Hierarchies and Information-Processing Organizations

LI, HAO[1]

The University of Hong Kong

November 12, 1998

This paper analyzes organizational structures that minimize information processing costs for a specific organizational task. Organizations consist of agents of limited ability connected in a network. These agents collect and process information, and make decisions. Organizations implement strategies—mappings from environmental circumstances to decisions. The strategies are exogenously given from a class of "pie" problems to be defined in this paper. The notion of efficiency is lexicographic: the primary criterion is minimizing the number of agents, and the secondary criterion is minimizing the number of connections between the agents. In this modeling framework, efficient organizations are not hierarchical for a large number of problems. Hierarchies often fail to exploit fully the information processing capabilities of the agents because in a hierarchy, subordinates have a single superior.

[1] This paper is part of the first essay of my dissertation. Financial support from the Alfred P. Sloan Foundation Dissertation Fellowship is gratefully acknowledged. The author would also like to thank helpful comments from In-Koo Cho, Michael Chwe, Sherwin Rosen, Lars Stole, seminar participants at the University of Chicago and at the University of Pennsylvania, and especially the Associate Editor and two anonymous referees.

1. Introduction

The study of organizational structures in economic literature has concentrated on a special form, the hierarchy. The idea that economic organizations, such as firms, are organized hierarchically was hinted at by Coase (1937), who first argued that firms exist because there are costs of using the market system. Whereas markets organize buying and selling through the price mechanism, coordination in an economic organization is thought to be carried out hierarchically through authority and supervision. There have been numerous works on the design of optimal hierarchies, where the central issue is to determine the span and the depth to minimize loss of control.[2] The presumption of hierarchical organizational structures constitutes a common characteristic in this line of research.

Organizational design problems can be analyzed from a useful angle by focusing instead on the role of information processing. Strategic decision making may require complicated information processing, but individual agents have only limited capabilities. Decision makers of an organization often rely on information provided by their subordinates, who in turn may rely on their own subordinates. This form of "decentralized" information processing can give rise to an organizational structure distinct from the structure of authority and supervision.

This paper analyzes organizational structures that minimize information processing costs for a specific organizational task. Agents of the information-processing organization are assumed to carry out "linear classification," which transforms a vector of real variables into a binary variable by computing a weighted sum and comparing it to a reference number. Only the result of the comparison is reported to superiors. A more capable agent has more reference numbers and can classify information into more categories. Mappings from the circumstances to the decisions are called "strategies." The goal of organizational design is to construct a network of information processing that implements a given strategy (making the desired decision under each circumstance). Since there can be many organizations that achieve this, organization designer naturally chooses on the basis of "efficiency."

[2] See, e.g., Williamson (1967) and Keren and Levhari (1989).

When all agents of the organization have the same information processing capability, an efficient organization minimizes employment. More generally, if more capable agents can be hired at greater wages, efficiency means minimizing the wage bill.

Organizational tasks are assumed to arise from a special class of strategies. One way to generate these strategies is through the following consultation process. A binary decision "yes" or "no" has to be made based on binary judgments of some experts. The judgments of experts are correlated in the sense that for any circumstance there exist two experts whose judgments determine the corresponding decision. Final decision thus requires identifying the two experts and their judgments. The space of the circumstances is two-dimensional and the experts are consulted in order. Expert 1 is asked for a judgment first, and then expert 2. If expert 2 agrees with expert 1, expert 3 is consulted. The decision depends only on the identity of the first expert whose judgment differs from those of his predecessors and the two judgments. This class of strategies has a simple geometric representation. Each expert can be viewed as a line in the two-dimensional space of circumstances who reports on which side a given circumstance lies. All lines pass through a single point in the space. This geometric interpretation motivates the name "pie strategies."

The main conclusion of this paper is that hierarchies are often inefficient in implementing pie strategies relative to alternative organizational forms. A defining characteristic of real-life hierarchies is accountability. In the context of information-processing organization, hierarchies can be defined as networks where a single superior is responsible for handling the information provided by any agent.[3] It is shown that any strategy can be implemented by a hierarchy, but this can be inefficient because information processing tasks performed by distinct agents in the hierarchy can be performed equally well by a single agent who reports to more than one superior. Moreover, when the number of experts becomes large, the efficient structures of information processing are typically non-hierarchical. Thus, hierarchies often fail to exploit fully the information processing capabilities of the agents because of the hierarchical principle of single superior.

[3] The notion of the hierarchy in organizations is quite complicated and has been given different meanings in the economic literature. The definition in the present paper emphasizes the single superior property by identifying it with the tree structure. See Radner (1992) for a broad discussion of the hierarchy.

Although pie strategies are special, the analysis suggests that the main result that hierarchies can be inefficient in information processing holds more generally. The intuition behind this result is that when an expert's judgment is critical in more than one circumstance and has different implications for the final decision in different circumstances, this "critical" expert must pass along his judgment to more than one immediate superior. For example, consider a computer firm that has to decide whether or not to invest in a software upgrade. The firm sells both software and hardware, and the investment is profitable in the short run only if, among other conditions, the upgrade and the firm's hardware product are complements (otherwise the firm is better off keeping its current version of software). Then, the estimate of future demand for the software upgrade from the firm's software market researcher is critical for the investment decision both when the estimate of future demand for the hardware is high and when it is low. A high estimate for the software upgrade is good news for the investment if future demand for the hardware is also high, but bad news if the latter is low. Similarly, a low estimate for the upgrade is bad news if demand for hardware is high, but good news if the latter is low. In this situation, the report from the software market researcher must be passed to the head of the hardware division and as well as to the head of the software division. Otherwise, the crucial information content in the report may get lost when it is aggregated by the head of the software division with other reports. Identifying strategies with critical experts is made easy by the restriction to pie strategies, but it is also possible for more general problems. Suggestions are given in section 4 on how to extend the result that hierarchies can be inefficient in implementing pie strategies to a more general binary decision problem. A broad conjecture, based on analysis of pie strategies, is that in modern organizations where the explosion of information sources has by far outstripped the improvement of individual capabilities, hierarchies are seldom the efficient information-processing organizational structure.

The conclusion that hierarchies are often inefficient in implementing pie strategies contrasts with the well-known result in the theory of parallel processing that tree structures are efficient for computing associative operations in "batch mode." (see, e.g., Radner, 1993). This contrast arises mainly from the difference between associative operations and threshold functions. The threshold functions examined in the present paper emphasize a particular type of information processing tasks—classification and recognition of circumstances. In real-world organizations, information processing tasks typically involve different types of operations. Indeed, as argued by Radner (1992), different decisions often rely on different information-processing networks within the same organization. The result of inefficiency of the hierarchy contributes to the understanding of the relation between efficient organizational structures and information processing tasks.

The next section provides a general description of the model of information-processing organizations discussed in this paper, including notions of linear classification, strategy, hierarchy, and an efficiency criterion. Section 3 introduces pie strategies and characterizes the efficient organizations. It is shown that an efficient organization can be a two-rank hierarchy, a three-rank hierarchy, or a three-rank non-hierarchy, depending on the strategy to be implemented. The number of middle managers is at most two in a three-rank organization. The proportion of strategies that can be efficiently implemented by a hierarchy is small when the number of experts is large. However, if more capable decision-makers can be hired at relatively low costs, efficient organizations are always two-rank hierarchies. The last section discusses related works and suggests how to extend some of the results to more general design problems. Proofs of the most of the propositions can be found in the appendix.

2. General Description of the Organization

2.1. Basic information processing function

DEFINITION 2.1. A function $g(b)$ of J binary variables, each of which b_j $(j = 1, \dots, J)$ takes a value of 1 or 0, is a single-threshold function if there is a weight vector $w = (w_1, \dots, w_J)$ and a threshold number θ such that

$$g(b) = \begin{cases} 1 & \text{if} \quad \sum_{j=1}^{J} w_j b_j \geq \theta \\ 0 & \text{if} \quad \sum_{j=1}^{J} w_j b_j < \theta. \end{cases}$$

The single-threshold function is widely used in various computing models. For example, it forms the basis of the perceptron (see Minsky and Papert, 1988, or Weisbuch, 1990), which is an early attempt at modeling the human brain. Here it is chosen to represent the basic information processing unit in the organization. In performing a single-threshold function, the unit aggregates J information sources into a single summary statistic. It is simple to characterize, and yet quite flexible because both the weight vector and the threshold can be modified.

