

**Real-Time Hierarchical Resource
Allocation with Quadratic Payoffs**

by

T. Van Zandt

2003/63/EPS

Working Paper Series

Real-Time Hierarchical Resource Allocation with Quadratic Payoffs

Timothy Van Zandt*
INSEAD

23 July 2003

Abstract

This paper presents a model in which resource allocations are calculated in real time by boundedly rational members of an administrative staff. We consider a class of hierarchical procedures in which information about payoff functions flows up and is aggregated by a hierarchy, while allocations flow down and are disaggregated by the hierarchy with decentralized decision making. We assume that the payoff functions are quadratic and that the payoff parameters follow first-order autoregressive processes. We define a team statistical optimality condition that formalizes the notion of decentralized decision making. We derive a reduced form that can be used to address specific questions about organizational structure and returns to scale.

JEL Classifications: D83, D23

Keywords: decentralization, hierarchies, bounded rationality, resource allocation, real-time computation

Author's address:

INSEAD
Boulevard de Constance
77305 Fontainebleau CEDEX
France

Voice: +33 1 6072 4981
Fax: +33 1 6074 6192
Email: tvz@econ.insead.edu
Web: zandtwerk.insead.edu

*This research was supported in part by grant IRI-9711303 from the National Science Foundation.

Contents

1	Introduction	1
2	A boundedly rational real-time decision model	3
2.1	Overview	3
2.2	The decision problem	4
2.3	The computation model	4
2.4	Computation procedures	6
2.5	Decision procedures	8
3	Hierarchically decomposed decision procedures	9
3.1	Overview on the role of organizational structures	9
3.2	Hierarchical decompositions	10
3.3	Team statistically optimal decision rules	12
4	Statistical assumptions	14
5	A class of hierarchical organizations	15
5.1	CF hierarchies	15
5.2	Decision rules	16
5.3	Payoffs	16
5.4	An example of the calculations of a tier-1 office	18
5.5	Associative computation	19
5.6	Calculation of decision rules	20
5.7	Staying synchronized	23
5.8	Summary	25
6	Returns to scale	25
6.1	Nature of the exercise	25
6.2	Benchmark: Zero managerial cost	26
6.3	Benchmark: Limit on decentralization	27
6.4	Positive managerial cost	27
7	Robustness	28
8	Related literature	30
8.1	Overview	30
8.2	Comparison with Geanakoplos and Milgrom (1991)	30
	Appendix: Proof of Proposition 6.2	32
	References	33

1 Introduction

Consider procedures for allocating resources (e.g., capital) within large organizations such as governments, firms, and universities. The flow of information in these procedures may resemble the hierarchical flow depicted in Figure 1.1. At the bottom of the hierarchy are the production shops, operatives, or whoever ultimately uses the resources. In the upper tiers are managers or administrators, who are independent of the shops. Information about the shops' valuations of the resource is aggregated by a flow of information up the hierarchy. Resources are recursively disaggregated by a flow of information down the same hierarchy. These procedures exhibit both *decentralized information processing*, which means that the resource allocations are calculated jointly by the members of the administrative staff, and *decentralized decision making*, which means that each node makes decisions constraining the resource allocations and that the decisions of different nodes of the hierarchy are calculated using different information.

This is an example of the complex flow of information and of the decentralization of decision making that exist in organizations (and in markets). Such decentralization cannot be explained solely by incentive problems, because an unboundedly rational principal would have no need to delegate decision-making tasks to other agents; if, for other reasons, it is necessary to contract with agents who have private information, then the principal can do no worse than use a direct revelation mechanism in which all agents communicate directly with the principal. Information transmission costs can lead to decentralization of decision making to agents who are exogenously endowed with private information, but such costs cannot alone account for transmission of information through those intermediaries (such as the administrators in Figure 1.1) who are not.

This paper presents a model of such hierarchically decentralized decision making that is driven by the bounded rationality of potential administrators. As in the "batch processing" models of Mount and Reiter (1990), Radner (1993), and Bolton and Dewatripont (1994), bounded rationality is modeled by constraints on the information processing that an agent can perform in a given amount of time. However, rather than modeling how best to calculate a single function, this paper builds a *real-time* model in which resources must be allocated each period and the shops' payoff functions are changing over time. Decision at any point in time can thus be based on information of heterogeneous lags. As a consequence, decentralized decision making can have benefits even in the absence of communication costs. Offices at the bottom of multilevel hierarchies allocate resources to a small number of shops and can thus use disaggregate and hence recent information, while offices at higher levels use more aggregate and hence older information but can still coordinate advantageous transfers between the divisions.

Van Zandt (2003b) presented an abstract version of such a model, which was used to convey basic ideas about decentralization and to compare batch and real-time processing. However, it did not permit the statistical assumptions necessary to calculate the profit for each hierarchical structure. In this sequel, we assume that each operative has a simple quadratic payoff function whose single parameter follows a stationary AR(1) process and is independent of the parameters of the other operatives. Besides being more quantitative, the current paper complements the methodological discussions in Van Zandt (2003b) with more formal definitions of the computation model and hierarchically decomposed decision procedures. We use the static

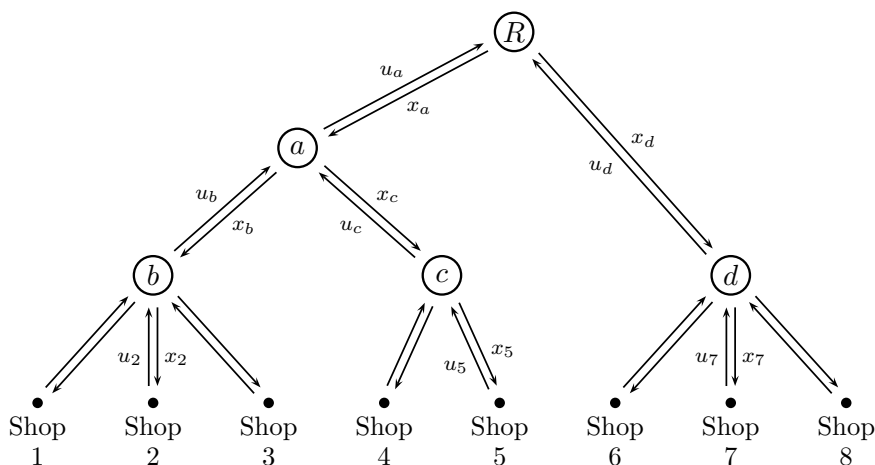


FIGURE 1.1. Hierarchically decomposed decision procedure. Payoff information may be aggregated through an upward flow of information. Allocations are disaggregated through a downward flow of information.

team theory model of hierarchical resource allocation in Geanakoplos and Milgrom (1991) as a tool for suggesting decision rules that take into account the stochastic properties of the payoff parameters and for deriving the expected payoffs for these rules. This team theory model is also used to clarify the meaning of decentralized decision making in our real-time computation model.

We ultimately derive a tractable reduced form. Our analysis of this reduced form is limited to basic results on the optimal scale of integration. There are several papers that establish limits to the optimal scale of centralized decision making (see e.g. Keren and Levhari (1983), Radner and Van Zandt (1992), and Van Zandt and Radner (2001)) even when administrative costs are not taken into account. However, because of the possibility of internal decentralization in our model, there is a limit to the optimal scale of integration only if the managerial wage is positive or if one bounds the extent of decentralization by restricting the number of tiers in the hierarchies.

The model is analyzed further in other work. Van Zandt (2003a) derives a much simpler model by restricting attention to balanced hierarchies (offices in the same tier have the same span and each office aggregates information with the same delay for each source) and making some continuous approximations. That paper also provides considerable “evidence” that optimal hierarchies are balanced, so that restricting attention to balanced hierarchies is without loss of generality. This result relies on the symmetry of the underlying model but does not follow trivially from it.

The reduced-form model of balanced hierarchies is then used in Van Zandt (2003c) to quantify the costs and benefits of decentralized decision making, to obtain results on the optimal scale of organization that are tighter than those reported in this paper, to characterize the optimal spans, and to show how optimal organizations depend on the speed at which the environment changes and on the managerial wage. For example, that paper finds that organizations are smaller and more internally decentralized the more rapidly the environment changes.

The rest of this paper is organized as follows. In Section 2 we specify the temporal resource allocation problem and then the decentralized computation model. The resulting temporal decision model with explicit rationality constraints is very flexible and could be used to represent a variety of market and nonmarket resource allocation procedures; however, we restrict our attention to a class of hierarchical procedures. Some general properties of hierarchical procedures are defined in Section 3. Then, after stating the statistical assumptions in Section 4, we define a specific class of hierarchical procedures in Section 5 and calculate their payoffs and administrative costs. Section 6 characterizes the optimal scale of integration, and Section 7 considers some perturbations to the model. Section 8 reviews some related papers, especially Geanakoplos and Milgrom (1991).

2 A boundedly rational real-time decision model

2.1 Overview

This paper views the administrative staff of an organization as a group of people with bounded computational ability who jointly calculate decisions in a temporal decision problem. This is called real-time decentralized information processing.¹

In Section 2, we specify the two components of such a model:

1. a *decision problem* with uncertainty and multiple decision epochs and infusions of information;
2. a model of the *decision process* by which informational inputs are transformed into decisions.

