Growing Adaptive Organizations: An Agent-Based Computational Approach

Joshua M. Epstein
February 26, 2003 Draft

Abstract

Individual agents endogenously generate internal organizational structures (e.g., local hierarchies and trading regimes). Optimal histories of structural adaptation in dynamic environments are computed and displayed.

* The author is a Senior Fellow in Economic Studies and in Governance Studies at The Brookings Institution, a Member of the Brookings-Johns Hopkins Center on Social and Economic Dynamics, and a Member of the External Faculty of The Santa Fe Institute.
† The author particularly acknowledges Software Engineer Joshua Miller of the Bios Group for implementing the model in Ascape, for computational analysis, and for valuable discussions. He also thanks Daniel Teitelbaum of the Bios Group for his contributions. For valuable comments and discussions, he thanks Chris Carroll, Joseph Harrington, Michael Heaney, John Miller, Benoit Morel, John Sterman, Anjali Sastry, Nelson Repenning, and Peyton Young. The Cap Gemini Ernst & Young Center for Business Innovation funded this research and the author thanks Christopher Meyer and Eric Mankin for their support and for numerous valuable discussions.
Introduction

What constitutes an adaptive organization? What would constitute optimal structural adaptation in a dynamic environment? Can one “grow” optimally adaptive organizations from the bottom up—that is, devise rules of individual behavior that endogenously generate optimal structural adaptations? There is, of course, a large literature on the origin of firms, on the size distribution of firms in an economy, and on a host of related topics (see refs 1-5, 7, 9-15, 17-18). However, I am unaware of any explicit model in which individual agents endogenously generate internal organizational structures that adapt optimally to dynamic environments. The present paper develops such a model, using the agent-based technique (5). It is important to note that I do not purport to model any existing organization, or to “fit” the model to data of any sort. Rather, the aim, using a highly idealized model, is to illuminate what sorts of individual (micro) agent rules confer adaptiveness on the larger (macro) enterprise. There will of course be many simplifying assumptions, but hopefully they will not entirely subvert that broad objective.

Conceptually, we posit that organizations, such as firms in an economy, exist in economic environments (defined carefully below) that may change over time; environments are dynamic. Over any period—over any day, let us say—organizations have a structure. That is to say, there is, in principle, a graph representing each agent’s information (internal and external) and its span of authority (its set of resources and permissible manipulations of them); and there is a specific deployment of resources (e.g., labor and capital). Ceteris paribus, the state of the economic environment in a given period, combined with the organization’s structure in that period, jointly determine the organization’s performance in that period: this could mean it’s total profit, it’s market share, or other measures. In general, changes in structure induce changes in performance.

Now, imagine stipulating a k-period environmental dynamic (we do this concretely below). Then, for every “candidate” k-period history of structural adaptation, there is a corresponding history of organizational performance (e.g., total profit over the k periods). It is therefore perfectly natural to pose the question: What is the optimal history of structural adaptation in that dynamic environment?

Given a particular environmental dynamic, would an optimal history of structural adaptation involve periods of extreme hierarchy, separated by relatively “flat” internal trading regimes? This is the sort of general question we wish to explore.

However, we take a generative approach. That is, we want a single fixed set of operating rules and parameters at the individual agent level that will generate of “grow,” an entire optimal history of structural adaptation “from the bottom up.” The autonomous agents, for example, should “grow” hierarchies when they are needed and dissolve them when they are obsolete. The aim is to characterize rigorously what would constitute optimal adaptiveness under various assumptions. While there is no empirical claim particularly, the results appear to call into question the optimality of certain ubiquitous forms, notably, the pyramidal hierarchy, which—at least in this model--turns out to be optimal only in quite restricted cases.
Organization

The paper proceeds in six parts. Part I gives the basic set-up of the model. Part II explores the spontaneous emergence and dissolution of hierarchy, with internal trade proscribed. In Part III, we explore internal trade alone as a means of re-allocating resources within the organization. Having explained the mechanics of hierarchy and trade, I introduce a particular dynamic environment. Against this environment, we examine the effectiveness of pure trade and pure hierarchy as solutions.

Now, without introducing some objective function, no claims can be made about optimality, or about the relative performance, or “fitness,” of different modes of organization. Accordingly, in Part IV, we introduce a very general objective function for the organization. By setting a parameter (k), it specializes to profit maximization (k=0), to market share maximization (k=1), and to hybrids of the two (0<k<1). Then, in Part V, for various choices of objective function (e.g., profit maximizing) we sweep the entire parameter space of the model for the optimal parameters. For those parameters, we then show the optimal history of structural adaptation. Finally, in Part VI various extensions are proposed.

Part I. Basic Model

I have used the term “environment” repeatedly and without definition. Of course, an organization’s economic environment could include everything from its technological opportunities to pollution regulations to the prime interest rate. While my model can be generalized (see extensions), the environment is represented simply as a dynamic pattern (a flux) of “opportunities,” depicted as a flow of red dots moving left to right toward the “market” of the enterprise. This “market” consists of 32 contiguous cells, depicted as a vertical array. The market is manned by the enterprise’s labor force of (at most) 32 workers, depicted as solid blue squares. Each of these workers is completely myopic, and controls just the cell he occupies. If an incident red dot runs into a blue square, the red is considered to be intercepted by the blue worker; that opportunity (red dot) has been “taken” by the enterprise. For an enterprise to suffer no red penetrations, workers (blue squares) would have to be positioned to intercept every incident red dot (see Figure 1).