The information processing capability of the single-threshold function is limited because it can divide the weighted sum of J information sources into only two intervals. This limitation will be the driving force of the results in this paper. Until section 3.4, all agents of the organization are assumed to have the information processing capability of the single-threshold function, or simply the threshold function. A more capable information processing unit can divide the weighted sum into three or even more intervals.[4] Double-threshold functions are defined below. Threshold functions with more thresholds can be similarly defined.

DEFINITION 2.2. A function $g(b)$ of J binary variables is a double-threshold function if there is a real vector w and two real numbers θ^1 and θ^2 such that

$$g(b) = \begin{cases} 0 & \text{if} \quad \theta^1 \le w \cdot b \le \theta^2 \\ 1 & \text{otherwise.} \end{cases}$$

2.2. Strategy and organization

A "strategy" is a mapping Y from a set of circumstances, S, to a set of decisions. S can be either finite or infinite. The following assumption is maintained throughout the paper.

(A1) The set of decisions has only two elements.

These two elements are called "true"(t) and "false"(f). The goal of organizational design is to use the basic information processing functions to structure a network that implements a given strategy efficiently. Networks are defined first; the notion of efficiency will be introduced later.

[4] It is implicitly assumed that a threshold operation is more important than addition in measuring information processing capability. Rubinstein (1992) uses a similar definition to study price discrimination by a monopolist who sells to consumers with limited abilities of perceiving prices.

There are three types of agents in the organization. The first type of agent is called "experts." Each expert, denoted as e, is a "linear classifier" defined as follows.

DEFINITION 2.3. A function $g(b)$ of J variables is a linear classifier if there is a weight vector w and a threshold θ such that

$$g(b) = \begin{cases} 1 & \text{if} \quad w \cdot b \geq \theta \\ 0 & \text{otherwise.} \end{cases}$$

A linear classifier is a generalized threshold function. The output of a linear classifier is a binary variable; the input allows some flexibility. For example, suppose that S is the two-dimensional Euclidean space and the two inputs of the linear classifier are x and y coordinates. The linear classifier divides S into two pieces by a line, and classifies each circumstance in S according to its position relative to the line.

The second type of agent is called "decision maker," denoted as d. The input of d is a vector of binary variables; the output is a decision t or f. Formally, a "decision function" is defined as a composite function of a threshold function and a mapping from a binary variable to $\{t, f\}$.

The third type of agent is called "middle manager," denoted as m. A middle manager performs a threshold function, whose input and output are both binary variables. By this definition, a middle manager can not be an agent that directly collects information from S, or an agent that makes the final decision.

The expert e, the decision maker d, and the middle manager m are generally called agents of the organization and denoted as a. The information processing functions they perform differ only in terms of the input or output. Denote the set of all agents as A. The following restriction on the organization follows partly from the definitions of the three types of agent, and is made to facilitate exposition.[5]

(R1) There is at least one e in A; there is only one d in A.

Given the set of agents, the first step in forming a network is to define a binary relation in A. Formally, a binary relation among agents in A is called "superior to" if it satisfies:

[5] Although there is only one decision to be made for each circumstance, it does not follow directly that there should be a fixed single decision maker for all circumstances. Restriction (R1) will be discussed in section 3.1.

[1] (transitivity) a_2 is superior to a_1 and a_3 is superior to a_2 imply that a_3 is superior to a_1; [2] (anti-symmetry) a_2 is superior to a_1 implies that a_1 is not superior to a_2. Denote the superior-to relation as $\prec$. If a_2 is superior to a_1, or $a_1 \prec a_2$, and if there is no a_3 in A such that $a_1 \prec a_3$ and $a_3 \prec a_2$, then a_2 is an "immediate superior" to a_1. The relation between a_1 and a_2 is called a "connection," and a_1 is said to "report to" a_2. Two agents are incomparable if neither of the two is superior to the other. The following restriction on the organization is implied by (R1) and the definitions of three types of agents.

(R2) There is a single agent d in A that is superior to all other a in A; for any m, there exists e such that $e \prec m$.

An "organization" is defined as a pair $(A, \prec)$ that satisfies restrictions (R1) and (R2). An organization defines a strategy, as it maps each circumstance in S to a decision. Alternatively, the organization is said to "implement" the strategy.

2.3. Structuring the organization

Consider the general form of an organization $(A, \prec)$. By the definition of $\prec$, the organization has no "loops." Thus, for any e in A, an "authority line" can be defined as a collection of k agents $e, a_2, \ldots, a_{k-1}, d$ such that $e \prec a_2 \prec \ldots \prec a_{k-1} \prec d$. A useful way of thinking about the organizational structure is to define a "ranking system." Formally, a ranking system of $(A, \prec)$ is an assignment of a positive number $r(a)$ to each a in A such that: [1] $r(a_1) < r(a_2)$ if $a_1 \prec a_2$; [2] a_1 and a_2 are incomparable if $r(a_1) = r(a_2)$. The lemma below formally states that $(A, \prec)$ can be ranked.

LEMMA 2.4. *There exists a ranking system for* $(A, \prec)$.

PROOF. Let R be the number of agents in the longest authority line of $(A, \prec)$, and denote the longest authority line as $e' \prec a'_2 \ldots \prec a'_{R-1} \prec d$. Assign a rank 1 to all e's in A. Consider the set of the agents each of whom is an immediate superior to some e. If any a in this set is a superior to another agent in the set, exclude a and all its superiors from the set. The resulting set is non-empty because it includes a'_2. Moreover, any two agents in the resulting set are incomparable. Assign a rank 2 to all the agents in the set. Then consider the set of agents each of which is an immediate superior to some a of rank

2. Repeat the above exclusion process. The process stops after $R-1$ rounds. By the definition of the longest authority line, d is the immediate superior to all agents of rank $R-1$, which must include a'_{R-1}. Each agent in A is ranked exactly once. *Q.E.D.*

Organizations satisfying (R1) and (R2) are called "feed-forward" networks by computer scientists (see, e.g., Muroga, 1971) because information is fed forward: if information is transmitted from a_1 to a_2, then $r(a_1) < r(a_2)$ for any ranking system r of the network. In a feed-forward network, information flows in one direction only. There is no feed-back in the network. As a model of information processing, the feed-forward network focuses on information aggregation and ignores the reverse process. The left diagram in Figure 2.1 illustrates a feed-forward network.

[Insert Figure 2.1 Here]

Since $\prec$ is not a complete ordering of A, there can be many ranking systems for a given organization. The ranking system constructed in the proof of Lemma 2.4, however, is uniquely defined. Denote this ranking system as R. In what follows, "rank" of any a in A refers to $R(a)$.

There is one feature of the ranking of the network in the left diagram of Figure 2.1 that is incongruous with ordinary language of organizational design. In the ordinary language, if an agent is an immediate superior to another agent, the difference between the ranks of the two is 1. A ranking system with such property is called a "regular" ranking system. The network in the left diagram of Figure 2.1 has a rank 3 agent that is an immediate superior to both a rank 2 agent and a rank 1 agent. A regular ranking system for this network cannot be defined. The following restriction is clearly both necessary and sufficient for a regular ranking system to exist.

(R3) For any a_1 and a_2 such that a_2 is an immediate superior to a_1, $R(a_2) = R(a_1) + 1$.

Organizations satisfying (R3) are called "feed-next" networks by computer scientists. A feed-next network is a special case of feed-forward networks, with the property that information can only be transmitted from one agent to another one of the next higher rank. Any feed-forward network can be replaced by a feed-next network with the same number of ranks that implements the same strategy. This can be done by adding to any connection violating (R3) a middle manager that simply relays information. The network in the right diagram of Figure 2.1 shows a feed-next network that can implement the same strategy as the feed-forward network in the left diagram. Until section 3.4, organization means a feed-next network. Hierarchies are defined next.

DEFINITION 2.5. A hierarchy is a network where each agent except for the decision maker has exactly one immediate superior.

The special feature of the hierarchy is that each agent is connected to only one superior. Indeed, for any given A, hierarchies have the smallest number of connections among all possible structures. The networks shown in Figure 2.1 are not hierarchies, because there is a rank 1 agent with two immediate superiors. It is the main characteristic of the "matrix organization" or "matrix management" that some agents of the organization report to more than one superior.[6] In the framework of this paper, matrix organizations are the alternative to hierarchies.