A *policy* is a decision rule for each decision epoch that is a function of available information. A *computation procedure* is a description of the processing of information. A policy and a computation procedure that calculates the policy are together called a *decision procedure*.

If this paper were about stochastic control by an unboundedly rational decision maker, then we would only specify the decision problem and would characterize policies that maximize the decision-theoretic payoff. Instead, we characterize the performance of decision procedures. Constraining decision making ability introduces an administrative cost of calculating decisions and also limits the set of feasible policies.

The decision problem is to allocate a resource to shops whose valuations of the resource change stochastically; it is described in Section 2.2. For the decision process, we use a simple model of parallel computation; we describe it informally in Section 2.3 and then give a formal axiomatic definition of computation procedures in Section 2.4. Decision procedures and their performance are defined in Section 2.5.

¹See Radner and Van Zandt (1992), Van Zandt (1999), and Van Zandt and Radner (2001) for other examples.

2.2 The decision problem

We consider an organization that allocates a single resource over time to $n > 2$ recipients, which we call *shops* and index by $i \in I$. We let there be a double infinity \mathbb{Z} of time periods (decision epochs) so that we can define stationary decision procedures without having to make exceptions for the first or last few periods. Let $x_{it} \in \mathbb{R}$ be the allocation to shop i in period t ; an allocation in period t is then $\{x_{it}\}_{i \in I}$. The organization's payoff in period t given such an allocation is $\sum_{i \in I} u(x_{it}, \gamma_{it})$, where $u(x_{it}, \gamma_{it})$ is shop i 's payoff and $\gamma_{it} \in \mathbb{R}$ is a random variable called i 's *payoff parameter*.

The payoff parameters are the source of uncertainty in this model; they are also the data from which resource allocations are computed. Realizations of the period- t parameters are freely observable at the beginning of period t , but we model the fact that it takes time for administrators to make decisions using new information. Hence, the allocations in each period are functions of *past* observations of the payoff parameters.

The additivity of the payoff across shops means that there are no externalities. The source of the coordination problem is that the system is closed and hence the resource is in fixed supply. For simplicity, we assume that the resource is perishable and cannot be reallocated intertemporally. Thus, the only intertemporal link in the model is the informational link between data and decisions. We assume that the amount of the resource available is not only deterministic but also the same in each period. Let \bar{x} be the per-shop amount. An allocation $\{x_{it}\}_{i \in I}$ must therefore satisfy $\sum_{i \in I} x_{it} \stackrel{\text{a.s.}}{=} n\bar{x}$, where we use the notation $\stackrel{\text{a.s.}}{=}$ ("almost surely equals") to emphasize that at least one side of an equality is a random variable.

This is the classic resource allocation problem that is so central to economics. The shops may be business units in a firm; the resource affects each unit's profitability given unmodeled decisions made within the unit. The shops may be production shops, where u inversely measures the cost of production; the resource may be an input that lowers each shop's cost given a fixed production assignment, or the resource may represent output and an allocation is an assignment of production targets meant to minimize the total cost of a given level of aggregate output. The shops may be consumers; the resource is then a consumption good that is allocated to maximize total welfare.

2.3 The computation model

As in actual organizations, the administrative staff consists of people (whom we call *agents*) who are hired not because they are exogenously endowed with private information about the organizations' payoff parameters but rather because their time and mental skills are needed in order to make decisions using available information. With the goal of parsimony, we introduce only those constraints on human information processing that are needed for the economic conclusions of the paper. These conclusions depend on *delay in aggregating information*. As explained in Van Zandt (1999), such aggregation delay could be due to either of two limits on human information processing: (a) the time it takes humans to read, understand, and interpret information; or (b) the time it takes humans to calculate using information they have already internalized. However, the delay cannot be due to the time and cost of transmitting information through a physical network. We incorporate only the constraints on

calculation because constraints on “reading and understanding” could be confused with the communication costs that have been used in team theory to explain the decentralization of decision making to agents who are exogenously endowed with private information.

Thus, the agents’ computational abilities are given by a set of elementary operations—functions that can be applied to raw data and prior results—and by the time each operation takes. Otherwise, there are no constraints. For example, agents have unlimited memory, can freely and instantaneously communicate (e.g., read and understand messages), and can synchronously execute the organization’s bureaucratic procedures. Furthermore, agents are identical, are drawn from an unlimited pool, and receive the same per-period wage only while busy performing operations. There are no other managerial costs.² Such parsimony strengthens the results by demonstrating that they can follow from just one limitation on human information processing. We also do not model any incentive problems that might arise, but we note that these would work against decentralization.

One consequence of these assumptions is that an operation can be assigned to any agent who is not otherwise busy because each agent has free and immediate access to all available information, including the results of previous operations. Hence, which agent performs each operation and what messages are sent between individual agents are not determinate. We therefore cannot derive results about the agent-level structure of organizations, but such micro structure is not the subject of this paper.

The appropriate representation of data and the appropriate set of elementary operations both depend on details of the decision problem; these should be rich enough to allow for the computation of a suitably rich set of decision rules. In Section 4, we impose various assumptions on the payoff function u and on the stochastic processes governing the payoff parameters such that linear decision rules are adequate. Hence, we let the set of elementary operations be addition, subtraction, and multiplication. We represent data and allocations as real numbers, as an approximation for fixed-precision arithmetic.³ We assume that these three elementary operations take the same amount of time—namely, one period.

We do not literally consider that constraints on the human ability to do arithmetic are important in organizations. Instead, the simplicity of the information processing tasks (computing linear decision rules) has been forced on us by the adoption of a concrete statistical decision problem with a finite-dimensional state space. The cost and benefits of decentralization that arise in this paper can also be illustrated qualitatively in an abstract model with complex elementary operations (see Van Zandt (2003b)). However, without the additional structure on the decision problem, we cannot obtain a quantitative model and hence can neither characterize optimal organizations nor perform comparative statics. Because economists find simple numerical decision problems to be useful proxies for actual complex economic situations, we must consider the computation of simple decision rules by simple agents to be a useful proxy for the computation of complex decision rules—using soft information and heuristic procedures—by complex human agents. What is important for this paper is that aggregating information takes time, both for the proxy and for real human decision making.

²In computer science, this simple model is called a “parallel random access machine”.

³With the limited set of elementary operations, the subtle complications that can arise with computation on real numbers (dealt with in Mount and Reiter (1996)) do not arise here.

2.4 Computation procedures

Section 2.3 described the computation informally. We could proceed to describe computation procedures informally as well. This can be an appropriate tactic, given the laboriousness of a formal specification. However, owing to the relative novelty of such modeling in economics, we consider it useful to give a formal axiomatic definition of computation procedures.

As discussed in Section 2.3, this model of computation is anonymous in that we cannot pin down the activities of individual agents. This greatly simplifies the formal model. To describe the decisions and managerial costs that result from the calculation of resource allocations, we need only record the operations performed at each point in time, identify the operands of these operations (either payoff parameters, numerical constants, or outputs of previous operations), and then identify which partial results are allocations. In particular, we do not need to record where information resides or who performs each operation.

Imagine the computation of a policy based on the informal description of computation from Section 2.3. Suppose that we form a graph whose nodes are the operations, allocations, data, and constants. Each edge connects an operand to its operation or connects a partial result to an allocation. Order the two edges that go to each operation so that they reflect the ordering of operands (since the order matters for subtraction). We thereby obtain an ordered directed acyclic graph (DAG), which is similar to the DAGs in Mount and Reiter (1990, 1998) and Reiter (1996) and to other types of graphs used to represent and analyze algorithms in computer science. This is how we represent a computation procedure.

We begin by defining a DAG to be any ordered directed acyclic graph (Definition 2.1). We then define some labels, called “attributes”, for identifying what each node in the DAG represents (Definition 2.2). In order to describe a computation procedure, a DAG should respect rules about (a) the number of operands (immediate predecessors) that each node should have and (b) the timing of operations—no operation at time t can have an operand that does not already exist (Definition 2.3). We then specify how the value of a node is determined recursively by the values of its operands (Definition 2.4).

Definition 2.1 A *DAG* is an ordered directed acyclic graph. Each node of a DAG is called a *statement*, and each statement’s immediate predecessors are called its *operands*.

Definition 2.2 An *attribute* is a string of one of the forms

$$\text{ADD}(t) \quad \text{SUB}(t) \quad \text{MUL}(t) \quad \text{ALLOCATION}(t, i) \quad \text{DATA}(t, i) \quad \text{CONSTANT}(r)$$

where $t \in \mathbb{Z}$, $i \in I$, and $r \in \mathbb{R}$. The value of t is called the *execution time*.

The attributes are used to label the statements in a DAG according to what the statements represent. **ADD**, **SUB**, and **MUL** statements represent addition, subtraction, and multiplication (respectively), and are called *operations*. Such statements have two operands each. An **ALLOCATION**(t, i) statement has one operand, which is thereby identified as the period- t allocation for shop i . A **DATA**(t, i) statement stands for γ_{it} and a **CONSTANT**(r) statement stands for the constant r , so that these can be operands of other statements. Such statements do not have their own operands.