All figures, tables, and other graphics appear in Appendix II.

An initial enterprise and an incoming red opportunity flux (the environment) is shown in Figure 1. An initial condition for the enterprise always consists of some distribution of blue workers (level-0 managers) and an initial level-1 management layer (comprised of 16 managers). Each level-1 manager controls a 2-cell market segment (the cell positioned 4 spaces to the left and the cell one space north of that). If, at any point, there are workers in cells controlled by a manager, that manager is depicted as a solid dot. If there are no workers under his control the manager is depicted as a hollow dot. So, in Figure 1, 7 solid managers actually control labor; the hollow rest are monitoring their sector of the market for penetrations. The environment is essentially a block of red dots in the south of the market. They are marching toward the enterprise’s space.
Workers cannot move themselves. The task of management is to allocate them effectively. Clearly, if no reallocation of labor occurs, the red dots will penetrate the blue enterprise’s front line. When a red penetrates, he is colored yellow, as shown in Figure 2.

Spans of Control

Given certain objective functions for the enterprise, it may prove efficient to generate a hierarchy (the mechanics of this are presented below). Figure 3 depicts the maximal hierarchy. In a hierarchy of any height, each level-n manager can shift workers about the $2^n$ cells he controls. As noted above, each level 1 manager controls a 2-cell market segment. Each level 2 manager controls a 4-cell segment, each level 3 an 8-cell segment, and so on up to the level-5 CEO, who can shift labor anywhere across the entire 32-cell market of the enterprise. This CEO agent supercedes all his subordinates, and so he is the only solid agent. In this exposition, the labor force is fixed.

Labor Allocation Rule

The labor allocation rule is syntactically identical for all managers, though of course their spans of control are not identical. The rule, however is:

$L$: Within your span of control, identify all labor not currently intercepting (call that List 1) and identify all sites subject to imminent (next period) attack; call that List 2. Choose a random laborer from List 1 and move him to a random site from list 2. Repeat until no sites are threatened or List 1 is exhausted, whichever occurs first.

Basically, the rule amounts to identifying everybody who’s leaning on his rake, and throwing him at an outstanding problem. Now, the essential problem arises as follows.

Labor is numerically adequate, but it is misallocated. There is a gap in blue coverage at cell three (counting up from the bottom), while there is an unopposed blue defender at Cell 2. Clearly, to block the penetration, he should be shifted up one site. However, neither Level-1 manager is empowered to do that, since each controls only the two sites below him (shown by arrows). There are two pure approaches to the problem:
[1] **Hierarchy.** A Level-2 manager (controlling all four sites) is created (as shown) to shift the free resource up to its efficient location.

[2] **Internal Trade.** The Level 1 manager who is facing penetration announces, or “posts,” his demand for labor to all other managers in his Level. Those with an excess supply of labor may choose to respond with an internal transfer.

Next I describe in detail the mechanics by which hierarchies endogenously emerge and dissolve, and then the mechanics of internal trade.

**Mechanics of Hierarchy with Trade Proscribed**

Recall that a level-n manager controls a market segment of 2^n cells. Over this segment, he allocates labor according to the labor allocation rule (L) stated above. The manager has two penetration thresholds, T_{min} and T_{max}. (To exercise the model, these are initially set by the user. Later we compute their optimal values). In this paper, we will assume that these values differ by management level but are common across a given level. Every manager has a finite memory (e.g., the last ten periods), m. The manager computes the average penetrations over this memory (i.e., he computes the total number of penetrations of his market segment over the last m periods and divides by m); call that result P.

**Upward Hierarchy Rule:** If P > T_{max}, then (with probability 1-upward inertia) a manager of level n+1 is created above him. Otherwise, the downward hierarchy rule applies (see below)

Here, upward inertia captures the reluctance with which managers call for superiors. Assuming this is not prohibitive, control now passes to this higher level manager, responsible for a (twice as large) market segment of width 2^{n+1}. Within this new span of control, the new manager applies the labor allocation rule. As a result of these re-allocations of labor, the number of penetrations of this (bigger) segment may be very low. This manager has his own T_{min} and T_{max}.

**Downward Hierarchy Rule:** If P < T_{min}, then (with probability 1-downward inertia) the manager is deleted from the structure, and control reverts to his subordinates. Otherwise, the upward hierarchy rule applies (see above).

Here, downward inertia captures the reluctance of managers to cede control or (“disappear”) when they are no longer needed. For expository simplicity, we will assume throughout that the T_{max} of a subordinate equals the T_{min} of a superior, and further, that there is a single value of upward and downward inertia per management level.

That completes the hierarchy specification. Given certain initial labor allocations and environmental dynamics, these simple rules are sufficient to generate hierarchies, and to dissolve them, endogenously. Unlike the standard neoclassical picture, in which management structure is absent and inputs are varied to maximize profit, in this model
inputs are fixed, and it is the management structure that is varied to optimize. Before demonstrating that, we present the alternative approach: internal trade.

*Mechanics of Internal Trade*

Just as before, over the segment they control, managers are computing $P$ (average penetrations over memory) and comparing it to $T_{max}$. Suppose that $P \geq T_{max}$. Rather than invoke the upward hierarchy rule, the manager can “post,” to all other managers in his level, his excess demand for labor: $P - T_{max}$. In turn, those managers will, with probability $1 - \text{horizontal inertia}$, transfer to the posting agent, their excess labor supply (technically, the minimum of that and the poster’s demand). What is this excess supply? It is total labor under the manager’s control minus labor currently intercepting reds, all net of the manager’s own excess demand (his own $P - T_{max}$). In summary:

**Demand Rule:** If $P \geq T_{max}$, then (with probability $1 - h$-inertia) post excess demand, $P - T_{max}$, for $t$ periods.