2.4. An efficiency criterion

The goal of organizational design is to use the generalized threshold functions described in section 2.2 to structure a network that implements a given strategy "efficiently." A natural measure of efficiency is the number of agents in the organization, because all the agents perform some kind of threshold functions. However, comparing the number of agents does not in general result in a complete ordering among organizations that implement the same strategy. For example, consider a multi-rank hierarchical organization. One can connect any agent a_1 to some a_2 of the next rank that is not its immediate superior. This new connection gives a_2 an additional input variable. A weight for this input can

[6] See Galbraith (1971) for an informal analysis of matrix organizations. The matrix organization was developed in 1960's by multi-divisional firms in an attempt to improve coordination among related product divisions. In a matrix organization, the directors of the two sales departments of two products, say, report to a common superior (coordinator), as well as to the division heads in the respective divisions. Despite its initial popularity, the matrix organization gradually fell out of fashion in the eighties due to various control problems it created within the organization.

be found such that a_2 performs the same threshold function as before, and the same strategy is implemented with one additional connection between a_1 and a_2. This additional connection is of course unnecessary, and it adds to the information processing task of a_2. Thus, one may also want to compare the number of connections among the organizations that implement the same strategy.

The number of agents provides the number of threshold functions necessary to implement a strategy. The number of connections, on the other hand, depends on both the number of threshold functions and the average number of input variables in each threshold function. The number of threshold functions in the network is more important as a measure of efficiency, because the information processing capability of an agent is identified with the number of thresholds in the function he performs rather than the number of additions in the function. Let $C(A, \prec) = \{a_1, a_2 \in A : a_1 \prec a_2 \text{ and } R(a_2) = R(a_1) + 1\}$ be the connections in $(A, \prec)$.

DEFINITION 2.6. $(A_1, \prec_1)$ is more efficient than $(A_2, \prec_2)$ for a strategy Y if the following is true:

[1] $(A_1, \prec_1)$ and $(A_2, \prec_2)$ implement Y;

[2] $|A_1| < |A_2|$, or $|A_1| = |A_2|$ and $|C(A_1, \prec_1)| < |C(A_2, \prec_2)|$.

The organization $(A_1, \prec_1)$ is efficient for strategy Y if no other organization is more efficient than $(A_1, \prec_1)$ for Y.

According to the first condition in the above definition, two organizations are compared only when they implement the same strategy. Thus, in designing a network, the first goal is to implement a given strategy—make the right decisions under all circumstances. The strategic capacity of the organization is given paramount importance. Consideration of efficiency comes only after. To see an implication of this, imagine that organizations compete with one another, each organization aiming to maximize its own payoff. When payoff maximization requires the organization to adopt relatively "sophisticated" strategies, i.e. strategies that entail fine classification of the state space S, there is a trade-off between the payoff and the cost of information processing. Sophisticated strategies in general require more experts and middle managers to assist the decision making process. The above definition of efficiency then implies that this trade-off is lexicographical: the cost of information processing is of secondary importance compared with the payoff. The trade-off is resolved in two steps: the optimal strategy that maximizes the organization's payoff is identified; then the organization that implements the optimal strategy and minimizes the cost of information processing is chosen.

3. The Pie Problem

The following simplifying assumption on the class of strategies is maintained for the rest of this paper.

(A2) The set of circumstances, S, is a subset of 2-dimensional Euclidean space.

When S satisfies (A2), the definition of expert implies that each expert is a line in the 2-dimensional space. If an organization has multiple experts, S is divided into "cells," each of which is a polygon formed by the lines. A strategy Y of the organization is an assignment of t or f to each cell. The cells in S can be rather arbitrary. To facilitate analysis, a further restriction is imposed on the strategies:[7]

(A3) All lines pass through a single circumstance in S.

Strategies satisfying (A3) are called "pie strategies," and the problem of constructing efficient organizations for pie strategies is called the "pie problem."

The geometric nature of pie strategies is best illustrated by the following problem of an air traffic controller.[8] The controller is situated at the center of airport arrival terminals, where planes are scheduled to arrive from different directions. The task of the controller is to sound an emergency alarm when a plane comes from an unexpected

[7] Li (1995) considers a related class of strategies (called checkerboard strategies) where the cells are formed by two groups of parallel lines. Conclusions in the present paper regarding inefficiency of the hierarchy hold for those strategies as well.

[8] Another example of pie strategies can be given in repeated two-by-two games. Repeated game strategies where action in each period depends only on the two players' past average payoffs satisfy (A2). If the average payoffs converge, the long-run relevant part of each player's strategy satisfies (A3). Cho and Li (1997) examine repeated two-by-two games played by networks of the kind described in the present paper.

direction, i.e., a direction incompatible with the arrival schedule. His strategy here is given by the schedule, and it naturally satisfies (A2) and (A3). The controller can employ simple monitoring devices, each of which can detect a plane when it appears in the half space that the monitor faces. The controller may also employ processors that can summarize a vector of binary inputs into a single binary variable. Constructing the efficient network with the monitoring devices and the processors to implement a given strategy provides an example of the pie problem.

Fix a pie strategy Y formed by L lines. Two adjacent lines form two "mirror slices." There are $2L$ slices in total. Strategy Y assigns t or f to each slice p, denoted as $Y(p) = t$ or $Y(p) = f$. Choose any line as l_1, number the lines counter-clockwise as $l_1, \ldots, l_L$. Number the slices counter-clockwise as $p_1, \ldots, p_{2L}$, starting from a slice formed by l_1 and l_2. Each line l_j, $j = 1, \ldots, L$, intersects two pairs of mirror slices, p_{j-1} and p_{j-1+L}, and p_j and p_{j+L} (modulo $2L$). A line l_j is indispensable if across the line there is a change in decision assignment among the 4 slices. Then, one of the following is true:

[1] One of the four slices has a different assignment from the remaining three;

[2] $Y(p_{j-1}) = Y(p_{j+L}) \neq Y(p_j) = Y(p_{j-1+L})$;

[3] $Y(p_{j-1}) = Y(p_{j-1+L}) \neq Y(p_j) = Y(p_{j+L})$.

Line l_j is called a type 1,2,3 line respectively, if it satisfies [1], [2], or [3]. For a type 1 line, the report of the corresponding expert is critical only in distinguishing two adjacent slices. For a type 2 line, the report of the corresponding expert is critical in distinguishing two pairs of adjacent slices, but it plays the same role in the two situations (the two slices on the same side the line are assigned with the same decision). For a type 3 line, not only the report of the corresponding expert is critical in distinguishing two pairs of adjacent slices, but the report is used differently in the two situations. Note that a type 3 line creates two pairs of mirror slices assigned with the same decision. It will be seen from the following analysis that the report from a type 3 line cannot be fully summarized by a single immediate superior.

3.1. Two-rank hierarchies

Some pie strategies can be implemented efficiently by a two-rank hierarchy. A two-rank hierarchy consists of a decision maker and L experts, corresponding to the L lines that form the strategy.

Proposition 3.1. *A necessary and sufficient condition for a pie strategy Y with L lines to be implemented by a two-rank hierarchy is*

(C1) *there are no $k, k' \leq L$, such that $Y(p_k) = Y(p_{k+L}) \neq Y(p_{k'}) = Y(p_{k'+L})$.*

Since all the lines of Y are indispensable, an organization that implements Y needs to have at least one expert for each l, and so the number of experts is at least L.

Corollary 3.2. *If a strategy satisfies* (C1)*, then it is efficiently implemented by a two-rank hierarchy.*

Under condition (C1) in Proposition 3.1, either there is no pair of mirror slices assigned with the same decision, or all slices in such pairs are assigned with the same decision. Figure 3.1 gives two examples of strategies that satisfy the condition. (In all figures of this section and the appendix, a slice assigned with t by a strategy is shaded.) The left diagram shows a strategy with two type 1 lines and a single pair of mirror slices assigned with the same decision. The right diagram shows a strategy with three type 2 lines and no pair of mirror slices assigned with the same decision.

[Insert Figure 3.1 Here]

Consider again the airport controller's problem mentioned before. Suppose that the controller has to carry out the strategy depicted in the left diagram of Figure 3.1. That is, the controller wants to set off an alarm except when a plane arrives from the lower-left unshaded quarter in the diagram. According to Proposition 3.1, the strategy can be implemented by a hierarchy of two monitors, each of whom watches half of the state space S, and a decision maker who receives the reports from the two monitors and decides whether to sound the alarm. In particular, monitor 1, say, watches the right half, and reports 1 whenever a plane appears there and 0 otherwise. Monitor 2 reports 1 whenever a plane appears in the upper half and 0 otherwise. The decision maker adds up the two reports and sounds the alarm if and only if the sum is greater than 0. The threshold can be any number between 0 and 1.

