, technological advance can almost be planned. Presumably, such planned advance would be more efficient than one relying on less complete knowledge.

However, it is also understood that technological advance has benefited from serendipitous discoveries and happy accidents. An oft-cited example is the development of Post-It notes (and an entire family of products) as a result of failed adhesive development. In cases like these, lack of understanding of technology in question ultimately led to the invention being made: had the 3M corporation had a complete understanding of the specific adhesive technology, its developers would not have made the error of creating "failed" adhesives, and possibly the need for Post-It notes would not have been found to exist. This dichotomy begs a question: what is the role of uncertainty in the development of technology?

Adder model with target-seeking, uncertainty and learning

To answer the question, we have first slightly reconceptualised Nelson's two key variables by adding a third, auxiliary variable: the space of adjacent possible. This represents in a stylized form the observation made by e.g. Arthur (2009) that technological "frontier" advances over time, as new technologies open new combinatory possibilities and create new needs that were not evident in advance. It can also be thought of as the space of needs that are "visible" to developers at any one time, or just reachable by any combination of current technologies. (As an example, any combination of the components available to a 16th century inventor could not have produced a transistor; therefore, transistor lay outside the space of adjacent possible at the time.) In its simplest, the size of this space is a function of the number of technologies in existence, but it may also be affected both by technological understanding and knowledge of user needs.

Following this reconceptualization, we modified the basic Adder framework by replacing random draw of components with a goal-seeking algorithm. In the model, developer agents have a perception (possibly incorrect) of user needs, i.e. target values, and an understanding (again, possibly incorrect) of properties of components in use. During each turn of the simulation, a single developer agent firsts selects a single perceived need it will attempt to satisfy. The need to satisfy, or the target *T*, is selected from a space of adjacent possible targets. This non-monotonically increasing one-dimensional space is simply the space of those target values that neighbour already discovered target values, but have not been yet discovered. The size of adjacent possible is governed by neighbourhood size *S*, which in this model simply means how many *n*-neighbours of already found targets are included in the adjacent possible (i.e. *S* of 2 would mean that adjacent possible would increase by a maximum of four, by first and second neighbours of each discovered target value. An unlimited *S* means that all targets are visible to the agents). In these simulations, we have kept the *S* fixed, but as mentioned above, it could also be dependent on technological understanding and knowledge of user needs.

Once the target is selected, the agent attempts to reach it by combining together available components. When developing a technology, the agent evaluates the expression

$$T \pm U_t - P \pm U_p$$

and

$$T \pm U_t - C \pm U_c$$

where *T* is the target, *P* the product of the technology under development, *C* the contribution of a component under evaluation, and U_t, U_p and U_c associated uncertainties. If a component is found that brings the new combination closer to target *T*, it is added to the combination (so that $P_{t+1} = P_t + C$), and the evaluation round starts again.

The goal of the agent is to get as close to the perceived target as possible using the technologies available at the time, using as few components as possible.

The search for new technologies is made more difficult by introducing a variable density of "valleys." These valleys represent combinations that are unfeasible for any reason. At no point in the development of technology can the combination's real value, *P,* equal any valley value. However, the valleys can be "leapfrogged" by adding sufficiently "large" components. (For example, if "3" is a valley value but "4" is not, combinations 2+1 and 2+1+1 are not viable, but a combination 2+2 is. Note that this is subtly different from the standard Adder model outlined in this paper.) Viable technologies are added to the repertoire of technologies and can be used as components in the following turns.

As the agent tests different components and repeatedly tries to satisfy a given target, it gains experience of both technology and the needs, enabling it to make more accurate assessments of what components are needed to satisfy a given need. The uncertainty associated with both technologies and needs diminishes according to a standard learning curve model,

$$U = U_i(x+1)^{\log_2 b}$$

where U_i is the initial uncertainty, x the number of times a technology has been used as a component, and b the learning percentage.

Results

The simulation was run with varying parameter values for initial uncertainty and target and valley densities. All the simulations had learning percentage set to 80%, and one primitive component, "1," and (+) operator were used. The effects of neighborhood size were tested for $S = \infty$ and 1. The plots of two representative simulation runs (n = 5, S = 1) are shown below:

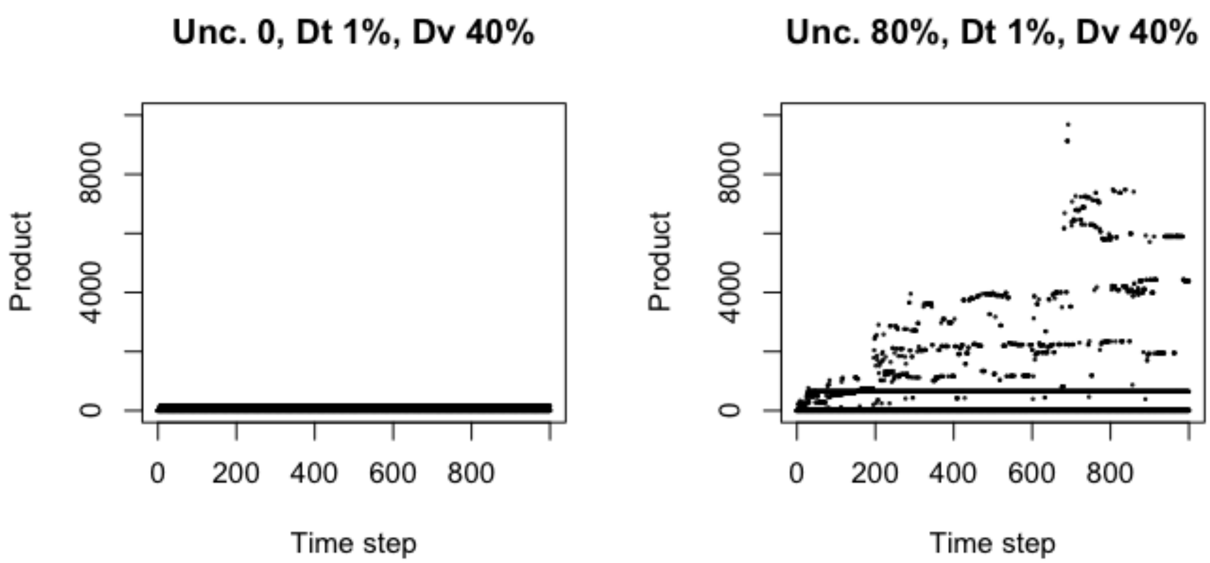

Figure 3: Representative Simulation Results, 5 Runs For Each Setting

On the left panel, the agent has perfect knowledge of user needs and technologies. On the right, it starts with an 80% initial uncertainty about target values and technology performance. In both simulations, the density of target values is 1% and the density of valleys or infeasible technologies 40% out of all the possible technologies. The results are robust to these parameters.

The preliminary results from this and other test runs indicate that the overall behavior of the model seems to be in line with stylized facts broadly used in other simulations of technological development, e.g. Silverberg and Verspagen (2005). For example, the technological development in the model often displays self-organized criticality and bursts of technological development: usually, the technological development is very slow over a number of turns, until a major valley is bypassed. After that, the speed of development may increase exponentially, until another large valley blocks development – for a while.

As the sample plots show, surprisingly, uncertainty seems to be a requirement for technological progress. When the agents have perfect knowledge of the needs and the performance of their technologies, they do not make mistakes that help them overcome technological barriers (valleys). Instead, they get stuck after a few non-primitive technologies, at most, are developed, and before the space of adjacent possible can expand.

The exception is when $S = \infty$, as agents are then always able to see new, unfulfilled needs. In these simulations, all the targets are reached quickly and efficiently, as can be expected. However, as this case corresponds to the case where all the possible *combinations* of components and all the possible user needs are known in advance, it would seem to be even more unrealistic than having perfect knowledge for components or user needs themselves.

A tentative conclusion drawn from these preliminary results might suggest that Nelson's two key variables do not actually regulate the advance of technology in a way originally suggested. The model tested what effects the understanding of the performance of specific technological components and knowledge of specific user needs have, and the conclusion seems to be that it is actually *uncertainty* rather than knowledge about these that enables developers to take risks and try out new ideas. These results would seem to square with anecdotal evidence and observations about several technological revolutions (including the steam engine and the personal computer revolution), which were not initiated by persons with complete knowledge and understanding of what the technology could do or what the user needs actually were.

However, the understanding and knowledge *do* seem to have an effect in a more roundabout way, if increased understanding increases the number of combinatory possibilities visible to the developers – i.e. the size of the space of adjacent possible. Although this assumption was not explicitly tested in the simulation, it would seem plausible that better understanding of technologies and better knowledge of user needs would help developers to think about new, successful combinations. Follow-up research will test whether these tentative conclusions hold, and pick apart the mechanisms through which knowledge and uncertainty affect technological advance.

DISCUSSION

In this paper, we have briefly discussed the two most common simulation frameworks used to model technological evolution. The Adder model is designed to be a streamlined version of an alternative model. Being easy to understand, implement, and modify, we believe it has potential to help researchers interested in the evolution of technologies to study settings hitherto unreachable with existing model frameworks.