**Supply Rule:** With probability $1 - h$-inertia, transfer to the posting agent $\min[h\text{ excess demand}, \text{your excess supply}]$.

**Switching Rule:** If, after $t$ periods of posting your excess demand, $P$ still exceeds $T_{max}$, then invoke the upward hierarchy rule.

Here, horizontal inertia, ($h$-inertia) is assumed to be the same in both directions (demand and supply) and would encompass all transaction activities, negotiations, contracts, hoarding, and so on. In a more elaborate version of the model, a literal internal labor market could of course be introduced.¹ But, for our purposes, we will optimize on all values of vertical and horizontal inertia, and leave aside the important question of how those values might be induced in practice--via price or other mechanisms. As we shall see, non-zero inertia will prove optimal given certain objective functions in some environments. The switching rule simply dictates that trade is attempted before resorting to hierarchy. Implicitly, that commits me to the view of hierarchy as a kind of (internal) market failure.

The entire set of parameters and the management layers to which they apply are displayed in Table 1. The numerical values employed in the various Runs are given in Appendix I.

| Table 1. The Organization’s Genome |

The first four management layers possess penetration thresholds, upward and horizontal inertias, and a trading time limit, while the top layer does not. The lowest level of management has no downward inertia, while the top manager does. The entire matrix of

¹ According to Coase (3), however, “the distinguishing mark of the firm is the supersession of the price mechanism.”
parameters can be thought of as a kind of genome for the organization, shown at the bottom of the Table.

Godel Number

As a strictly mathematical matter, one can compress the entire genome into a single integer by the expedient of Godel numbering. Each element of the genome is (or can be converted to) an integer $g_i$. If $L$ is the length of the genome (here 20) then the product of the first $L$ primes, each raised to $g_i$, yields a unique integer, $G$, given by

$$G = \prod_{i=1}^{L} p_i^{g_i}$$

This Godel number encodes the entire adaptive repertoire of the organization, as we shall see. With this apparatus in place, then, let’s put the model through some basic paces, before introducing the objective function required to discuss optimality in any sense.

Part II. Hierarchical Solutions with Internal Trade Clamped Off

To begin, we will ban all internal trade and study the emergence of hierarchy only. As an environment for the first model runs, we posit a heavy opportunity flux (or “attack”) in the south. The resources of the organization (“the defense”), however, are deployed largely in the north. This is depicted in Frame 1 of Run 1. This gross misallocation quickly leads to the southern market segments being over-run, as in Frame 2 of Run 1. To generate the hierarchy, we set penetration thresholds and upward inertia levels to very low levels. To maintain the hierarchy, we set downward inertias to high levels. Heuristically, we can think of low values as zeroes and high values as ones. In that case, the genome of interest is shown in Table 2. The actual numerical parameter values used for all runs are given in Appendix I (and could of course, be normalized to fall in the unit interval).

| Table 2. The Genome for Immediate and Permanent Hierarchy |

Hierarchy emerges quickly as agents in the successive layers record excess penetrations, promptly calling into play ever higher levels of management, as in Frame 3 of Run 1. However, it is not until the very top level of management (level 5) is called into being that an agent (the CEO) has sufficient vision to notice the global north-south misallocation and correct it, as shown in Frame 4 of Run 1.
This adaptation reminds one of a computer company that is focused exclusively on mainframes (the northern market segment), while the opportunity of PCs approaches (the red flux in the south). But it takes a visionary CEO to “see” the strategic error and shift the resources of the firm to the south, exploiting the opportunity.

Of course, having solved the problem, the hierarchy is no longer needed; labor is in place. High level vision is superfluous. But bureaucracies have immense inertia and once constructed are hard to dissolve. This phenomenon is generated by the model’s high level-5 value of downward inertia.

In summary, one recipe for a large persistent hierarchy is as follows: Start with a strategic misallocation of resources. Set penetration thresholds and upward inertias to low values. This grows the hierarchy. High downward inertia then blocks its dissolution. Below, we introduce costs and objective functions for the firm. And, we will see that, given certain objectives, the ability to grow a hierarchy is highly adaptive, while the inability to dissolve it is not.

A more adaptive performance is recorded in Run 2. Here, everything is as before (the genome is exactly as in Table 2), except that downward inertias are zero in all management layers. In this case, hierarchy is again spontaneously generated, and it then solves the strategic problem, exactly as in Run 1. But, having done so, it dissolves, leaving a “lean” structure overseeing a well-allocated work force. The corporate culture here is “When you need help, ask for it (low upward inertia). When you are no longer needed, bow out (low downward inertia).” The upward trajectory is as shown in Frames 1-4 of Run 1 above, but the hierarchy dissolves, leaving the structure shown in Figure 5.

Again, it should be emphasized that the agents are doing all of this from the bottom up without central direction. Each fixed set of agent thresholds and inertias generates an entire history of structural adaptation (and labor re-allocations) to a dynamic environment. In Run 2, the history involves the construction of hierarchy, a strategic reallocation of the organization’s resources, and the dissolution of hierarchy, all “self-organized,” if you will.