But this strategy can also be implemented by the two monitors, without the decision maker, if each monitor has an alarm and pulls it whenever a plane appears in the half space he watches. In the sense that the final decision is made by the two monitors under different circumstances, one can say that decision making in this design problem can be "decentralized." Note that when a plane appears in the region watched by both monitors, namely the upper-right quarter in the left diagram of Figure 3.1, two alarms will be sounded. This does not cause any trouble in this example of airport controller, but it may not be the case in general depending on the nature of the decision to be made.

Restriction (R1) requires a fixed agent in the organization to make the choice under all circumstances. Although there is only one decision to be made for organizations considered in this paper, this decision does not have to made by a single agent under every circumstance, as shown by the example above. It is important to justify (R1) on the grounds of efficiency, especially when agents within an organization are assumed to share the same objective.

It turns out that the kind of decentralized decision making discussed above does not occur when the number of lines in a strategy is greater than 2. For example, for the strategy with three lines depicted in the right diagram of Figure 3.1, no single monitor has sufficient information to sound an alarm under any circumstance. One may consider other kinds of decentralized decision making where the right of making the final decision is dispersed within the organization. But if there is no fixed decision maker, the organization must have a mechanism to select a decision maker according to the circumstance.[9] Insofar as implementation of this mechanism is costly, such organization may be inefficient.

[Insert Figure 3.2 Here]

[9] The problem of assigning the right of decision making under decentralized information is the subject of mechanism design in communication networks. See, e.g., Reichelstein (1992). This type of model does not consider the computation involved in the mechanism, which is the focus of information processing in this paper.

Figure 3.2 illustrates a strategy with 4 lines that cannot be implemented by a two-rank hierarchy. There is one pair of mirror slices assigned with t, and another pair assigned with f. The inability of two-rank hierarchies to implement strategies where (C1) fails stems from the fact that the information provided by experts cannot be adequately utilized by a single decision maker due to his limited information processing capability. The decision maker performs a linear threshold function with fixed weights that partitions the weighted sum of reports into two sets. The weights have to be such that the weighted sum falls into one set for each circumstance assigned with t, and it falls into the other set for each circumstance assigned with f. When (C1) is not satisfied, such weights cannot be found. It is then impossible for the decision maker to make the right choices under all circumstances.

3.2. Three-rank hierarchies

By Proposition 3.1, a strategy Y where condition (C1) fails cannot be implemented by a two-rank hierarchy due to the limited capability of the decision maker. In general, one can think of two ways to overcome the limitation. One is to hire a more capable decision maker. This will be discussed later. The second way is to expand the hierarchy by adding middle managers. This helps because middle managers can summarize the information provided by the experts into reports that can be handled by the decision maker with limited capability.

PROPOSITION 3.3. *A necessary and sufficient condition for a pie strategy Y with L lines to be implemented by a three-rank hierarchy with L experts is*

(C2) *there is no $k = 1, \ldots, L$ such that $Y(p_{k-1}) = Y(p_{k-1+L}) \neq Y(p_k) = Y(p_{k+L})$.*

The sufficiency part of Proposition 3.3 is proved by constructing a network of L experts and two middle managers. Under (R3), the minimum number of middle managers for an organization of three ranks is two. The following corollary is straightforward.

COROLLARY 3.4. *If a strategy fails* (C1) *but satisfies* (C2)*, then it is efficiently implemented by a three-rank hierarchy.*

[Insert Figure 3.3 Here]

Figure 3.3 illustrates a three-rank hierarchy that efficiently implements the strategy shown in Figure 3.2. The first middle manager assigns 1 to a circumstance if and only if it falls into the shaded slice formed by l_1 and l_2. The second middle manager assigns 1 to a circumstance if and only if it falls into the shaded slice formed by l_3 and l_4. The decision maker assigns 1 to a circumstance if and only if one middle managers assigns 1 to it. Thus, the two middle managers break down the original decision problem into two parts, formed by l_1 and l_2, and by l_3 and l_4, enabling a decision maker with limited information processing capability to make the right decisions under all circumstances. Proposition 3.3 shows that this observation is generally true when (C2) is satisfied.

For the strategies that can be implemented by three-rank hierarchies, the number of middle managers is two, independent of the number of experts at the lowest rank of the hierarchy. For these strategies, efficient hierarchies can be very flat and quite effective in aggregating information provided by the experts.

Condition (C2) in Proposition 3.3 is equivalent to there being no type 3 line among the lines forming the strategy. Figure 3.4 gives one strategy formed by 3 lines with one of them (line l_2) a type 3 line. Not only there is one pair of mirror slices assigned with t and another pair assigned with f, but they are adjacent to each other.

[Insert Figure 3.4 Here]

A strategy that fails condition (C2) cannot be implemented by a three-rank hierarchy with L experts. Thus, adding a rank of middle managers is only partially effective in overcoming the limitation on the hierarchy's implementation capacity. In fact, the proof of Proposition 3.3 indicates that a strategy with type 3 lines cannot be implemented by a hierarchy of any rank. When (C2) fails, adding more ranks of middle managers does not help. However, note the provision in Proposition 3.3 that the organization has the same number of experts as the number of lines that form the strategy. In particular, for each type 3 line, only one expert is allowed in the organization. When this restriction is dropped, it can be shown that any strategy can be implemented by a three-rank hierarchy, although often inefficiently. This will be easy after characterizing the efficient non-hierarchies that implement strategies where (C2) fails.

3.3. Efficient organizations for general pie strategies

If non-hierarchies are allowed, and if there is no restriction on the number of middle managers, then any Y can be implemented by an organization with three ranks. To see this, define a "region" as a collection of adjacent slices with the same assignment by Y. Let N_t be the number of t-regions (regions assigned with t). For each t-region, define a middle manager as the superior to the two lines representing the two boundaries of the region, such that the middle manager assigns 1 to a circumstance if it falls into the region, and 0 otherwise. Then construct an organization with three ranks and N_t middle managers where the decision function assigns t to a circumstance if it is assigned with 1 by any of the middle managers. Clearly, this organization implements the strategy. Note that the organization is non-hierarchical if there are type 2 or type 3 lines, because each such line must report to two middle managers. The following result shows that this construction is typically inefficient, and that two middle managers are sufficient.

Proposition 3.5. *Any pie strategy can be implemented by a three-rank organization with two middle managers.*

The next corollary follows directly from Propositions 3.3 and 3.5.

Corollary 3.6. *If a strategy fails* (C2)*, then it is implemented efficiently by a three-rank non-hierarchy with two middle managers.*

The left diagram in Figure 3.5 illustrates a three-rank non-hierarchy that efficiently implements the strategy shown in Figure 3.4. The first middle manager assigns 1 to a circumstance if and only if it falls into the shaded slice formed by l_1 and l_2. The second middle manager assigns 1 to a circumstance if and only if it falls into the shaded slice formed by l_2 and l_3. The decision maker assigns 1 to a circumstance if and only if one middle managers assigns 1 to it. The expert representing the type 3 line l_2 reports to both middle managers. This makes the network non-hierarchical, but it is the key to efficiency.

The right diagram in Figure 3.5 illustrates a three-rank hierarchy that implements the same strategy as the left diagram. The single expert representing l_2 is replaced by two experts, which have exactly the same position but report to the two middle managers separately. Indeed, any Y can be implemented by a three-rank hierarchy with two middle managers if the organization has two experts corresponding to each type 3 line. Such three-rank hierarchy is inefficient. The inefficiency comes from the failure of the hierarchy to fully utilize the information processing capabilities of the experts representing type 3 lines, due to the constraint that in the hierarchy each agent can report to only one superior.

[Insert Figure 3.5 Here]

The issue of inefficient utilization of information processing capabilities of some agents in the organization is also at the center of "skip-level reporting." In skip-level reporting, an agent of rank r directly reports to another agent of rank $r' \geq r + 2$. This possibility is ruled out by (R3). When this restriction is dropped, efficient organizations generally have three ranks with only one middle manager.

PROPOSITION 3.7. *If* (R3) *is dropped, the efficient organization that implements a strategy where* (C1) *fails has three ranks with one middle manager.*

Thus, the three-rank organizations constructed in Propositions 3.3 and 3.5 are inefficient if (R3) is dropped. The information processing task of either one of two middle managers in the organizations can be performed by the decision maker, without affecting the decision function performed by the decision maker.[10] This requires the decision maker to receive reports from the single middle manager and some of the experts. The experts who report to the decision maker have to wait for the middle manager to complete his information processing task before they send reports directly to the decision maker, or the decision maker has to store the reports from these experts until he receives the

[10] Radner (1993) studies networks designed to add up numbers with minimum delay, i.e., the time interval between receiving a batch of numbers by the network and obtaining the final sum. Efficient networks also require skip-level reporting to fully utilize the capacities of processors at higher levels and to reduce idle time.

report from the middle manager. Thus, skip-level reporting introduces a timing problem to organizational design, without reducing the total time necessary to complete the information processing task in implementing a strategy. Insofar as this timing problem requires additional mechanism within the organization to deal with it, one may not say that the organization constructed in Proposition 3.7 is more efficient than the organizations constructed in Propositions 3.3 and 3.5.