This description of operands is formalized in Definition 2.3, which also incorporates the following assumptions about timing: allocations are made and data become available during the instant that begins each period; operations take place during the rest of the period; and constants are timeless.

Definition 2.3 A *computational procedure* is a DAG in which each statement has finitely many predecessors, together with an assignment of an attribute to each statement, such that:

1. operations (ADD, SUB, and MUL statements) have two operands, ALLOCATION statements have one operand, and DATA and CONSTANT statements have no operands; and
2. an operand of a statement with execution time t_1 is either (a) an operation with execution time $t_2 < t_1$, (b) a DATA statement with execution time $t_2 \leq t_1$, or (c) a CONSTANT statement.

We typically denote a computation procedure by \mathcal{P} . Where no confusion can arise, we use \mathcal{P} to denote just the set of statements, and we commonly refer to a statement solely by its attribute.

Definition 2.4 gives substance to our interpretation of the statements by defining, for each statement, a random variable that is called its value and that is a function of the value of its operands.

Definition 2.4 Let \mathcal{P} be a computation procedure. The *value* of each statement $p \in \mathcal{P}$ is a random variable, denoted $v(p)$. These values are defined recursively (starting at CONSTANT and DATA statements) as follows:

Attribute	Operands	Value
CONSTANT(r)		r
DATA(t, i)		γ_{it}
ALLOCATION(t, i)	o	$v(o)$
ADD(t, i)	$o_1 \ \& \ o_2$	$v(o_1) + v(o_2)$
SUB(t, i)	$o_1 \ \& \ o_2$	$v(o_1) - v(o_2)$
MUL(t, i)	$o_1 \ \& \ o_2$	$v(o_1) \times v(o_2)$

The *value* of an edge in the DAG of \mathcal{P} is the value of its initial vertex.

Computation procedures may have infinitely many statements. However, we can illustrate part of a procedure by showing the subgraph containing a finite set of statements and the edges between these statements. For example, Figure 2.1 shows a computation procedure that calculates $\alpha_1 \gamma_{1t} + \alpha_2 \gamma_{2t}$, where α_1, α_2 are constants and it is presumed that $\{1, 2\} \subset I$. Although we cannot pin down the agents involved in this calculation, we can see that there is decentralized information processing. The two MUL operations are performed at the same time and hence must be performed by different agents. Although there are a total of three operations, the delay is two thanks to this decentralization.

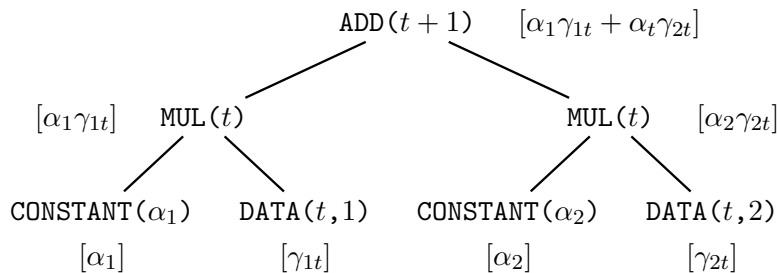


FIGURE 2.1. A procedure for calculating $\alpha_1 \gamma_{1t} + \alpha_2 \gamma_{2t}$. Each node is labeled with its attribute and, in square brackets, its value.

2.5 Decision procedures

A decision procedure is a computation procedure that calculates the resource allocation for each period.

Definition 2.5 A *decision procedure* is a computation procedure \mathcal{P} with the following properties.

1. For all $i \in I$ and $t \in \mathbb{Z}$: there is a unique statement $p_{it} \in \mathcal{P}$ whose attribute is $\text{ALLOCATION}(t, i)$.
2. For all $t \in \mathbb{Z}$: $\sum_{i \in I} v(p_{it}) \stackrel{\text{a.s.}}{=} n\bar{x}$.

Then \mathcal{P} 's *policy* is $\{x_{it} \equiv v(p_{it})\}_{i \in I, t \in \mathbb{Z}}$.

In this paper, we study only those decision procedures for which the expected payoffs and the number of operations in each period are time-invariant. Therefore, we define performance measures only for such decision procedures, rather than specifying a general rule for aggregating payoffs and managerial costs across periods.

Definition 2.6 Let \mathcal{P} be a decision procedure and let $\{x_{it}\}_{i \in I, t \in \mathbb{Z}}$ be the allocation computed by \mathcal{P} . Suppose $\sum_{i \in I} E[u(x_{it}, \gamma_{it})]$ is the same in each period; let U be the constant value. Suppose the number of operations in \mathcal{P} whose execution time is t is the same in each period;⁴ let Y be the constant value. Then U is called \mathcal{P} 's *expected payoff* and Y is called \mathcal{P} 's *administrative load*. There is a parameter $w \geq 0$ called the *managerial wage*, and wY is called \mathcal{P} 's *cost*. We denote \mathcal{P} 's *profit* as $\Pi \equiv U - wY$.

⁴The number of operations executed each period is deterministic because our computation model does not include conditional flow control.

3 Hierarchically decomposed decision procedures

3.1 Overview on the role of organizational structures

The computation model is quite flexible and the set of decision procedures is expansive. The question “which decision procedure has the highest profit?” is well-posed once we add assumptions on payoffs, but it is not easily answered. It may not even be the most interesting question; given the discrete nature of the model, the details of the best decision procedure are likely to obscure more robust qualitative properties. The purpose of this paper is not to answer that question but rather to define a restricted class of decision procedures with interesting identifiable properties. Their comparative advantages and disadvantages are qualitatively robust and quantifiable.

The procedures in that class have hierarchical structures in which the flow of information is as pictured in Figure 1.1. However, the notion of “structure” that we develop is different from that of other models of networks—such as those described by Geanakoplos and Milgrom (1991), Radner (1993), and Mookherjee and Reichelstein (1996)—in which the nodes represent individual agents. Although we used a graph-theoretic construct in the computation model, those DAGs have no direct relationship to organizational structure. Furthermore, we cannot base a notion of structure on the flow of information between individual agents because the assignment of operations to agents is arbitrary and we cannot differentiate the knowledge of different agents.

To understand the role that structure plays in our theory, consider the role it plays in actual organizations. At a micro level (and to an outsider who spends a day in an unfamiliar firm), an organization consists of a collection of people, some coming and going in the same day, with a constant buzz of activities and exchanging of information. An organizational chart or other formal model of the organization’s structure is an attempt to see the forest through the trees—to step back and see order and structure that exist only at a macro level. It is useful both as a way to understand what is actually happening in an organization and as a way to design organizations. The basic component of such a model is not an employee but rather a set of activities or responsibilities, because who performs them may change from day to day and some employees even split their time across several parts of the organization. The distinction between offices, departments, and other parts of the structure is blurry; different observers can come up with different macro models of the same micro activity, depending on what features of the organization they want to emphasize. Yet some models are more compelling than others.

This is the kind of structure that we want to represent in our model. As in real organizations, we limit ourselves to a macro model of hierarchical structures. We use the model not simply to make sense of given decision procedures but also as a framework for designing them. One purpose is to distinguish between decentralized decision making and decentralized information processing. We defend the model not as the only possible representation of the structure of the decision procedures we study but rather as a particularly compelling one.

Definition 3.1 A *hierarchy* is a rooted tree. It is represented by $\langle I, J, R, \{\Theta_j\}_{j \in J} \rangle$, where I is the set of leaves, J is the set of nonleaf nodes, $R \in J$ is the root, and Θ_j is the set of children of $j \in J$. We say that the root is at the top and thus moving away from the root means moving down the hierarchy. For a hierarchy $\langle I, J, R, \{\Theta_j\}_{j \in J} \rangle$, we define the following.

1. Nodes are also called *units*; leaf nodes are also called *shops*; nonleaf nodes are also called *offices*.
2. A child is also called a *subordinate* and a parent is also called a *superior*.
3. For $k_1, k_2 \in I \cup J$: $k_1 \prec k_2$ means that k_1 is below k_2 , and $k_1 \lesssim k_2$ means that either $k_1 \prec k_2$ or $k_1 = k_2$.
4. For $j \in J$: The *span* of j , denoted s_j , is equal to the number of j 's subordinates. That is, $s_j \equiv \#\Theta_j$, where $\#$ denotes the number of elements in a set.
5. For $k \in I \cup J$: *Division* k , denoted θ_k , is equal to the set of leaves below k in the hierarchy. That is, $\theta_k \equiv \{i \in I \mid i \lesssim k\}$. The *size* of division k is $n_k \equiv \#\theta_k$.
6. For $k \in I \cup J$: The *tier* h_k of node $k \in I \cup J$ is the length of the longest path from k to a leaf in division k . Thus, the tier of each leaf is 0, whereas the root has the highest tier, which we denote by H and call the *height* of the hierarchy. Note that, for $j \in J$, $h_j = 1 + \max \{h_k \mid k \in \Theta_j\}$.

TABLE 3.1. Definitions and notation for hierarchies.

3.2 Hierarchical decompositions

We divide our definition of hierarchical structure into two steps.