Of course, the Adder is not meant to be everything for everyone. NK models are still very well suited for studying complex interactions in a relatively static environment. Similarly, percolation models are perfectly adequate for studying the spread of technologies or concepts over time. A benefit of these simpler models is that they are largely parameterized by a single variable – the K in NK models and the density of available sites in the percolation model. Such simplicity makes for parsimonious models, a desirable feature in most cases. However, we believe that unmodified NK and percolation models are not very good fit for studying the evolution of "ideas" over time and in a dynamic environment, i.e. where the ideas themselves have an effect on the selection environment.

The model lends itself well to further research. In addition to some of the possibilities outlined above, the model can be used in e.g. studying the effects of constraints on technological development, the effect of different product development strategies, and perhaps even the workings of a broader, interlinked economy of technology-developing organizations. The flexibility of the model is briefly demonstrated through application to an interesting research question, and the results suggest that the model can contribute usefully to scientific discussion. It should be also noted that the actual implementation of the example model took a researcher without prior programming experience only about a week, an indication of the benefits of simplicity. Our future papers will take greater advantage of the model detailed here, and we welcome any researcher who is interested in joining the effort.

REFERENCES

Almirall, E., Casadesus-Masanell, R., 2010. Open versus closed innovation: A model of discovery and divergence. *Academy of Management Review* 35, 27-47.

Altenberg, L., 1997. NK fitness landscapes, in: Back, T., Fogel, D., Michalewicz, Z. (Eds.), *The Handbook of Evolutionary Computation*. Oxford University Press, Oxford.

Anderson, P.W., 1983. Suggested mode for prebiotic evolution: The use of chaos. *Proceedings of the National Academy of Sciences of the United States of America* 80, 3386.

Arthur, B.W., 2009. *The Nature of Technology: What it is and how it evolves*. Free Press, New York.

Arthur, W.B., Polak, W., 2006. The evolution of technology within a simple computer model. *Complexity* 11, 23-31.

Auerswald, P., Kauffman, S.A., Lobo, J., Shell, K., 2000. The production recipes approach to modeling technological innovation: An application to learning by doing. *Journal of Economic Dynamics and Control* 24, 389-450.

Bak, P., Wiesenfeld, K., 1988. Self-Organized Criticality: An Explanation for 1/f Noise. *Physical Review A* 38, 364.

Basalla, G., 1989. *The Evolution of Technology*. Cambridge University Press, Cambridge, UK.

Dennett, D.C., 1995. *Darwin's Dangerous Idea: Evolution and the meanings of life*. Simon & Schuster, New York.

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

Frenken, K., 2001. *Understanding Product Innovation using Complex Systems Theory*. PhD thesis, Faculty of Social and Behavioral Sciences, University of Amsterdam.

Frenken, K., 2006. *Innovation, Evolution and Complexity Theory*. Edward Elgar, Cheltenham and Northampton.

Ganco, M., Hoetker, G., 2009. NK Modeling Methodology in the Strategy Literature: Bounded Search on a Rugged Landscape, in: Ketchen, D., Bergh, D. (Eds.), *Research Methodology in Strategy and Management Volume 5*. Emerald, pp. 237-268.

Gilfillan, S.C., 1935. *The Sociology of Invention*. Follett Publishing, Chicago.

Grebel, T., 2004. *Entrepreneurship: A New Perspective*. Routledge, London & New York.

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

Kauffman, S.A., Macready, W.G., 1995. Technological evolution and adaptive organizations. *Complexity* 1, 26-43.

Levinthal, D.A., 1997. Adaptation on Rugged Landscapes. *Management Science* 43, 934-950.

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

Nelson, R. R. (2005). Perspectives on technological evolution. In K. Dopfer (Ed.), *The Evolutionary Foundation of Economics* (pp. 461-471). Cambridge: Cambridge University Press.

Rivkin, J.W., 2000. Imitation of complex strategies. *Management Science* 46, 824-844.

Schumpeter, J.A., 1939. *Business Cycles*. McGraw-Hill, New York.

Silverberg, G., Verspagen, B., 2005. A percolation model of innovation in complex technology spaces. *Journal of Economic Dynamics and Control* 29, 225-244.

Simon, H.A., 1982. Reply: Surrogates for Uncertain Decision Problems, in: *Models of Bounded Rationality, Volume 1*. MIT Press, Cambridge, Mass., pp. 235-244.

Stauffer, D., Aharony, A., 1994. *Introduction to Percolation Theory*. Taylor and Francis, London.

Usher, A.P., 1954. *A History of Mechanical Inventions*. Harvard University Press, Cambridge, Mass.

Valente, M., 2008. Pseudo-NK: an Enhanced Model of Complexity. LEM Working Paper 2008/26, Università dell'Aquila.

White, A.A., 2008. A developmental perspective on technological change. *WORLD ARCHAEOLOGY* 40, 597-608.