Propositions 3.1, 3.3, and 3.5, together with their corollaries, completely characterize the structure of efficient organizations for pie strategies. The next question is how "likely" a hierarchy will emerge as an efficient organization under some suitably defined probability measure. Since (C1) implies (C2), the latter is both necessary and sufficient for a hierarchy to be an efficient organization for a strategy Y. To define one reasonable probability measure, consider all strategies with L lines. Fix the orientation of the lines and imagine assigning t and f randomly to the $2L$ slices. There are 2^{2L} possible assignments, each producing a different strategy. Suppose that each of the 2^{2L} strategies is equally likely. Then one can compute the likelihood of an efficient hierarchy by counting the number of strategies where (C2) is satisfied, or equivalently, by calculating the probability that none of the L lines is of type 3. For the sake of the argument, suppose that L is an even number, and consider every other line.[11] The probability that any line is of type 3 is $2/16$, since 2 out of 16 possible assignments of decisions to the 4 slices intersected by the line require a type 3 line. Thus, the probability that none of the L lines is of type 3 is less than $(14/16)^{L/2}$, which converges to zero as L increases.

PROPOSITION 3.8. *The likelihood of an efficient hierarchy for a pie strategy goes to zero as L becomes arbitrarily large.*

3.4. Efficient organizations when the decision maker is more capable

The results obtained so far demonstrate that efficient organizations are generally nonhierarchical because the hierarchy can fail to fully utilize the information processing capabilities of the agents. As mentioned above, the implementation capacity of a hierarchy is

[11] This simpler argument is suggested by a referee.

also limited by the capability of the decision maker. In fact, it is straightforward to show that any Y can be implemented by a two-rank hierarchy if the decision maker can perform a threshold function with a sufficiently large number of thresholds. To see this, suppose that Y is formed by L lines. For each $k = 1, \ldots, L$, assign a weight $w_k = 2^{k-1}$ to the report from the expert corresponding to l_k. Then each slice p_k corresponds to a weighted sum $w \cdot b_k = 2^L - 2^k$, and each p_{k+L} corresponds to $w \cdot b_k = 2^k - 1$. One can check that each of the $2L$ slices is assigned with a different sum, and so Y can be implemented by a two-rank hierarchy with a decision maker that performs a threshold function with at most $2L - 1$ thresholds. The following result shows that in fact two thresholds are sufficient.

PROPOSITION 3.9. *Any strategy can be implemented by a two-rank hierarchy if the decision maker can perform a double-threshold function.*

The results of Propositions 3.5 and 3.9 illustrate that the structure of efficient organizations depends crucially on the information processing capability of the decision maker. Furthermore, the two propositions identify a form of increasing returns to information processing capability of the decision maker.[12] According to Propositions 3.3 and 3.5, when (C1) is not satisfied, the efficient organization that implements the strategy has three ranks and two middle managers. Proposition 3.9 states that the same strategy can be implemented by a two-rank organization without middle managers if the decision maker is "twice" as capable as the experts. Thus, if a decision maker that is twice as capable costs less than three times what an expert costs, the optimal choice of organization is a two-rank hierarchy.

4. Concluding Remarks

The works that are most closely related to the present paper may be loosely called the "network design" literature.[13] These works differ in the nature of information processing

[12] Increasing returns to "ability" in various forms are common in works on organizational structure. Earlier works on optimal hierarchy design, such as Williamson (1967), Beckmann (1977), and Rosen (1984), assume a production technology that exhibits increasing returns. In Geanakoplos and Milgrom (1991), increasing returns are a consequence of an information acquiring technology.

[13] For a survey of recent developments in this literature, see Van Zandt (1996).

function carried out by the networks and in the criterion for selecting a particular structure as the efficient one, but they all assume that agents share the same objective.

One of the first contributions to the network design literature is the theory of teams of Marschak and Radner (1972). They analyze a situation where there are different decisions to be made based on different sets of information. The focus of their analysis is efficient use of information in an organization with diverse information, rather than efficient information processing. Radner (1993) studies networks that implement associative operations such as addition. The criterion of efficiency induces a trade-off between the number of processors of an organization and the time it takes the organization to complete the information processing task. He shows that relatively simple hierarchies are sufficient to achieve near efficiency when the organization is overloaded with tasks. However, the networks he considers are all hierarchical as defined in the present paper.[14] Radner and Van Zandt (1992) use the same framework to study the returns to scale from information processing. They show that both increasing and decreasing returns can occur, depending on characteristic of the organization's environment. Sah and Stiglitz (1986) analyze a situation where agents of an organization are prone to mistakes in selecting investment projects. They show that the performance of an economic organization of a given structure, in terms of the probability of selecting the right projects, depends on the characteristics of the environment, and they provide conditions under which a hierarchical organization is efficient. Marschak and Reichelstein (1992) examine communication networks that minimize the total size of messages transmitted in the decision making process among agents with diverse information. They identify situations where hierarchies are efficient, and they find that such situations are rare. Although this finding is consistent with that of the present paper, their focus is the communication cost within the network, while the focus of this paper is the computational aspect of information processing.

In this paper the hierarchical form is examined from the point of view of minimizing information processing cost for given strategies. What has emerged from the analysis of

[14] Even for associative operations, hierarchical networks need not be efficient for "real time" processing. See Van Zandt (1996).

a special class of strategies is that, more often than not, the efficient organization is non-hierarchical. Although it is also shown that hierarchies are efficient if decision makers are more capable, it can be more reasonable to assume limited information processing capabilities. Under this assumption, non-hierarchical organizations tend to emerge as a result of fully utilizing the information processing capabilities of the agents of organizations.

Although pie strategies are special, the analysis suggests that the result that hierarchies can be inefficient in information processing holds in more general problems. In a pie problem with L experts, the space of circumstances is partitioned into $2L$ regions, each represented by an L-dimensional vector of binary reports from the experts. Efficiency in information processing requires each expert corresponding to a type 3 line to report to more than one middle manager. More generally, we can think of the space of circumstances partitioned into 2^L regions, each represented by an L-dimensional vector, and a strategy Y that maps the 2^L vectors into the two decisions. The analog of a type 3 line is an expert, say the first one, such that

$$Y(1, b_2, \ldots, b_n) = Y(0, b'_2, \ldots, b'_n) \neq Y(0, b_2, \ldots, b_n) = Y(1, b'_2, \ldots, b'_n).$$

That is, expert 1's report is critical in distinguishing two pairs of circumstances, and moreover, the report is used differently in the two situations. It can be shown that strategies with such expert cannot be implemented efficiently by a hierarchy.[15] Since the argument is the same as in the case of pie strategies discussed in the previous sections, a sketch of proof suffices. First, it can be shown that any strategy with a type 3 expert cannot be implemented by a two-rank hierarchy, as in the necessary part of Proposition 3.1. Second, such strategy cannot be implemented by a hierarchy of any rank if the type 3 expert's information processing task is not replicated by other experts, as shown in the necessary part of Proposition 3.3. It then follows that for a hierarchy to implement the strategy, any type 3 expert's information processing task must be carried out by at least two experts that report to different middle managers. But then they can be replaced by a single expert

[15] The sufficiency part of Proposition 3.3 does not hold in the general case. That is, a strategy that does not have a type 3 expert may not be implemented efficiently by a hierarchy. See Li (1995) for an example.

that reports to the respective middle managers (a reverse of the argument in Proposition 3.5), reducing the number of agents in the network. Thus, by the efficiency criterion, the hierarchy cannot be the efficient structure that implements a strategy with type 3 experts. Complete characterization of the structure of the efficient information-processing organizations in this more general binary decision problem is beyond the scope of this paper and is an interesting topic for future research.

The results in this paper suggest that for the hierarchy to be the predominant choice in economic organizations, new ingredients must be added to the model of information-processing networks. One candidate is diverse incentives within an organization. When agents do not share the same objectives as the organization itself, efficient non-hierarchies may create control loss due to lack of accountability. Since in hierarchies a single superior is responsible for handling the information provided by any given agent, it can be easier for the organization to monitor the information processing process. Casual empiricism seems to suggest that the hierarchy enjoys an advantage over non-hierarchies from a control point of view. How to incorporate diverse incentives in a model of information-processing networks is an interesting issue worth pursuing.