The first step (Definition 3.2) is meant to capture two ideas and a restriction. The first idea is that the basic unit of an organizational structure is a set of activities or responsibilities. In our model, such a unit is a set of statements in the decision procedure. The second idea is that the interesting pattern of communication between the units of an organization is given not by the physical network through which information is transmitted but rather by who produces and who uses information. In our model, a message is an edge in the DAG connecting two statements that belong to different units. The restriction is that we consider only *hierarchical* structures, meaning that the messages connect the units of the structure to form a tree whose leaves represent shops. (The reader should consult Table 3.1 for a definition of hierarchy; although just a tree, we use special terminology and notation.)

Thus, given a decision procedure, a hierarchical structure is any partitioning of the set of statements such that: the elements of the partition can be identified with nodes of a hierarchy (items 1 and 2 in Definition 3.2); the leaves of the hierarchy correspond to the shops (item 3); and communication is only between subordinates and superiors (item 4).

Definition 3.2 A *hierarchical structure* for a decision procedure \mathcal{P} is $\langle J, R, \{\Theta_j\}_{j \in J}, \{\mathcal{P}_k\}_{k \in I \cup J} \rangle$ such that:

1. $\langle I, J, R, \{\Theta_j\}_{j \in J} \rangle$ is a hierarchy for which $s_j \geq 2$ for $j \in J$;
2. $\{\mathcal{P}_k\}_{k \in I \cup J}$ is a partition of \mathcal{P} ;
3. for each shop $i \in I$, \mathcal{P}_i consists of shop i 's DATA and ALLOCATION statements in \mathcal{P} ;
4. for $k_1, k_2 \in I \cup J$ such that $k_1 \neq k_2$, there is an edge in the DAG from a statement in \mathcal{P}_{k_1} to a statement in \mathcal{P}_{k_2} (called a *message* from k_1 to k_2), or vice versa, if and only if k_1 is a superior or subordinate of k_2 .

For $k \in I \cup J$, the statements in \mathcal{P}_k are said to be *assigned to* or *performed by* k . A message is considered to be sent in the period in which the initial vertex is executed.

The second step (Definition 3.3) relates organizational structure to decision making in an organization. Every relevant message has some effect on organization behavior and hence represents a decision in a weak sense. Yet there is an intuitive distinction between purely informational messages and messages that represent decisions and so constrain others. For example, consider the hierarchical procedure depicted in Figure 1.1. If a unit ignores a message from a subordinate about payoff parameters, the only consequence is that the quality of decisions is degraded. In contrast, a unit cannot ignore a message from a superior about an aggregate allocation without violating feasibility constraints. We identify such resource allocation messages as decisions.

Let \mathcal{P} be a decision procedure and let $\langle J, R, \{\Theta_j\}_{j \in J}, \{\mathcal{P}_k\}_{k \in I \cup J} \rangle$ be a hierarchical structure for \mathcal{P} . For $k \in I \cup J$ and $t \in \mathbb{Z}$, the aggregate allocation to division k is $x_{kt} \equiv \sum_{i \in \Theta_k} x_{it}$. For each period $t \in \mathbb{Z}$, each office $j \in J$, and each subordinate $k \in \Theta_j$, we want to identify a message $X(k, t)$ from j to k whose value is x_{kt} and which we interpret as a decision by j about the aggregate allocation of division k . Furthermore, having identified also the message $X(j, t)$ by which j (if not the root) is informed of its own period- t allocation, we require that office j perform all the operations that lie between $X(j, t)$ and $X(k, t)$ in the DAG; we can then adopt the view that j decides how to subdivide its allocation among its subordinates.

Definition 3.3 A *hierarchical decomposition* of a decision procedure \mathcal{P} is $\langle J, R, \{\Theta_j\}_{j \in J}, \{\mathcal{P}_k\}_{k \in I \cup J}, X \rangle$ such that $\langle J, R, \{\Theta_j\}_{j \in J}, \{\mathcal{P}_k\}_{k \in I \cup J} \rangle$ is a hierarchical structure for \mathcal{P} and such that X is a function from $(I \cup J \setminus \{R\}) \times \mathbb{Z}$ to the set of messages with following properties.

1. For $\langle k, t \rangle \in (I \cup J \setminus \{R\}) \times \mathbb{Z}$: $X(k, t)$ is a message from k 's superior to k .
2. All messages from superiors to subordinates are in the range of X .
3. For $k \in I \cup J \setminus \{R\}$ and $t \in \mathbb{Z}$: $x_{kt} \stackrel{\text{a.s.}}{=} v(X(k, t))$.
4. For $i \in I$ and $t \in \mathbb{Z}$: the attribute of the terminal vertex of $X(i, t)$ is ALLOCATION(t, i).
5. For $j \in J \setminus \{R\}$, $k \in \Theta_j$, and $t \in \mathbb{Z}$: any path in the DAG from the terminal vertex of $X(j, t)$ to the initial vertex of $X(k, t)$ lies in \mathcal{P}_j .

Definition 3.3 is very restrictive. The only messages are between an office and its subordinates, and the only information an office sends to its subordinates are resource allocations. This precludes, for example, that offices in the same tier of a hierarchy share information or that offices send informational statistics that help subordinate offices predict their payoffs. Such information sharing would be advantageous if the payoff parameters of different shops were correlated, and yet could be consistent with an identifiable hierarchical disaggregation of resource allocations. We opted for the restrictive definition because it does not preclude the procedures defined in Section 5.

3.3 Team statistically optimal decision rules

The notion of a hierarchical decomposition as a representation of decentralized decision making still lacks content. For example, every decision procedure has a trivial centralized hierarchical decomposition with a single office; if it has others, how do we identify which is the right one? Furthermore, a hallmark of decentralized decision making is that different offices use different information to calculate the period- t allocations of their subordinates; how can we formally define what information is used in a decision and ensure that it is used in a meaningful way? This section addresses these questions by identifying (in Definition 3.4) the information each office uses to compute allocations and then defining (in Definition 3.5) a distributed “statistical optimality” condition that is drawn from Geanakoplos and Milgrom (1991).

The information set φ_{jt} from which office j computes the period- t allocation of its subordinates is identified roughly as follows. For a subordinate k of j , we already identified a message $X(k, t)$ as j 's decision about k 's period- t allocation. Consider the sub-DAG consisting of $X(k, t)$ and its predecessors; the sources (nodes without predecessors) are **CONSTANT** and **DATA** statements. These data are the information from which the organization computes $X(k, t)$. We do not treat all this information as j 's dataset because offices other than j may perform some of the processing that lies in the DAG between these data and $X(k, t)$. Instead, traversing the DAG backward from $X(k, t)$, we stop each time that we encounter a message to j (i.e., each time we would otherwise leave the set of statements performed by j). We repeat this for $k \in \Theta_j$ and let φ_{jt} be the vector of values of the messages thus found—but excluding j 's own allocation $X(j, t)$ so that we can treat it separately.

Definition 3.4 Let $\langle J, R, \{\Theta_j\}_{j \in J}, \{\mathcal{P}_k\}_{k \in I \cup J}, X \rangle$ be a hierarchical decomposition of a procedure \mathcal{P} .

1. For all $t \in \mathbb{Z}$, $j \in J$, and $k \in \Theta_j$: Office j is said to use a message to calculate k 's period- t allocation if there is a path in the DAG from the final vertex of the message to $X(k, t)$ such that the vertices of the path are all in \mathcal{P}_j .
2. For all $t \in \mathbb{Z}$ and $j \in J$: Let φ_{jt} be the vector of the values of the messages (other than $X(j, t)$ if $j \neq R$) that office j uses to calculate the period- t allocation of at least one of its subordinates. Then φ_{jt} is called j 's *period- t information set*.
3. For all $t \in \mathbb{Z}$ and $j \in J$: For $k \in \Theta_j$, let f_{kt} be the function such that $\langle \varphi_{jt}, x_{jt} \rangle \xrightarrow{f_{kt}} x_{kt}$. Then $\{f_{kt}\}_{k \in \Theta_j, t \in \mathbb{Z}}$ is called office j 's *period- t decision rule*.

Geanakoplos and Milgrom (1991) present a team theory model in which resources are allocated by a recursive disaggregation of allocations down a hierarchy. There is no upward flow of information; instead, offices (which are called managers in their model) are endowed with a set of feasible signals. Other differences between that model and ours is that the former is static and has no constraints on computation. Yet their static “limited-information, unlimited-computation” model is useful for our temporal “unlimited-information, limited-computation” model. For a given hierarchical decomposition and a given period t , we can treat the period- t information of each office as exogenously fixed and ask whether the decision rules for that period would be optimal in the team theory model, given the fixed information structure. We call this criterion *team statistical optimality*.

We first define a version of the model in Geanakoplos and Milgrom (1991) for fixed hierarchies, which we call the *team theory model*. Its exogenous components are:

1. a set I of shops;
2. a per-capita quantity \bar{x} of a resource to be allocated to the shops in I ;
3. for each shop $i \in I$, a payoff function u_i such that, for $x_i \in \mathbb{R}$, $u_i(x_i)$ is a random variable;
4. a hierarchy $\langle I, J, R, \{\Theta_j\}_{j \in J} \rangle$; and
5. an information structure $\{\varphi_j\}_{j \in J}$ where, for $j \in J$, φ_j is a random object that has sample space Φ_j and is called office j 's information set.