In the Runs presented thus far, the agents generate the maximum possible hierarchy (5 levels), which then persists entirely (Run 1) or dissolves entirely (Run 2). In a different environment, fixed settings identical to those of Run 1 (permanent hierarchy) generate a very different history of structural adaptation.

In Run 3, the firm’s resources are initially concentrated in the middle third of the market, while the opportunity flux is advancing in the northern and the southern thirds.
Hierarchy arises, as before, to recognize and correct a strategic misallocation. But, rather than the single maximum five-level hierarchy, the firm consists of two stable local hierarchies of medium height, one in the north and one in the south.

**Observations on the Pure Hierarchy Runs**

The structural adaptations presented thus far unfold without central direction, “from the bottom up,” as a result of the agent rules and parameters in the genome. As just illustrated, any fixed genome will generate different histories of structural adaptation in different dynamic environments. In a sense, the genome “encodes” an entire repertoire of adaptations. Thus far, there are no costs of hierarchy and no objective functions, so no ranking of structures is possible. Before introducing costs and objectives, we explore internal trade as the allocative mechanism, with hierarchy proscribed.

**Part III. Internal Trade**

A pure trade genome is given in Table 3. The salient features are zero horizontal inertia and unlimited time for trade.

<table>
<thead>
<tr>
<th>Table 3. The Genome for Pure Trade</th>
</tr>
</thead>
<tbody>
<tr>
<td>The value of unity for level 1 upward inertia clamps out any hierarchy. The same Run 1 environment and initial misallocation of labor that produced hierarchy and central (top down) reallocation under genome 1 produces a trading solution here. As shown in Figure 4, trade results in the same final correction as in Run 1. Importantly, however, long range reallocation by internal trade is generally slower than under in-place hierarchy, since it requires a sequence of low level, or “local,” managerial responses, rather than the single global reallocation possible under extreme hierarchy.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Run 4. The Run 1 Attack Handled Through Pure Trade</th>
</tr>
</thead>
<tbody>
<tr>
<td>The split attack that resulted in two local hierarchies can also be handled through trade, as demonstrated in Figure 5 (with settings as in the previous run). Notice that global reallocations (i.e., allocation across the entire market front) occur here as well. Indeed, stripped of all economic interpretations, this model concerns tradeoffs between centralized and decentralized approaches to long-range coordination.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Run 5. The Run 3 Split Attack Handled Through Pure Trade</th>
</tr>
</thead>
</table>

A Dynamic Environment

Thus far, the environments have been very straightforward; the flux of incident dots has not changed direction over time. As a final preliminary before introducing the objective function, we explore the performance of pure trade and pure hierarchy in a more complex
environmental dynamic. Here, the organization is confronted with diagonal patterns of incoming opportunities whose slope alternates periodically; it is a sawtooth moving left to right.

Run 6 demonstrates that trade succeeds in intercepting a number incoming reds, but the majority penetrate the market. Trade lags behind this dynamic.

![Run 6. Performance of Pure Trade: Dynamic Environment](image)

By contrast, once the full hierarchy is erected, it is vastly more efficient in blocking penetrations, as shown in Run 7.² (The black lines indicate top-down directives)

![Run 7. Performance of Pure Hierarchy: Dynamic Environment](image)

**General Tradeoffs**

Now, even without introducing any mathematical objective function, one can clearly see that there are tradeoffs between the two pure approaches. The benefit of hierarchy is that it prevents penetrations (protecting market share). However, if salaries increase dramatically with management level, this solution will be expensive. By contrast, pure trade sacrifices market share (allows much penetration) while avoiding the costs of multiple higher levels of management. Given a particular environmental dynamic and cost structure for the firm, then, what levels of penetration and hierarchy actually maximize profit, or market share, or combinations of the two? Equivalently, given a particular objective function, what is the optimal (fixed) genome? What history of structural adaptation does it generate?

**Part IV. Objective Functions**

All numerical assumptions are provided in Appendix I. Over an accounting period \( t \) (e.g., a day) define:

\( R(t) = \text{Revenue} = \text{The Value of Intercepts} = (\text{The Value Per Intercept})(\text{Total Number of Intercepts}). \)

² For fixed memory, trade is slower than hierarchy because long-range re-allocation is effected via a series of myopic applications of \( L \) (where spans are small) versus a single global application of \( L \), where the CEO’s span is global. So, for fixed memory, the relative issue is a sequence of local \( L \)-applications versus a single global application. The absolute gap between the two is a function of memory \( m \). The larger is memory, the slower is trade, because for higher \( m \), the moving average over which \( P \) is computed is ever more stable, so trade lags the sawtooth. By the time \( P \) exceeds the threshold, the environment has moved on. If \( m \) is a tiny number, there are huge adjustments to every environmental blip. In principle, one would optimize on \( m \) as well as the other parameters.
\( W(t) = \text{The Wage Bill} = \sum_i \text{(Number of Managers in Level } i\text{)}(\text{Wage at Level } i) \).

Here, we assume that the wages increase as the cube of the level. Specifically, the wage at level \( i = c(i + 1)^3 \).

\( T(t) = \text{The Transaction Costs of Internal Trade} = (\text{Per Trade Cost})(\text{Number of Trades}) \)

\( P(t) = \text{Penetration Costs} = (\text{The Cost Per Penetration})(\text{Number of Penetrations}) \).