Modeling organizations as decentralized information-processing networks is also useful in identifying the additional information processing cost that must be incurred if a hierarchy is chosen over an efficient non-hierarchy for other reasons. Moreover, it is worth noting that if the criterion of efficiency is more than cost-minimization, the hierarchy can have certain advantages even when incentives are not an issue. One example is a situation in which agents in an information-processing network may make mistakes. The result in this paper shows that a cost-minimizing organization need not be hierarchical in that some agents report to more than one superior, although the same strategy can always be implemented by a hierarchy if additional agents duplicate their information processing tasks. Such redundancy is costly, but when agents are prone to mistakes, it provides a convenient way for the organization to monitor the information processing process by comparing the reports from the agents that share the same information processing tasks.

Appendix

For each $j = 1, \ldots, L$, line l_j is represented by expert e_j. Assume that e_j assigns 1 to circumstances that fall on the same side of l_j as p_0, and 0 if otherwise. Then, each slice p_j ($j = 1, \ldots, L$) is uniquely represented by an L-dimensional binary vector, with the first $j-1$ coordinates equal to 0 and the rest $L-(j-1)$ coordinates equal to 1. Denote this vector as b_j. Let b_{j+L} be the binary vector that represents p_{j+L}. Note that b_{j+L} is the binary complement of b_j. This is denoted as $b_{j+L} = \bar{b}_j$. Finally, l_j is called a type $1t$ line if it is a type 1 line and only one of the four slices intersected by l_j is assigned with f. Define type $1f$ lines in a similar way.

A.1. Proof of Proposition 3.1

(Necessity) Assume that (C1) is violated by p_{k_1} and p_{k_2}. By way of contradiction, suppose that a two-rank hierarchy with experts e_1, …, e_L, and a decision maker d implements Y. Denote the threshold function performed by d as $\Delta(w \cdot b - \theta)$, where b is an L-dimensional binary vector that represents some p in Y. Since (C1) is violated by p_k and $p_{k'}$, without loss of generality I can write $\Delta(w \cdot b_k - \theta) = \Delta(w \cdot \bar{b}_k - \theta) = 1$, and $\Delta(w \cdot b_{k'} - \theta) = \Delta(w \cdot \bar{b}_{k'} - \theta) = 0$. The first equation implies that the sum of all weights is greater than or equal to 2θ, while the second implies that it is smaller than 2θ, a contradiction.

(Sufficiency) It suffices to show that there exists a threshold function $g(b) = \Delta(w \cdot b - \theta)$ with weight vector w and threshold θ, such that for $j = 1, \ldots, 2L$, $\Delta(w \cdot b_j - \theta) = 1$ if $Y(p_j) = t$ and 0 if $Y(p_j) = t$. Under (C1), Y cannot have type 3 lines. Furthermore, Y cannot have both type $1t$ and type $1f$ lines: either all lines are of type $1t$ or type 2, or all lines are of type $1f$ or type 2. For the first case (the second case is similar), consider the following assignment of weights: for each l_j,

$$w_j = \begin{cases} 1, & \text{if } l_j \text{ is of type } 1t\text{, and either } Y(p_j) = f \text{ or } Y(p_{j-1+L}) = f \\ -1, & \text{if } l_j \text{ is of type } 1t\text{, and either } Y(p_{j-1}) = f \text{ or } Y(p_{j+L}) = f \\ 2, & \text{if } l_j \text{ is of type 2 and } Y(p_j) = f \\ -2, & \text{if } l_j \text{ is of type 2 and } Y(p_{j-1}) = f. \end{cases} \tag{A.1}$$

Let $\sigma = \sum_{j=1}^{L} w_j$ be the sum of the weights. It can be shown by induction that under

(A.1), for all $j = 1, \ldots, L$,

$$\begin{cases} w \cdot b_j = w \cdot b_{j+L} = \sigma/2, \text{ if } Y(p_j) = Y(p_{j+L}) = t \\ w \cdot b_j = \sigma/2 + 1 \text{ and } w \cdot b_{j+L} = \sigma/2 - 1, \text{ if } Y(p_j) = t \text{ and } Y(p_{j+L}) = f \\ w \cdot b_j = \sigma/2 - 1 \text{ and } w \cdot b_{j+L} = \sigma/2 + 1, \text{ if } Y(p_j) = f \text{ and } Y(p_{j+L}) = t. \end{cases} \tag{A.2}$$

The existence of g then follows. *Q.E.D.*

A.2. Proof of Proposition 3.3

(Necessity) To prove the proposition by contradiction, suppose that l_k is a type 3 line, and that Y is implemented by a three-rank hierarchy with experts $e_1, \ldots, e_L$. Let m be the unique immediate superior to e_k. The four slices adjacent to l_k are p_{k-1}, p_k, p_{k-1+L}, and p_{k+L}. By an argument similar to the first part of the proof of Proposition 3.1, it is impossible for the function $M(p)$ carried out by m to satisfy $M(p_{k-1}) = M(p_{k-1+L}) \neq M(p_k) = M(p_{k+L})$. Then one of the following must be true:

[1] $M(p_{k-1}) = M(p_k)$;

[2] $M(p_{k-1+L}) = M(p_{k+L})$;

[3] $M(p_{k-1}) = M(p_{k+L}) \neq M(p_k) = M(p_{k-1+L})$.

The first two cases are clearly impossible, because l_k reports to only one immediate superior, and $Y(p_{k-1}) \neq Y(p_k)$. For case [3], without loss of generality, I can assume that $M(p_{k-1}) = M(p_{k+L}) = 1$, and $M(p_k) = M(p_{k-1+L}) = 0$. Let $\Delta(\omega \cdot v - t)$ be the threshold function of the decision maker, where v is a binary vector that denotes the reports from all middle managers. Let v_1 and v_2 be the vectors that represent p_{k-1} and p_k respectively. Since l_k has only one immediate superior, no other middle manager except m distinguishes p_{k-1} from p_k. Therefore, $\omega \cdot v_1 = \omega \cdot v_2 + \tilde{\omega}$, where $\tilde{\omega}$ is the weight assigned by the decision maker to the report from m. Similarly, the vectors that represent p_{k-1+L} and p_{k+L}, v_1' and v_2' respectively, satisfy $\omega \cdot v_1' + \tilde{\omega} = \omega \cdot v_2'$. Since $Y(p_{k-1}) = Y(p_{k-1+L}) \neq Y(p_k) = Y(p_{k+L})$, I can assume that $\Delta(\omega \cdot v_1 - t) = \Delta(\omega \cdot v_1' - t) = 1$, and $\Delta(\omega \cdot v_2 - t) = \Delta(\omega \cdot v_2' - t) = 0$. But these conditions imply that $\omega \cdot v_1 \geq t > \omega \cdot v_1 - \tilde{\omega}$, and $\omega \cdot v_1' \geq t > \omega \cdot v_1' + \tilde{\omega}$, a contradiction.

(Sufficiency) Fix any Y that satisfies (C2). Assume that it does not satisfy (C1). (The proof is trivial if it does.) I construct a three-rank hierarchy with L experts and two middle managers that implements Y. For each $i = 1, 2$, $\mathcal{M}^i$ denotes the set of lines that report to the i-th middle manager m^i. Let $L^i = |\mathcal{M}^i|$, with $L^1 + L^2 = L$. For the decision function of the hierarchy, each slice p is represented by a vector of reports $(M^1(p), M^2(p))$ from the two middle managers. Each M^i can be thought of as the threshold function carried out by m^i which implements a strategy formed by the L^i lines in $\mathcal{M}^i$. Denote the strategy as Y^i, with the convention that $Y^i(p) = t$ if and only if $M^i(p) = 1$. I will show that both Y^1 and Y^2 satisfy (C1), and so by Proposition 3.1 M^1 and M^2 can be found which implement Y^1 and Y^2 respectively. Moreover, the L lines can be selected to form Y^1 and Y^2 so that the following is satisfied:

$$\begin{cases} M^1(p) \neq M^2(p), \text{ for all } \; p \; \text{ such that } Y(p) = t \\ M^1(p) = M^2(p) = 0, \text{ for all } \; p \; \text{ such that } Y(p) = f. \end{cases} \tag{A.3}$$

The sufficiency of (C2) then follows by defining a decision maker d that maps (1,0) and (0,1) to 1 and (0,0) to 0, which can be done with a weight vector (1,1) and a threshold 0.5.