A decision rule for office j is a collection $\{f_k: \Phi_j \times \mathbb{R} \rightarrow \mathbb{R}\}_{k \in \Theta_j}$ of functions, where $f_k(\varphi_j, x_j)$ represents j 's allocation to k , that satisfies the following resource constraints:

$$\sum_{k \in \Theta_j} f_k(\varphi_j, x_j) \stackrel{\text{a.s.}}{=} x_j \text{ for } x_j \in \mathbb{R}.$$

Given a decision rule for each office, we can recursively calculate the allocation of each node of the hierarchy as follows. The allocation of the root is $x_R \equiv n\bar{x}$. Given that the allocation of office j is the random variable x_j , it follows that the allocation of $k \in \Theta_j$ is the random variable $x_k \equiv f_k(\varphi_j, x_j)$. The expected payoff is then $\sum_{i \in I} E[u_i(x_i)]$. The decision rules are optimal in this team theory model (for the given hierarchy and information structure) if they yield the highest expected payoff of all decision rules.

Definition 3.5 A decision procedure \mathcal{P} and its hierarchical decomposition $\langle J, R, \{\Theta_j\}_{j \in J}, \{\mathcal{P}_k\}_{k \in I \cup J}, X \rangle$ are said to be *team statistically optimal* if, for all $t \in \mathbb{Z}$, the decision rules $\{\{f_{kt}\}_{k \in \Theta_j}\}_{j \in J}$ are optimal in the team theory model when (i) the set of shops is I , (ii) the per-capita resource is \bar{x} , (iii) for $i \in I$, shop i 's payoff function is $u(\cdot, \gamma_{it})$, (iv) the hierarchy is $\langle I, J, R, \{\Theta_j\}_{j \in J} \rangle$, and (v) the information structure is $\{\varphi_{jt}\}_{j \in J}$.

Optimal hierarchically decomposed decision procedures need not be team statistically optimal. Statistically *suboptimal* decision rules might be less complex or might generate better information for other offices. However, under the statistical assumptions we impose in

Section 4, team statistically optimal decision rules are computationally simple and provide superiors with sufficient statistics for relevant information.

In subsequent sections, team statistical optimality plays three roles. The first is to give credibility to our identification of an office's information by ensuring that the office actually uses this information. The second is to give credibility to our identification of a hierarchical decomposition by ensuring uniqueness. (If we coarsen a hierarchical decomposition then the team statistical optimality condition becomes stronger, because some decisions previously assigned to different offices must now be optimal using pooled information.) The third role is operational: the characterization of team statistically optimal decision rules is useful for designing and describing simple but "statistically aware" decentralized decision procedures.

4 Statistical assumptions

Before presenting (in Section 5) the hierarchically decomposed procedures, we state our statistical assumptions and give a characterization of team statistical optimality.

We assume that the payoff functions are quadratic. This is motivated purely by analytic simplicity. Such payoffs are nonmonotonic and hence are at best a local approximation for most applications.

Assumption 4.1 $u(x, \gamma) = \bar{u} - (x - \gamma)^2$.

We can now obtain a simple characterization of team statistical optimality. Fix a hierarchically decomposed decision procedure. For each office j and period t , let $\gamma_{jt} \equiv \sum_{i \in \Theta_j} \gamma_{it}$ be the sum of the period- t payoff parameters of the shops under j ; we call γ_{jt} the *aggregate payoff parameter* of office or division j . (Note the recursion $\gamma_{jt} = \sum_{k \in \Theta_j} \gamma_{kt}$.) For $j \in J$, $k \in \Theta_j$, and $t \in \mathbb{Z}$, let $\hat{\gamma}_{jt}^j \equiv E[\gamma_{jt} | \varphi_{jt}]$ and $\hat{\gamma}_{kt}^j \equiv E[\gamma_{kt} | \varphi_{jt}]$. (Thus, in the notation $\hat{\gamma}_{kt}^j$ as compared to γ_{kt} , the "hat" indicates that it is an expected value and the superscript j indicates that the expectation is conditional on j 's information.)

The assumption in Proposition 4.1 is that j 's superiors do not have additional information that would help j estimate the aggregate payoff parameters of its subordinates. This holds if office j 's information is a sufficient statistic for γ_{kt} , for $k \in \Theta_j$, with respect to the information of offices above it in the hierarchy.

Proposition 4.1 *Assume that, for $j \in J$ and $k \in \Theta_j$,*

$$E[\gamma_{kt} | \varphi_{jt}] \stackrel{\text{a.s.}}{=} E[\gamma_{kt} | \{\varphi_{\ell t} \mid \ell \in J, \ell \succ j\}].$$

The hierarchically decomposed decision procedure is team statistically optimal if and only if

$$(4.1) \quad f_{kt}(\varphi_{jt}, x_{jt}) \stackrel{\text{a.s.}}{=} \hat{\gamma}_{kt}^j + \frac{n_k}{n_j} (x_{jt} - \hat{\gamma}_{jt}^j)$$

for all $j \in J$, $k \in \Theta_j$, and $t \in \mathbb{Z}$.

The expected payoff each period is then equal to $\Pi^{\text{ni}} + \sum_{j \in J} v_j$, where

$$(4.2) \quad \Pi^{\text{ni}} \equiv n\bar{u} - \frac{1}{n} (n\bar{x} - E[\gamma_{Rt}])^2 - \sum_{i \in I} \text{Var}(\gamma_{it})$$

is the no-information maximized expected payoff and

$$(4.3) \quad v_j \equiv \left(\sum_{k \in \Theta_j} \frac{1}{n_k} \text{Var}(\hat{\gamma}_{kt}^j) \right) - \frac{1}{n_j} \text{Var}(\hat{\gamma}_{jt}^j)$$

is called the value of j 's information. If the estimates $\{\hat{\gamma}_{kt}^j\}_{k \in \Theta_j}$ are independent, then

$$(4.4) \quad v_j = \sum_{k \in \Theta_j} \left(\frac{1}{n_k} - \frac{1}{n_j} \right) \text{Var}(\hat{\gamma}_{kt}^j).$$

PROOF. Geanakoplos and Milgrom (1991) characterized the optimal decision rules for a more general version of this static team theory model. Their characterization is an extension of Crémer (1980). \square

We impose the following statistical assumptions in the rest of this paper.

Assumption 4.2

1. The stochastic processes $\{\{\gamma_{it}\}_{t \in \mathbb{Z}}\}_{i \in I}$ are identically and independently distributed (i.i.d.).
2. For $i \in I$, $\{\gamma_{it}\}_{t \in \mathbb{Z}}$ is a stationary first-order autoregressive process,

$$(4.5) \quad \gamma_{it} = \beta \gamma_{i,t-1} + \epsilon_{it},$$

such that $|\beta| < 1$ and the random variables $\{\epsilon_{it}\}_{t \in \mathbb{Z}, i \in I}$ are i.i.d. and have mean 0.

Under Assumption 4.2, the problem is symmetric with respect to the shops. The mean of γ_{it} is 0. Let $\sigma^2 \equiv \text{Var}(\gamma_{it})$, which is equal to $\text{Var}(\epsilon_{it})/(1 - \beta^2)$. Let $b \equiv \beta^2$.

Consider the no-information benchmark. In this case, each shop's allocation is \bar{x} and the expected payoff is equal to $n(\bar{u} - \bar{x}^2 - \sigma^2)$, from equation (4.2). The constant term \bar{u} does not affect the relative profit of different hierarchies (since it does not figure in equation (4.3)). Therefore, we normalize \bar{u} so that the no-information expected payoff is 0.

Assumption 4.3 $\bar{u} = \bar{x}^2 + \sigma^2$.

5 A class of hierarchical organizations

5.1 CF hierarchies

We study a particular class of hierarchically decomposed decision procedures, called *CF hierarchies*. The ‘‘CF’’ stands for ‘‘constant flow’’, reflecting the fact that the calculations and information flows are the same in each period. Formally, a CF hierarchy is defined to be not the decision procedure itself but rather a list of the key parameters of the decision procedure, as follows.

Definition 5.1 A *CF hierarchy* is a quintuple $\langle I, J, R, \{\Theta_j\}_{j \in J}, \{T_j\}_{j \in J} \rangle$ such that $\langle I, J, R, \{\Theta_j\}_{j \in J} \rangle$ is a hierarchy and, for $j \in J$, T_j is a binary tree with leaves Θ_j (which is called j 's *aggregation tree*).

(By *binary tree* we mean a rooted tree in which every interior node has exactly two children.)

In Sections 5.2–5.7, we define the decision procedure \mathcal{P} and hierarchical decomposition $\langle J, R, \{\Theta_j\}_{j \in J}, \{\mathcal{P}_k\}_{k \in I \cup J}, X \rangle$ corresponding to each CF hierarchy $\langle I, J, R, \{\Theta_j\}_{j \in J}, \{T_j\}_{j \in J} \rangle$ and then derive the profit. We build up the definition step by step so as not to get lost in the details. The binary trees $\{T_j\}_{j \in J}$ are used to describe the way in which information is aggregated, but we do not reach this until Section 5.5.