Then, with \( 0 \leq k \leq 1 \), the general per period objective function \( F(t) \) is:

\[
[1] \quad F(t) = R(t) \cdot W(t) \cdot T(t) \cdot kP(t)
\]

If \( k = 0 \), the firm is a profit-maximizer. It doesn’t care about penetrations (i.e., market share). If \( k = 1 \), the firm is a market-share maximizer, and accounts penetrations as costs. If \( k = 0.5 \), the firm cares about both market share and profit. Since net returns depend on the genome \( G \), we think of \( F \) as parameterized by \( G \), and will adopt the notation \( F(t; G) \) to denote the return in period \( t \) for genome \( G \). With the entire above apparatus in hand, we can now address the core question.

**The Central Question**

Specify an environmental dynamic over some time horizon \( T \). For any objective function (\( k \)-value), each fixed genome \( G \)—each vector of thresholds and inertias—will generate some history of structural adaptation, and some corresponding stream of net returns \( F(t; G) \). (Again, the \( G \) entering into this expression as a parameter could be the Godel number for the full vector of thresholds and inertias). The problem then is to find that particular \( G \) which maximizes cumulative returns over the horizon. In short, the problem is to determine:\

\[
[2] \quad G^* = \arg \max \sum_{t=0}^{T} F(t; G)
\]

In a more evolutionary framing, the unit of selection is the genome. Selection pressures are exerted by the specified environmental dynamic over the time horizon \( T \) and the specified objective function \( F(t; \cdot) \). The genome’s fitness is then given by \( \sum_{t=0}^{T} F(t; G) \).

Then [2] asks for the genome with highest fitness.\(^4\)

---

\(^3\) We are actually less interested in \( \sum_{t=0}^{T} F(t; G^*) \).

\(^4\) The computational evidence suggests uniqueness, but is not conclusive.
**Combinatorial Optimization**

Here we have a familiar, if nontrivial, problem in optimization. Our genome is of length 20. Hence if there are four admissible values of each parameter, the space of all genomes is of size $2^{40}$. And the fitness landscape associated with it is rugged. There is a vast literature on the general problem, with techniques including various forms of gradient ascent (“hill climbing”), genetic programming, simulated annealing, and so on. We will take a direct approach suitable for the heuristic problem at hand. First, we will shrink the search space and then survey it by brute force. Specifically, we will assume single enterprise-wide values for each of the four inertia variables (up, down, horizontal supply, horizontal demand). This cuts the space roughly in half and permits a sweep.

The procedure is as follows: For each of the roughly $2^{20}$ genomes, we record cumulative returns over the time horizon (500 periods). For $k=0.0$, $k=0.5$, and $k=1.0$, we return that genome (the vector of parameter values) that maximizes the cumulative objective function $\sum_{t=0}^{T} F(t;G)$. Then we apply those values and “watch” the optimal history of structural adaptation.

**Part V. Results**

Based on earlier runs, we have expectations about the extreme cases of profit maximization ($k=0$) and market-share maximization ($k=1$), so they are presented first.

**Profit Maximizing**

For profit maximization, the optimal genome generates the flat pure trade solution. The enterprise accepts substantial penetration (sacrifices substantial market share) but avoids the costs of hierarchy. This optimal history of structural adaptation is already shown in Run 6. Note again the way trade lags behind the environment and the high level of yellow penetrators.

**Market Share Maximizing**

For market share maximization, the optimal genome *immediately* generates the maximum hierarchy\(^5\). Here, profit is sacrificed (costs are ignored) to avoid penetrations. The optimal upward inertia is literally zero. Since the upward hierarchy rule generates

---

\(^5\) Here, and in the oscillating organizations below, hierarchies are generated. The question arises, where do these managers come from? One answer is that they come from the market for managers--and in the model they’re being paid a very high price to join (the level cubed). It’s a partial equilibrium picture that does not depict the spot market for managerial talent, but in principle, that is where they come from.
hierarchy when \( p \geq 0 \), it generates hierarchy before any reds are even in view. In this respect, it differs from Run 7. Otherwise, it is exactly as in that Run. The first two frames, however, are of interest, and are shown in Figure 8.

![Figure 8 Market-Share Maximizing: Immediate Hierarchy](image)

In a sense, this explains the extreme hierarchies observed in military organizations (and perhaps even their peacetime enormity\(^6\)). If the objective is to dexterously allocate forces across a fluid battlefield, “top down command” makes more sense than holding a tank auction. (The same point applies to disaster relief, emergency room operations, etc.).\(^7\)

These results are in the nature of sanity tests for the model. They are the results we expect, and they were largely anticipated in the earlier runs. What of the hybrid case of \( k=0.5 \)? Here, the enterprise cares about both profit and market share. If profit maximization is flat (no hierarchy) and market-share maximization is the maximum hierarchy, what would one expect for the hybrid case? Intuition would suggest an intermediate hierarchy.

*Hybrid Objective*

Quite to my surprise, the optimal genome for the \( k=0.5 \) case, generates no fixed geometry at all. The optimal adaptive organization has a *variable geometry*. This variable geometry firm oscillates between the flat trade regime and a hierarchy of intermediate height. Spatially, the oscillating structure “chases” the sawtooth environment wave up and down the front over time. This “traveling wave” organization has amplitude (maximum hierarchy) 4 and period 10. Run 8 shows this optimal adaptive performance.

![Run 8. Optimal Adaptive Oscillations](image)

The second best genome looks very much like the winner. But the third place genome differs in an interesting way. It, too, is an oscillator. But it has lower frequency (30 periods) and higher amplitude (5), building and then dissolving hierarchies of the maximum height possible, as shown in Run 9.