The formation of Y^1 and Y^2 with the desired properties is accomplished by a repetitive procedure where in each step lines in $\mathcal{M}^1$ are selected to be moved to $\mathcal{M}^2$. To begin, let $\mathcal{M}^1$ be the collection of all the lines in Y, and $Y^1(p) = Y(p)$; let $\mathcal{M}^2 = \emptyset$ and $Y^2(p) = f$ for all p. Since Y^1 does not satisfy (C1), there are at least one type $1t$ line and one type $1f$ line. Choose a type $1t$ line l_{k_1} and a type $1f$ line l_{k_2} such that $k_1 < k_2$ and the lines l_k, $k = k_1 + 1, \ldots, k_2 - 1$, if any, are of type 2. It follows that $Y^1(p_{k_1}) = f$ or $Y^1(p_{k_1+L^1}) = f$. (Otherwise, since l_{k_1} is a type $1t$ line, $Y^1(p_{k_1}) = Y^1(p_{k_1+L^1}) = t$, but then l_{k_1+1} must be a type $1t$ line, contradicting the definitions of l_{k_1} and l_{k_2}.) Without loss of generality, I assume that $Y^1(p_{k_1}) = f$. Since p_{k_1} is a type $1t$ line, $Y^1(p_{k_1+L^1}) = Y^1(p_{k_1-1}) = Y^1(p_{k_1-1+L^1}) = t$. Consider the assignments of Y^1 to p_{k_1-2} and p_{k_1-2+L}. Since there are no type 3 lines in Y^1, one of the following must be true:

[1] $Y^1(p_{k_1-2}) = f, \quad Y^1(p_{k_1-2+L^1}) = t;$

[2] $Y^1(p_{k_1-2}) = t, \quad Y^1(p_{k_1-2+L^1}) = f.$

In case [1], decompose the slices of Y^1 and form new Y^1 and Y^2 as shown in Figure A.1. The first diagram from the left is Y^1 before decomposition (only relevant lines are shown), the middle diagram shows the new strategy Y^1, and the right diagram shows the new addition to Y^2. The new $\mathcal{M}^1$ has $L^1 - 2$ lines. The new Y^1 still satisfies (C2). With the two type $1t$ lines in the original Y^1, l_{k_1} and l_{k_2}, moved to $\mathcal{M}_2$, the number of type $1t$ lines is reduced by 2. Note that both l_{k_1} and l_{k_1-1} in Y^2 are type $1f$ lines.

[Insert Figure A.1 Here]

Case [2] is similar. In both case [1] and case [2], the decomposition of the strategy Y^1 satisfies (A.3). Note that Y^2 satisfies (C1) after the decomposition because it consists of only type $1f$ lines and possibly type 2 lines. Moreover, in either case, the number of $1t$ lines in Y^1 is reduced by 2, while the number of $1f$ lines in Y^1 does not change. I can then proceed to define new l_{k_1} and l_{k_2} and repeat the above procedure. The desired properties hold after each step of the procedure. After a finite number of steps, Y^1 has no $1t$ lines and therefore satisfies (C1), and the selection of lines into $\mathcal{M}_1$ and $\mathcal{M}_2$ is completed. *Q.E.D*

A.3. Proof of Proposition 3.5

If there are no type 3 lines, the proposition follows from Proposition 3.3. Suppose there is only one type 3 line l_k. Without loss of generality, assume that $Y(p_{k-1}) = Y(p_{k-1+L}) = t$ and $Y(p_k) = Y(p_{k+L}) = f$. Add a new line l'_k between l_k and l_{k+1}, as shown in Figure A.2. The left diagram shows the original strategy (only relevant lines are shown); the right diagram shows how l'_k is added. Note that l'_k is a type 1 line and l_k is changed from type 3 to type 1. Let Y' be the transformed strategy shown in the right diagram. Because there are no type 3 lines in Y', by Proposition 3.3, Y' can be implemented by a three-rank hierarchy with two middle managers. Note that Y' differs from the original strategy Y only in the two slices formed between l_k and l'_k. Thus, if in the three-rank hierarchy l'_k is replaced with a line $\tilde{l}_k$ that has the same position as l_k, the resulting three-rank hierarchy implements Y. Moreover, in this hierarchy, l_k and $\tilde{l}_k$ report to different middle managers. Otherwise, one of them can be eliminated with its weight added to that of the other one, and the resulting three-rank hierarchy implements Y, a contradiction to Proposition 3.3. Since l_k and $\tilde{l}_k$ report to different middle managers in the three-rank hierarchy, by eliminating $\tilde{l}_k$ and having l_k report to both middle managers, I can convert the three-rank hierarchy into a three-rank non-hierarchy that implements Y.

[Insert Figure A.2 Here]

Clearly, the above argument does not depend the assumption that Y has only one type 3 line. This completes the proof of the proposition. *Q.E.D*

A.4. Proof of Proposition 3.7

Fix any strategy Y where (C1) fails. By Propositions 3.3 and 3.5, there is a three-rank efficient organization (hierarchy or non-hierarchy) with two middle managers which implements it. Denote the two middle managers as m^i, $i = 1, 2$. Let $\mathcal{M}^i$ be the set of experts who report to m^i (note $\mathcal{M}^1 \cap \mathcal{M}^2$ may be non-empty). Let $\Delta(w^i \cdot b^i - \theta^i)$ be the threshold function of m^i, and $\Delta(\omega \cdot v - t)$ be the threshold function of the decision maker. By the constructions of efficient organizations in Propositions 3.3 and 3.5, I can assume without loss of generality that a slice p is assigned with t if and only if $v = (1, 0)$ or $v = (0, 1)$, and f if and only if $v = (0, 0)$.

Construct a new three-rank organization, with the original middle manager m^2 as the only middle manager, and a new decision maker that takes reports from both m^2 and the experts in $\mathcal{M}^1$. Denote the new decision function as $\Delta(\omega' \cdot v' - \theta^1)$, where $v' = (v_2, b^1)$ is vector of reports from m^2 and the experts in $\mathcal{M}^1$, and where the weight vector $\omega' = (\omega_2', w^1)$ satisfies $\omega_2' + \min_{\{b^1 | w^1 \cdot b^1 < \theta^1\}} w^1 \cdot b^1 \geq \theta^1$. That is, the new decision maker assigns ω_2' to the report v_2 from m^2, and he assigns w^1 to reports b^1 from the experts in $\mathcal{M}^1$. This new three-rank organization implements the same strategy Y. *Q.E.D*

A.5. Proof of Proposition 3.9

Fix any Y formed by L lines. First, suppose that Y satisfies (C2) but not (C1). (If Y also satisfies (C1), the construction of the double-threshold function becomes trivial.) By Proposition 3.3, Y can be implemented by a three-rank hierarchy with two middle managers. Recall that Proposition 3.3 is established by a decomposition procedure where in each step lines are selected to be moved from $\mathcal{M}^1$ to $\mathcal{M}^2$ until all L^1 lines that form Y^1 are of type $1f$ and type 2 and Y^1 satisfies (C1). A double-threshold function g that implements Y can be found by reversing the decomposition procedure used in the proof of Proposition 3.3. As the first step, consider the last step of the decomposition procedure. Let Y' be the strategy before the decomposition. As in the proof of Proposition 3.3, I consider two cases.

Case [1] is depicted in Figure A.1. The left diagram shows Y', and the middle diagram shows Y^1. Let $L' = L^1 + 2$ be the number of lines in Y'. Denote the $2L'$ slices in Y' in the usual fashion. Define two new slices $p_{k_1-2,k_1} = p_{k_1-2} \cup p_{k_1-1} \cup p_{k_1}$ and $\overline{p}_{k_1-2,k_1} = p_{k_1-2+L'} \cup p_{k_1-1+L'} \cup p_{k_1+L'}$. Let $J = \{k_1-2, k_1-1, k_1, k_1-2+L', k_1-1+L', k_1+L'\}$. For each $j \notin J$, let b^1_j be the L^1-dimensional binary vector representing p_j in Y^1, obtained by eliminating the (k_1-1)-th and k_1-th elements of b_j, the vector that represents p_j in Y'. Let $b^1_{k_1-2,k_1}$ and $\overline{b}^1_{k_1-2,k_1}$ be the binary vectors representing, respectively, p_{k_1-2,k_1} and $\overline{p}_{k_1-2,k_1}$, obtained by eliminating the (k_1-1)-th and k_1-th elements of b_{k_1-2} and $b_{k_1-2+L'}$. Since Y^1 satisfies (C1), by the construction in the proof of Proposition 3.1, there exist weights w^1 for the L^1 lines of Y^1 that satisfy (A.1), so that for all $j \notin J$,

$$\begin{cases} w^1 \cdot b^1_j = w^1 \cdot b^1_{j+L'} = \sigma/2, \text{ if } Y^1(p_j) = Y^1(p_{j+L'}) = f \\ w^1 \cdot b^1_j = \sigma/2+1 \text{ and } w^1 \cdot b^1_{j+L'} = \sigma/2-1, \text{ if } Y^1(p_j) = t \text{ and } Y^1(p_{j+L'}) = f \\ w^1 \cdot b^1_j = \sigma/2-1 \text{ and } w^1 \cdot b^1_{j+L'} = \sigma/2+1, \text{ if } Y^1(p_j) = f \text{ and } Y^1(p_{j+L'}) = t, \end{cases} \tag{A.4}$$

where σ is the sum of weights w^1. Moreover, since $Y^1(p_{k_1-2,k_1}) = f$ and $Y^1(\overline{p}_{k_1-2+L^1}) = t$ (see the middle diagram of Figure A.1),

$$w^1 \cdot b^1_{k_1-2,k_1} = \sigma/2 - 1; \quad w^1 \cdot \overline{b}^1_{k_1-2,k_1} = \sigma/2+1. \tag{A.5}$$