5.2 Decision rules

The decision procedure and hierarchical decomposition are team statistically optimal, so the policy calculated by each office j has $f_{kt}(\varphi_{jt}, x_{jt}) \stackrel{\text{a.s.}}{=} \hat{\gamma}_{kj}^j + \left(\frac{n_k}{n_j}\right)(x_{jt} - \hat{\gamma}_{jt}^j)$. The stream of partial results $\{\hat{\gamma}_{jt}^j\}_{t \in \mathbb{Z}}$ is the information that office j sends up to its superior. The only information flowing down the hierarchy are the resource allocations.

Thus, office j receives from a subordinate k that is not a shop the stream $\{\hat{\gamma}_{kt}^k\}_{t \in \mathbb{Z}}$. For a subordinate i that is a shop, the stream is the raw data $\{\gamma_{it}\}_{t \in \mathbb{Z}}$. To maintain common notation, we define $\hat{\gamma}_{it}^i \equiv \gamma_{it}$ for $i \in I$ and $t \in \mathbb{Z}$.

Given that the shops' payoff parameters are independent, only those data sent by subordinate k help j estimate γ_{kt} . Let's take as given that the calculations are stationary and that, as a consequence, the stream $\{\hat{\gamma}_{kt'}^k\}_{t' \in \mathbb{Z}}$ is itself a Markov process (this is easily verified once we complete the specification of the calculations). Then office j 's estimate of γ_{kt} uses only one of the messages sent by k , meaning that $\hat{\gamma}_{kt}^j \stackrel{\text{a.s.}}{=} E[\gamma_{kt} | \hat{\gamma}_{k,t-L_{jk}}^k]$ for some integer L_{jk} . Because $\{\gamma_{kt}\}$ is also (This holds if office j 's information is a sufficient statistic for γ_{kt} with respect to the information of offices above it in the hierarchy.) an AR(1) process with coefficient β and $\hat{\gamma}_{k,t-L_{jk}}^k$ is independent of the innovation terms between $t - L_{jk}$ and t (being itself calculated prior to $t - L_{jk}$), we have $\hat{\gamma}_{kt}^j = \beta^{L_{jk}} \hat{\gamma}_{kt}^k$.

5.3 Payoffs

We can think of L_{jk} as the extra lag that j adds to the information about k , owing to as-yet unspecified computational delay. We can calculate the expected payoff in terms of these lags even before we derive the lags from the computation model.

The information for different subordinates is statistically independent and hence so are the estimates $\{\hat{\gamma}_{kt}^j\}_{k \in \Theta_j}$. Therefore, by Proposition 4.1,

$$(5.1) \quad v_j = \sum_{k \in \Theta_j} \left(\frac{1}{n_k} - \frac{1}{n_j} \right) \text{Var}(\hat{\gamma}_{kt}^j).$$

We can derive $\text{Var}(\hat{\gamma}_{kt}^j)$ recursively. Since $\hat{\gamma}_{kt}^j = \beta^{L_{jk}} \hat{\gamma}_{kt}^k$, we have $\text{Var}(\hat{\gamma}_{kt}^j) = b^{L_{jk}} \text{Var}(\hat{\gamma}_{kt}^k)$ (recall that $b \equiv \beta^2$). However, it is useful to derive a nonrecursive formula by viewing office j 's

information as summary statistics of individual payoff parameters. As we move up a hierarchy, each office adds an extra lag to the data. For $k_1, k_2 \in I \cup J$ such that $k_1 \succsim k_2$, we define the *cumulative lag* $L_{k_1 k_2}$ of k_1 's information about k_2 to be the sum of $L_{\ell_1 \ell_2}$ for $\ell_1 \in J$ and $\ell_2 \in \Theta_{\ell_1}$ such that ℓ_1 and ℓ_2 are on the path from k_1 to k_2 in the hierarchy:

$$(5.2) \quad L_{k_1 k_2} \equiv \sum_{\substack{\ell_1 \in J, \ell_2 \in \Theta_{\ell_1} \\ k_1 \succsim \ell_1 \succ \ell_2 \succ k_2}} L_{\ell_1 \ell_2}.$$

(If $k_1 = k_2$, then trivially $L_{k_1 k_2} = 0$; if $k_1, k_2, k_3 \in I \cup J$ and $k_1 \succsim k_2 \succ k_3$, then $L_{k_1 k_3} = L_{k_1 k_2} + L_{k_2 k_3}$.) Then office j 's information consists of summary statistics of $\{\gamma_{i,t-L_{ji}}\}_{i \in \theta_j}$. As shown in Proposition 5.1, these summary statistics are as good as the raw data $\{\gamma_{i,t-L_{ji}}\}_{i \in \theta_j}$ for the purpose of calculating $\hat{\gamma}_{kt}^j$.

Proposition 5.1 *For $j \in J$ and $k \in \Theta_j$,*

$$(5.3) \quad \hat{\gamma}_{kt}^j \stackrel{\text{a.s.}}{=} \sum_{i \in \theta_k} \beta^{L_{ji}} \gamma_{i,t-L_{ji}},$$

$$(5.4) \quad \hat{\gamma}_{jt}^j \stackrel{\text{a.s.}}{=} \sum_{i \in \theta_j} \beta^{L_{ji}} \gamma_{i,t-L_{ji}}.$$

PROOF. The proof is by induction on the tier of the office. As the basis of induction, we note that equation (5.4) would hold trivially if j were in tier 0 and hence a shop. As the inductive step: (a) we show below that equation (5.3) holds for an office j in tier $h \geq 1$ if equation (5.4) holds for all offices and shops in lower tiers; and (b) we note that, if equation (5.3) holds for $k \in \Theta_j$ then equation (5.4) follows from $E\left[\sum_{k \in \Theta_j} \gamma_{kt} \mid \varphi_{jt}\right] \stackrel{\text{a.s.}}{=} \sum_{k \in \Theta_j} E[\gamma_{kt} \mid \varphi_{jt}]$.

Consider then part (a) of the inductive step. Let $h \in \{1, \dots, H\}$ and suppose that equation (5.4) holds for $j \in I \cup J$ such that $h_j < h$. Let $j \in J$ be such that $h_j = h$. For each $k \in \Theta_j$, we have $h_k < h$ and thus equation (5.4) holds when the symbol j is replaced by k in that equation. Since $\hat{\gamma}_{kt}^j = \beta^{L_{jk}} \hat{\gamma}_{kt}^k$, it follows that

$$\hat{\gamma}_{kt}^j = \beta^{L_{jk}} \sum_{i \in \theta_k} \beta^{L_{ki}} \gamma_{i,t-L_{ki}-L_{jk}} = \sum_{i \in \theta_k} \beta^{L_{ji}} \gamma_{i,t-L_{ji}},$$

which is equation (5.3). □

Proposition 5.2 *The expected payoff is $\sum_{j \in J} v_j$, where*

$$(5.5) \quad v_j = \sigma^2 \sum_{k \in \Theta_j} \left(\frac{1}{n_k} - \frac{1}{n_j} \right) \sum_{i \in \theta_k} b^{L_{ji}}.$$

PROOF. In equation (5.6), the first equality follows from Proposition 5.1 and the second follows from the fact that $\{\gamma_{i,t-L_{ji}}\}_{i \in \theta_k}$ are independent.

$$(5.6) \quad \text{Var}(\hat{\gamma}_{kt}^j) = \text{Var}\left(\sum_{i \in \theta_k} \beta^{L_{ji}} \gamma_{i,t-L_{ji}}\right) = \sigma^2 \sum_{i \in \theta_k} b^{L_{ji}}.$$

Substituting equation (5.6) into equation (5.1) yields equation (5.5). □

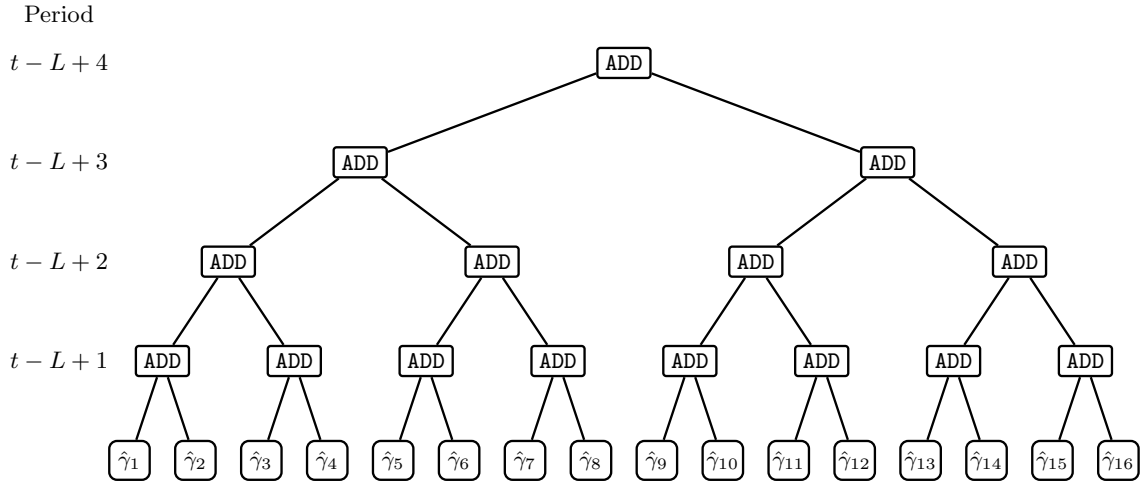


FIGURE 5.1. The sub-DAG for the calculation of $\sum_{k \in \Theta_j} \hat{\gamma}_{kt}^j$ in the example of Section 5.4 ($\hat{\gamma}_k$ represents $\hat{\gamma}_{kt}^j$).