![Run 9. Third Place Oscillating Firm](image)

Time series of the first and third place organizations’ oscillations are shown in Figure 9.

![Fig 9. Oscillation Time Series Compared](image)

Unconstrained maximization of the hybrid (\( k=0.5 \)) is different in principle from constrained profit maximization. This, too, was studied and generates yet a third, different, oscillating solution.\(^8\)

---

\(^6\) Obviously, this also has a great deal to do with political and industrial interests.

\(^7\) Relatedly, see Weitzman (11).

\(^8\) On dynamic organizational forms, see (8, 10).
Toward a Generative Definition of “Design”

Now, consider the question, “What is the winner’s ‘design’?” Surely, it is not a particular structure, since—given the winner’s genome—the structure changes in a dynamic environment. But, it is not a particular sequence of structures either, since a different dynamic environment will yield a different adaptive structural history. Rather, the winner’s design can only be: the complete adaptive repertoire generable by (encoded in) the enterprise’s genome. And the genome encodes a huge adaptive repertoire, realized differently in different environments, just as different selection pressures induce different realizations of a species’ fixed genomic endowment.

Part VI. Extensions and Future Research

As an initial demonstration of the approach and of its potential for producing counterintuitive results, perhaps the foregoing exposition is sufficient. But, it leaves a great many issues unaddressed. Among the more obvious ones are robustness, mechanism design, imperfect decision-making, and multiple firms.

Robustness

How sensitive are the optimization results to variations in assumptions about manager compensation, transaction costs, and other numerical parameters? Implicitly, the cost of collapsing a hierarchy is zero, when clearly, there are some associated costs. Are there cost levels that would fundamentally alter the relationship between trade and hierarchy?

How sensitive are the optimization results to variations in assumptions about the environmental dynamic? We optimized given a single dynamic environment, the sawtooth. But organizations often face different dynamic environments. One could confront organizations with a range of environmental dynamics (e.g., spike, square wave, and sawtooth) and ask for the genome that maximizes the sum or average of returns over those.

The Pyramid Puzzle

In this general connection, the above analysis suggests that hierarchy is effective for solving long-range re-allocation problems that can suddenly spike in volatile environments. But, why is hierarchy ubiquitous in stable settings? Behavioral psychology suggests that humans often (a) place asymmetric weight on losses (as against gains) and (b) confuse probabilities with expected values (over-rating risks). One conjecture that might explain the ubiquity of pyramids it that we over-rate the loss from unanticipated shocks, and then over-rate their probability, thus maintaining a huge over-capacity (hierarchy) to cope (top-down) with rare environmental spikes. In any event, one extension would be to rigorously characterize the circumstances (environment,
objective function) under which permanent hierarchy does emerge as optimal. This would be a type of inverse problem.

**Mechanism Design**

Having determined (by brute computational force), the optimal propensities to hoard and transfer labor, the economists issue becomes: What mechanism would induce the optimal behavior? What incentive structure would induce rational actors to perform optimally. This brings in a huge literature on mechanism design, markets, and games. Modeling approaches doubtless abound. One thought is to explore a variation on classical wage determination. Rather than pay agents their marginal revenue product (MRP), suppose they were paid some convex combination of all the MRPs in their management layer. Rational agents would then have a vested interest in the productivity (intercepts) of others, mitigating the inefficiency of hoarding.

**Imperfect Decision-Making**

One reason hierarchy allocates effectively is that CEOs are assumed (like everyone else) to follow the labor allocation rule L to perfection, putting unoccupied labor precisely where it is needed. If one were to assume stubborn hidebound CEOs, “sticking to their guns” despite misallocation, hierarchy might fare poorly indeed. Adding noise to the information available at each layer—possibly having it increase with each layer—might also enrich the tale substantially. It might also prove to be necessary if there were an empirical exercise. Another obvious extension would be to allow CEOs to add labor.

**Multiple Firms**

In the formulation above, firms face an environment, but they do not face other firms. Most firms would probably regard other firms as part of their environment. I introduce multiple firms as follows. Simply imagine firms in a “ring.” Opportunities not intercepted by firm n pass through and become opportunities for firm n+1, as illustrated for two firms in Figure 10.

![Figure 10. Two Competing Firms](image)

The order in which firms play could be randomized each cycle. New opportunities can be fed in continuously or not.

An illustrative two-firm dynamic case where there are no new opportunities (after 50 cycles) is recorded in Run 10. The competitors exhibit different adaptive histories. The right firm remains flat throughout, while the left firm erects the maximum hierarchy
(Frame 3) and later dissolves it (Frame 5), by which point the two firms have essentially divided the market.

**Run 10. Two Firms**

A different experiment would be to “peg” one firm to some strategy, and sweep the parameter space for strategies that will defeat it. Milton Friedmann (6) famously argued that the assumption of profit-maximization was warranted on the evolutionary grounds that firms deviating from it would ultimately be selected out. To explore this, one could lock one firm onto profit maximization (our $k=0$ case) and see whether its competitor (the parameter sweep) could “discover” the strategy of operating at a loss in order to monopolize market share in the short term, driving the first firm out of business, and then “relaxing” into a more profitable strategy having killed off the competition. Of course, it would be natural to co-evolve strategies in the general case.