Assign weights w to the L' lines in Y' such that $w_k = w^1_k$ for $k \neq k_1, k_1-1$, $w_{k_1} = 3$, and $w_{k_1-1} = -3$. Since the weights of l_{k_1} and l_{k_1-1} sum up to 0, the sum of the weights w is still σ. Moreover, for all $j \notin J$, $w \cdot b_j = w^1 \cdot b^1_j$, $w \cdot b_{k_1-2} = w^1 \cdot b^1_{k_1-2,k_1}$, and $w \cdot b_{k_1-2+L'} = w^1 \cdot \overline{b}^1_{k_1-2,k_1}$. Thus, $w \cdot b_{k_1-1} = w^1 \cdot b^1_{k_1-2,k_1} + 3$, and $w \cdot \overline{b}_{k_1-1} = w^1 \cdot \overline{b}^1_{k_1-2,k_1} - 3$. Equations (A.4) and (A.5) then imply that

$$\begin{cases} w \cdot b_j = \sigma/2+1,\ \sigma/2+2, \text{ or } \sigma/2-2, \text{ if } Y'(p_j) = t \\ w \cdot b_j = \sigma/2 \text{ or } \sigma/2-1, \text{ if } Y'(p_j) = f. \end{cases} \tag{A.6}$$

It follows from (A.6) that Y' can be implemented by a double-threshold function with weights w and thresholds θ^1 and θ^2, where $\theta^1 \in [\sigma/2-2, \sigma/2-1)$, and $\theta^2 \in (\sigma/2, \sigma/2+1]$.

Case [2] is similar. In both cases, there is a double-threshold function that implements the strategy before the last step of the decomposition procedure in the proof of Proposition 3.3. Moreover, (A.6) is satisfied in both cases, and the same two thresholds can be chosen for the double-threshold function. Consider repeating this reverse procedure. Since each step of the decomposition procedure in the proof of Proposition 3.3 is independent from others, the steps in the reverse procedure are also independent. In particular, the sum of the weights does not change after each step, and the two thresholds need not be adjusted. In a finite number of steps, the reverse procedure establishes a double-threshold function that implements the original strategy Y.

If Y does not satisfy (C2), the same transformation as in the proof of Proposition 3.5 can be applied, where for each type 3 line l_k, a new type 1 line l'_k is added and l_k is changed to a type 1 line. This results in a strategy Y' which satisfies (C2) and which can be implemented by a double-threshold function. Then, each l'_k can be replaced with a line $\tilde{l}_k$ that has the same position as l_k, and the resulting double-threshold function implements the original strategy Y. After eliminating each $\tilde{l}_k$ and adding its weight to l_k, the double-threshold function implements Y with L experts. *Q.E.D*

References

Martin Beckmann (1977): "Management Production Function and the Theory of the Firm," *Journal of Economic Theory* 14, 1-18.

In-Koo Cho and Li, Hao (1997): "How Complex Are Networks Playing Repeated Games?" forthcoming in *Economic Theory*.

Ronald Coase (1937): "The Nature of the Firm," *Economica* 4, 387-405.

Jay R. Galbraith (1971): "Matrix Organizational Designs," *Business Horizons*, February, 29-40.

John Geanakoplos and Milgrom, Paul (1991): "A Theory of Hierarchies Based on Limited Managerial Attention" *Journal of the Japanese and International Economies* 5, 205-225.

Michael Keren and Levhari, David (1989): "Decentralization, Aggregation, Control Loss and Costs in a Hierarchical Model of the Firm," *Journal of Economic Behavior and Organization* 11, 213-236.

Hao Li (1995): *Two Essays on Networks and Organizations*. Doctoral dissertation. The University of Chicago.

Jacob Marschak and Radner, Roy (1972): *Economic Theory of Teams*, Yale University Press.

Thomas A. Marschak and Reichelstein, Steslice (1992): "Network Mechanisms, Informational Efficiency, and Hierarchies," mimeo, University of California at Berkeley.

Marvin L. Minsky and Papert, Seymour A. (1988): *Perceptrons: An Introduction to Computational Geometry*, 2nd. edition, MIT press.

Saburo Muroga (1971): *Threshold Logic and Its Applications*, John Wiley & Sons.

Roy Radner (1992): "Hierarchy: The Economics of Managing," *Journal Economic Literature* 30, 1382-1415.

Roy Radner (1993): "Organization of Decentralized Information Processing," *Econometrica* 61, 1109-1146.

Roy Radner and Van Zandt, Timothy (1992): "Information Processing in Firms and Returns to Scale," *Annales d'Economie et de Statistique* 25/26, 265-298.

Sherwin Rosen (1984): "Authority, Control and the Distribution of Earnings," *Bell Journal of Economics* 13, 311-323.

Ariel Rubinstein (1992): "On Price Recognition and Computational Complexity in a Monopolistic Model," *Journal of Political Economy* 101, 473-484.

Raaj K. Sah and Stiglitz, Joseph E. (1986): "The Architecture of Economic Systems: Hierarchies and Polyarchies," *American Economic Review* 76, 716-727.

Timothy Van Zandt (1996): "Decentralized Information Processing in the Theory of Organizations," forthcoming in *Contemporary Economic Development Reviewed, Volume 4: The Enterprise and its Environment*, edited by Murat Sertel, London: MacMillan Press Ltd.

Gérard Weisbuch (1990): *Complex System Dynamics: An Introduction to Automata Networks*, Lecture Note Volume II, Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley.

Oliver E. Williamson (1967): "Hierarchical Control and Optimum Firm Size," *The Journal of Political Economy* 75, 123-178.

Figure 2.1

Figure 3.1

Figure 3.2

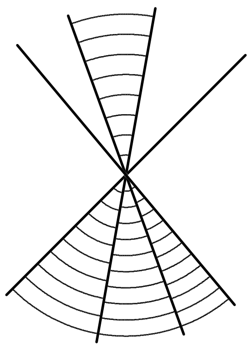

Figure 3.3

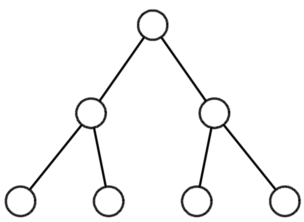

Figure 3.4

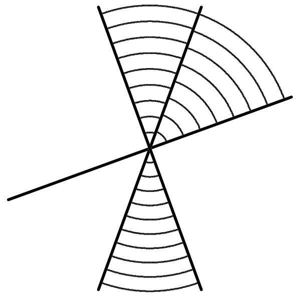

Figure 3.5

Figure A.1

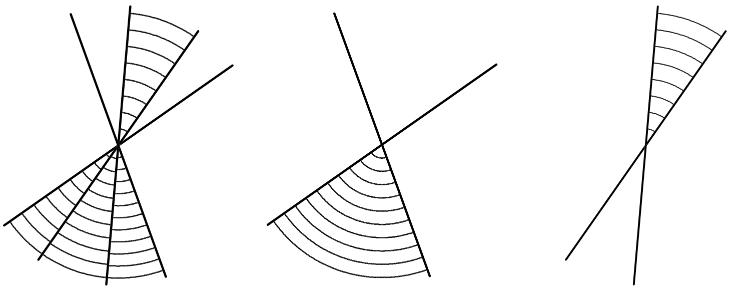

Figure A.2