5.4 An example of the calculations of a tier-1 office

Now we get to the actual calculations, which determine the lags and the managerial costs. Consider first an example in which j is an office in tier 1. All subordinates are shops, but we will favor the notation Θ_j for the set of subordinates, s_j for their number, and k for a typical subordinate. Suppose that $s_j = 16$.

Office j should finish calculating $\{x_{kt}\}_{k \in \Theta_j}$ by the beginning of period t . It begins this calculation some time before then, say in period $t - L$. Suppose that it gathers all its data in that period, so that this data is $\{\gamma_{k,t-L}\}_{k \in \Theta_j}$. It has L periods to calculate

$$x_{kt} = \hat{\gamma}_{kt}^j + \frac{1}{n_j}(x_{jt} - \hat{\gamma}_{jt}^j)$$

for each $k \in \Theta_j$, where $\hat{\gamma}_{kt}^j = \beta^L \gamma_{k,t-L}$ and $\hat{\gamma}_{jt}^j = \sum_{k \in \Theta_j} \hat{\gamma}_{kt}^j$. There are five steps.

- (a) The first is to calculate $\hat{\gamma}_{kt}^j = \beta^L \gamma_{k,t-L}$ for each subordinate. There are $s_j = 16$ of these MUL operations, but—by assigning them to different agents within the office—they can all be done during period $t - L$.
- (b) Next, the office must add these 16 partial results to calculate $\hat{\gamma}_{jt}^j = \sum_{k \in \Theta_j} \hat{\gamma}_{kt}^j$. This requires $s_j - 1 = 15$ ADD operations, which cannot be performed at the same time. Still, some decentralization is possible, as seen in the sub-DAG shown in Figure 5.1. The inputs $\{\hat{\gamma}_{kt}^j\}_{k \in \Theta_j}$ are divided into 8 pairs and assigned to different agents, so that the pairs can be summed concurrently in period $t - L + 1$. The eight partial results are then divided into four pairs that are summed concurrently in period $t - L + 2$. The partial results are divided into two pairs that are summed in period $t - L - 3$. The calculation is completed by summing these two partial results in period $t - L - 4$. During the computation, the number of partial results is divided in half each period, so that the answer is obtained in $\log_2 s_j = 4$ periods.

- (c) Then the office subtracts $x_{jt} - \hat{\gamma}_{jt}^j$ in period $t - L + 5$ and
- (d) multiplies $(1/n_j)$ times $(x_{jt} - \hat{\gamma}_{jt}^j)$ in period $t - L + 6$.
- (e) Finally, for each $k \in \Theta_j$, it adds the partial results $\hat{\gamma}_{kt}^j$ and $(1/n_j)(x_{jt} - \hat{\gamma}_{jt}^j)$. These $s_j = 16$ operations can be done concurrently in period $t - L + 7$.

In order for the entire calculation to be finished just in time (by the beginning of period t), we should set $L = 8$. Note that 8 is the total delay or number of periods it took to calculate the allocation.

What we have described are just the calculations of the period- t allocation. There are other calculations going on in office j at the same time. For example, in period $t - 7$, the office begins calculating the allocation for period $t + 1$.

Sections 5.5–5.7 describe the computation in its full generality. Section 5.5 explains how we allow for flexibility in the aggregation of information. Sections 5.6 and 5.7 extend the description to offices in arbitrary tiers; it is similar except for some tricky timing issues.

5.5 Associative computation

The summation $\sum_{k \in \Theta_j} \hat{\gamma}_{kt}^j$ is a key step in the calculations. It is through this *aggregation* of information that coordination of allocations to j 's subordinates takes place—that is, by which the allocation for one subordinate depends on information about other subordinates. Furthermore, this aggregation creates the cumulative lags that cannot be eliminated through decentralized information processing and therefore make decentralized decision making advantageous.

Let T be the sub-DAG for this summation. Then T is necessarily a binary tree whose leaves are the s_j data and whose interior nodes are the $s_j - 1$ ADD operations. It can be balanced (meaning that the depths of the leaves are within 1 of each other as in Figure 5.1) but need not be. Up to isomorphism, any binary tree with s_j leaves is possible; Figure 5.2 shows three examples. The aggregation tree T_j pins down or “describes” how office j aggregates information, as follows: T is isomorphic to T_j , such that the leaf $\hat{\gamma}_{kt}^j$ in T corresponds to the leaf k in T_j .

To understand the lags that aggregation imposes, suppose that $\Theta_j = \{1, \dots, 5\}$ and that j 's only task is to sample each data stream $\{\gamma_{kt}\}_{t \in \mathbb{Z}}$ and compute the sum of the sample by period 0. Let $-L_{jk}$ be the period in which stream k is sampled, so that the lag of this datum is L_{jk} . We time the operations in order to minimize the lags. Then L_{jk} equals the number of ADD operations in the DAG on the path from the datum $\gamma_{k, -L_{jk}}$ to the root, which in turn equals the depth of this datum in the DAG. This is also the depth of k in T_j , which we denote by $\delta(k, T_j)$. For example, if the aggregation tree is T_a in Figure 5.2, then the ADD operations above datum 3 are performed in periods -2 and -1 , respectively, so that datum 3 is first used in period $-\delta(3, T_a) = -2$.

Among binary trees with s_j nodes, the balanced ones have the lowest maximum aggregation delay. However, the profile of depths of such a tree does not dominate that of other binary trees with the same number of leaves. Compare the balanced tree T_a in Figure 5.2, whose profile

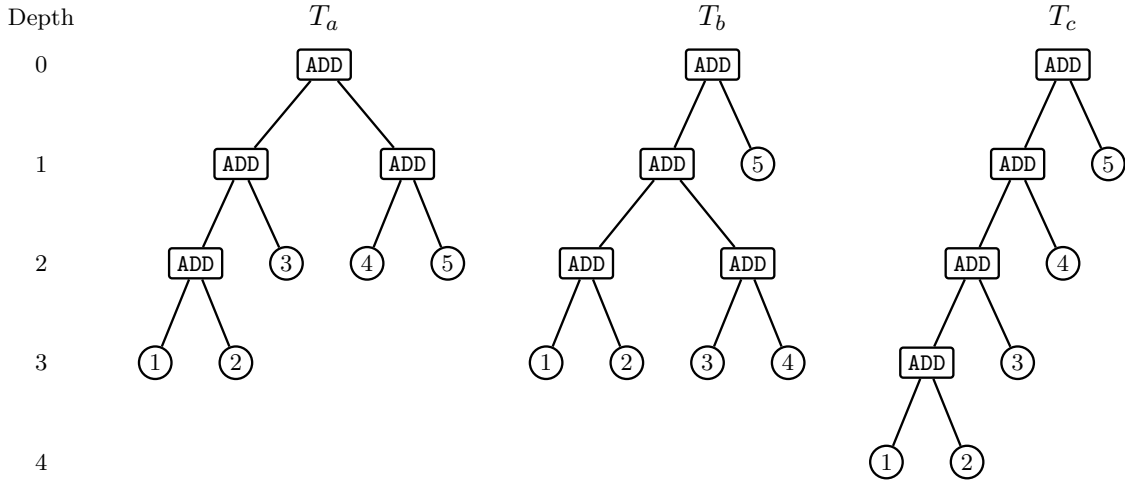


FIGURE 5.2. Three binary trees for aggregating 5 numbers.

of depths is $\{3, 3, 2, 2, 2\}$, with the imbalanced trees T_b and T_c , whose profiles are $\{3, 3, 3, 3, 1\}$ and $\{4, 4, 3, 2, 1\}$, respectively. The maximum depth is higher in T_c than in T_a , but datum 5 has a lower depth in T_c than in T_a . By not restricting attention to balanced aggregation trees, we allow offices to treat subordinates asymmetrically and to use recent data about at least some of the them.

Consider how this aggregation fits into the overall computation of resource allocations. Suppose again that j is the sole office in the hierarchy and there are five shops. Suppose T_j is the binary tree T_a in Figure 5.2. Then the sub-DAG for the calculation of the period- t allocation is shown (along with the times of the operations) in Figure 5.3. The lag L_{jk} is equal to the number of operations on the path from the datum to the allocation. The operations other than the aggregation incur a lag of 4, as in Section 5.4, while the aggregation incurs a lag of $\delta(k, T_j)$, as described in the preceding paragraphs. Thus, $L_{jk} = 4 + \delta(k, T_j)$.