**Summary**

We developed an agent-based model in which hierarchies and internal trading regimes emerge endogenously. And, we generated a variety of structural adaptations in a range of contrived dynamic environments. We introduced a rather general objective function for the enterprise and for a variety of special cases (profit-maximizing, market-share maximizing, and hybrid), we determined the optimal genome for a particular test environment. Applying that genome, we then generated, and depicted graphically, the optimal history of structural adaptation to the test environment. This winning history was, to me, quite unexpected, involving oscillating “flat” trading regimes and hierarchies of intermediate height in perpetual motion up and down the spatial market as a travelling wave: a variable geometry firm. A number of extensions and directions for future research were discussed.
APPENDIX I. Numerical Assumptions and Parameter Definitions

Table A1: Numerical Assumptions

<table>
<thead>
<tr>
<th>Run</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>Up</th>
<th>Down</th>
<th>Give</th>
<th>Ask</th>
<th>Formation</th>
<th>Switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.7</td>
<td>1.4</td>
<td>2.0</td>
<td>4.0</td>
<td>10</td>
<td>20</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.7</td>
<td>1.4</td>
<td>2.0</td>
<td>4.0</td>
<td>10</td>
<td>20</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0.7</td>
<td>1.4</td>
<td>2.0</td>
<td>4.0</td>
<td>10</td>
<td>20</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.7</td>
<td>1.4</td>
<td>2.0</td>
<td>4.0</td>
<td>10</td>
<td>20</td>
<td>10</td>
<td>1</td>
<td>0.25</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0.7</td>
<td>MAX</td>
<td>MAX</td>
<td>MAX</td>
<td>MAX</td>
<td>20</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>0.7</td>
<td>MAX</td>
<td>MAX</td>
<td>MAX</td>
<td>MAX</td>
<td>20</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>0.7</td>
<td>1.4</td>
<td>2.0</td>
<td>4.0</td>
<td>5.0</td>
<td>20</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>0.5</td>
<td>1.0</td>
<td>3.0</td>
<td>4.0</td>
<td>5.0</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>0.0</td>
<td>1.0</td>
<td>1.5</td>
<td>4.0</td>
<td>0.0</td>
<td>8</td>
<td>8</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>0.5</td>
<td>1.0</td>
<td>1.5</td>
<td>2.0</td>
<td>4.0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>0.0</td>
<td>1.0</td>
<td>3.0</td>
<td>4.0</td>
<td>2.0</td>
<td>0</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
</tbody>
</table>

* Where no value is given (-), the value is irrelevant.

**Key**
- T1 – T5: Thresholds 1 – 5
- Up/Down: Up and Down Inertias (Vertical)
- Give/Ask: Prob Ask and Give Labor (Horizontal)
- Formation: Attack Formation
- Switch: Attack Formation Switch Probability
- Max: Some large number (i.e. 20)

**Fixed Values**
- Cost per penetration: 100.
- Salary for Rank 0: 1
- Salary for inactive management: rank.
- Salary for active management: (rank + 1)^3.
- Cost per trade: 5
- Profit per intercept: 200
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Active Rank</td>
<td>1</td>
<td>Highest ranking manager who is active</td>
</tr>
<tr>
<td>Attack Formation</td>
<td>1</td>
<td>Determines the environment (the approach the attackers take)</td>
</tr>
<tr>
<td>Attack Formation Switch Probability</td>
<td>0.4</td>
<td>Probability of switching directions of certain attacks</td>
</tr>
<tr>
<td>Attacker Speed</td>
<td>1</td>
<td>Number of spaces attackers move per iteration</td>
</tr>
<tr>
<td>Cost Per Penetration</td>
<td>1</td>
<td>Cost to profit function of a successful attack</td>
</tr>
<tr>
<td>Cost Per Trade</td>
<td>5</td>
<td>Cost to profit function of trading labor</td>
</tr>
<tr>
<td>Down Inertia</td>
<td></td>
<td>Number of periods a manager's $p$ must be below $t$ for it to go inactive</td>
</tr>
<tr>
<td>History Length</td>
<td>10</td>
<td>The number of periods over which average penetration and intercepts is calculated</td>
</tr>
<tr>
<td>Management Salary Multiplier</td>
<td>10</td>
<td>Multiplier of management's rank cubed to determine salary</td>
</tr>
<tr>
<td>Msk</td>
<td>0</td>
<td>K-constant in profit function</td>
</tr>
<tr>
<td>Nearness Line of Sight</td>
<td>TRUE</td>
<td>Reserved</td>
</tr>
<tr>
<td>Prob Ask For Labor</td>
<td>1</td>
<td>Probability a manager will ask others of its rank labor</td>
</tr>
<tr>
<td>Prob Create Attacker</td>
<td>0.7</td>
<td>Probability an attacker is created</td>
</tr>
<tr>
<td>Prob Create Front Line Defender</td>
<td>15</td>
<td>Probability a front-line defender is created</td>
</tr>
<tr>
<td>Prob Give Labor</td>
<td>1</td>
<td>Probability a manager will give others of its rank for labor</td>
</tr>
<tr>
<td>Radius</td>
<td>1</td>
<td>Reserved</td>
</tr>
<tr>
<td>Random Edge Ratio</td>
<td>0</td>
<td>Reserved</td>
</tr>
<tr>
<td>Size</td>
<td>5</td>
<td>Reserved</td>
</tr>
<tr>
<td>Size</td>
<td>2</td>
<td>Reserved</td>
</tr>
<tr>
<td>Size</td>
<td>31</td>
<td>Reserved</td>
</tr>
<tr>
<td>Size</td>
<td>3</td>
<td>Reserved</td>
</tr>
<tr>
<td>Threshold Rank 1</td>
<td>0.7</td>
<td>Threshold of managers of rank 1</td>
</tr>
<tr>
<td>Threshold Rank 2</td>
<td>1.4</td>
<td>Threshold of managers of rank 2</td>
</tr>
<tr>
<td>Threshold Rank 3</td>
<td>2</td>
<td>Threshold of managers of rank 3</td>
</tr>
<tr>
<td>Threshold Rank 4</td>
<td>4</td>
<td>Threshold of managers of rank 4</td>
</tr>
<tr>
<td>Threshold Rank 5</td>
<td>10</td>
<td>Threshold of managers of rank 5</td>
</tr>
<tr>
<td>Up Inertia</td>
<td>10</td>
<td>Number of periods a manager's $p$ must be above $t$ for it to go inactive</td>
</tr>
<tr>
<td>Value Per Intercept</td>
<td>200</td>
<td>Revenue garnered from each intercept</td>
</tr>
</tbody>
</table>
References


Growing Adaptive Organizations: An Agent-Based Computational Approach

Appendix II. Tables and Graphics

Joshua M. Epstein*

February 20, 2003 Draft

* The author is a Senior Fellow in Economic Studies and in Governance Studies at The Brookings Institution, a Member of the Brookings-Johns Hopkins Center on Social and Economic Dynamics, and a Member of the External Faculty of The Santa Fe Institute.
Figure 1: Initial Enterprise and Environment

Figure 2: Penetrators

Figure 3. The Maximum Hierarchy
Figure 4: The Essential Problem

Table 1. The Organization’s Genome

<table>
<thead>
<tr>
<th>Hierarchy:</th>
<th>Relevant Management Layers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Penetration Threshold $t$</td>
<td>1  2  3  4  ___</td>
</tr>
<tr>
<td>Upward Inertia     $u$</td>
<td>1  2  3  4  ___</td>
</tr>
<tr>
<td>Downward Inertia   $d$</td>
<td>___  2  3  4  5</td>
</tr>
<tr>
<td>Trade:</td>
<td></td>
</tr>
<tr>
<td>Horizontal Inertia $h$</td>
<td>1  2  3  4  ___</td>
</tr>
<tr>
<td>Trading Time Limit  $l$</td>
<td>1  2  3  4  ___</td>
</tr>
</tbody>
</table>

$G=\{t_1, t_2, t_3, t_4, u_1, u_2, u_3, u_4, d_2, d_3, d_4, d_5, h_1, h_2, h_3, h_4, l_1, l_2, l_3, l_4\}$
Table 2. The Genome for Immediate and Permanent Hierarchy

<table>
<thead>
<tr>
<th>Relevant Management Layers</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hierarchy:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Penetration Threshold $t$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>___</td>
</tr>
<tr>
<td>Upward Inertia $u$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>___</td>
</tr>
<tr>
<td>Downward Inertia $d$</td>
<td>___</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>1</td>
</tr>
<tr>
<td><strong>Trade:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Horizontal Inertia $h$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>___</td>
</tr>
<tr>
<td>Trading Time Limit $l$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>___</td>
</tr>
</tbody>
</table>

$G = \{0,0,0,0,0,0,0,0,#,#,#,#,1,0,0,0,0,0,0,0,0\}$

Note: A hatch mark denotes “irrelevant.” To ensure that our Godel numbering is consistent, we could legislate a single constant for all such occurrences.

Run 1: Emergence of Permanent Hierarchy
Figure 5: Emergence and Dissolution of Hierarchy: Final State

Run 3: Local Intermediate Hierarchies

Frame 1:

Frame 2:
Table 3: The Genome for Pure Trade

<table>
<thead>
<tr>
<th>Relevant Management Layers</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Penetration Threshold</td>
<td>$t$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Upward Inertia</td>
<td>$u$</td>
<td>1</td>
<td>#</td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>Downward Inertia</td>
<td>$d$</td>
<td>___</td>
<td>#</td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>Trade:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Horizontal Inertia</td>
<td>$h$</td>
<td>0</td>
<td>#</td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>Trading Time Limit</td>
<td>$l$</td>
<td>1</td>
<td>#</td>
<td>#</td>
<td>#</td>
</tr>
</tbody>
</table>

$G=\{0,0,0,1,\#,\#,\#,\#,\#,\#,\#,0,\#,\#,\#,1,\#,\#,\#\}$
Run 4: The Run 1 Attack Handled Through Pure Trade

Frame 1

Frame 2

Frame 3
Run 5: The Run 3 Split Attack Handled Through Pure Trade
Run 6: Performance of Pure Trade: Dynamic Environment
Run 7: Performance of Pure Hierarchy: Dynamic Environment
Figure 8: Market-Share Maximizing: Immediate Hierarchy (k=1)
Run 8: Optimal Adaptive Oscillations
Run 9: Third Place Oscillating Firm

Frame 1

Frame 2

Frame 3
Figure 9: Oscillation Time Series Compared

Optimal
Third Place

Fig. 10: Two Competing Firms
Ring of Competing Firms

Opportunities missed by firm j (yellow) are opportunities (reds) to firm j+1

Firm j  Firm j+1

Run 10: Two Firms

Frame 1:
Final frame of movie – steady state