5.6 Calculation of decision rules

Consider the calculation of the period- t decision rule by an arbitrary office $j \in J$. Recall that this decision rule is

$$(5.7) \quad x_{kt} = \hat{\gamma}_{kt}^j + \frac{n_k}{n_j} (x_{jt} - \hat{\gamma}_{jt}^j)$$

for each $k \in \Theta_j$, where $\hat{\gamma}_{kt}^j = \beta^L \gamma_{k,t-L}$ and $\hat{\gamma}_{jt}^j = \sum_{k \in \Theta_j} \hat{\gamma}_{kt}^j$. We extend the example in Section 5.4 by allowing an arbitrary number of subordinates, allowing flexibility in the aggregation of information as described in Section 5.5, and adjusting the timing in case the office is not in tier 1.

We have to adjust the timing if j is not in tier 1 because then j has at least one subordinate that is an office. This subordinate (and other offices below this one, if any) needs time to disaggregate its period- t allocation and hence must learn it before period t . Let τ_j be the integer such that j communicates the period- t allocation to its subordinates in period $t - \tau_j$.

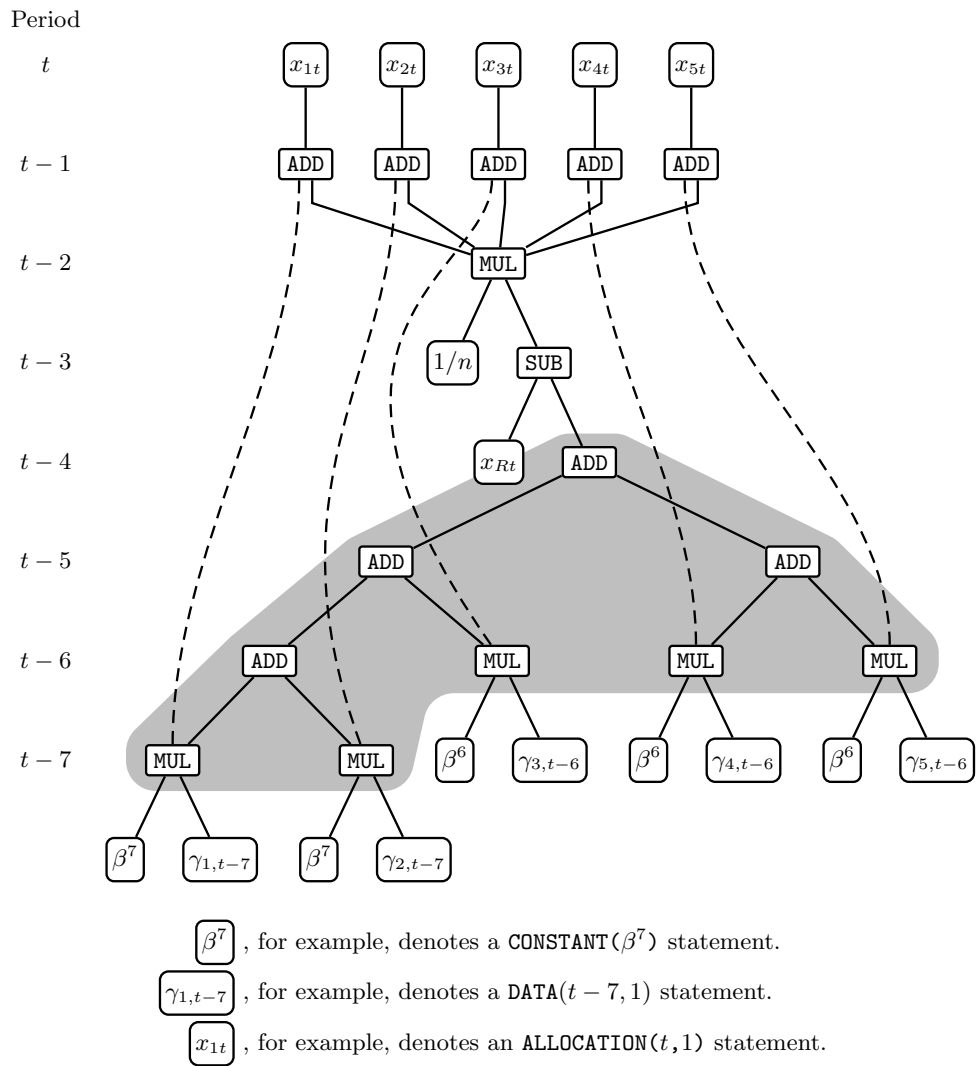


FIGURE 5.3. The subgraph of the DAG of a two-tier hierarchy, containing the statements for the period- t allocation. The shaded area is the subgraph of the calculation of $\sum_{i \in I} \hat{\gamma}_{it}^R$, which is isomorphic to the binary tree T_a in Figure 5.2.

Step	Periods	Calculation	Operations		
	$t - \tau_j - \dots$		Type	#	Delay
(a)	d_{jk}	$\{\hat{\gamma}_{kt}^j := \beta^{L_{jk}} \times \hat{\gamma}_{t-L_{jk}}^k\}_{k \in \Theta_j}$	MUL	s_j	1
(b)	$d_{jk} - 1, \dots, 4$	$\hat{\gamma}_{jt}^j := \sum_{k \in \Theta_j} \hat{\gamma}_{kt}^j$	ADD	$s_j - 1$	$\delta(k, T_j)$
(c)	3	$x_{jt} - \hat{\gamma}_{jt}^j$	SUB	1	1
(d)	2	$\{\Delta_{kt} := \frac{n_k}{n_j} \times (x_{jt} - \hat{\gamma}_{jt}^j)\}_{k \in \Theta_j}$	MUL	s_j	1
(e)	1	$\{x_{kt} := \hat{\gamma}_{kt}^j + \Delta_{kt}\}_{k \in \Theta_j}$	ADD	s_j	1

Total: $4s_j$ $4 + \delta(k, T_j)$

$$\tau_j = 3(h_j - 1)$$

$$d_{jk} = 4 + \delta(k, T_j)$$

$$L_{jk} = d_{jk} + 3(h_j - h_k - 1)$$

TABLE 5.1. The calculations performed by office $j \in J$ for the allocation of resources in period t . In step (b), $\hat{\gamma}_{kt}^j$ is used as an input in period $t - \tau_j - (d_{jk} - 1)$; the entire summation is completed in period $t - \tau_j - 4$.

Taking τ_j as given (we characterize it in Section 5.7), the following describes j 's calculation of equation (5.7). The steps are listed in Table 5.1; here we outline them in reverse order.

- (e) Define $\Delta_{kt} \equiv (n_k/n_j)(x_{jt} - \hat{\gamma}_{jt}^j)$. The last step is to calculate $x_{kt} := \hat{\gamma}_{kt}^j + \Delta_{kt}$ for each $k \in \Theta_j$. There are s_j of these operations, but they are performed concurrently in period $t - \tau_j - 1$.
- (d) In period $t - \tau_j - 2$, Δ_{kt} is calculated by multiplying n_k/n_j and $x_{jt} - \hat{\gamma}_{jt}^j$. There are also s_j of these operations,⁵ which are performed concurrently in period $t - \tau_j - 2$.
- (c) The coefficients $\{(n_k/n_j) \mid k \in \Theta_j\}$ are constants and hence these fractions need not be calculated. Instead, the step that precedes (d) is to calculate $x_{jt} - \hat{\gamma}_{jt}^j$; this single operation is performed in period $t - \tau_j - 3$.
- (b) The preceding step is to calculate the sum $\hat{\gamma}_{jt}^j = \sum_{k \in \Theta_j} \hat{\gamma}_{kt}^j$ so that the calculation is completed just before period $t - \tau_j - 3$. We described how this is done in Section 5.5. The ADD operation p of which $\hat{\gamma}_{kt}^j$ is an operand is executed in period $t - \tau_j - 3 - \delta(k, T_j)$. There are $s_j - 1$ ADD operations in this step.
- (a) The first step is to calculate $\hat{\gamma}_{kt}^j := \beta^{L_{jk}} \hat{\gamma}_{k,t-L_{jk}}^k$ for $k \in \Theta_j$. There are s_j of these MUL operations. Note that $\hat{\gamma}_{kt}^j$ is calculated in period $t - \tau_j - 4 - \delta(k, T_j)$, just before it is used as an operand in step (b).

⁵There may be subordinates $k_1, k_2 \in \Theta_j$ such that $n_{k_1} = n_{k_2}$, in which case this calculation needs only be performed for one of these subordinates. However, we ignore this potential labor-saving improvement and instead always count s_j operations for this step.

The time between (i) when office j uses $\hat{\gamma}_{k,t-L_{jk}}^k$ as an input, which is period $t - \tau_j - 4 - \delta(k, T_j)$, and (ii) when j finishes calculating the period- t allocation of its subordinates, which is period $t - \tau_j$, is thus equal to $4 + \delta(k, T_j)$; we denote this delay by d_{jk} .

5.7 Staying synchronized

We still have to determine τ_j and the lags L_{jk} .

If office j is in tier 1, then the lags simply equal the delays: $L_{jk} = d_{jk}$ for each $k \in \Theta_j$. Office j informs each subordinate (which is a shop) of its period- t allocation in period t , so that $\tau_j = 0$. It uses the datum about subordinate k in period $t - d_{jk}$, at which point $\gamma_{k,t-d_{jk}}$ is the most recent payoff parameter.

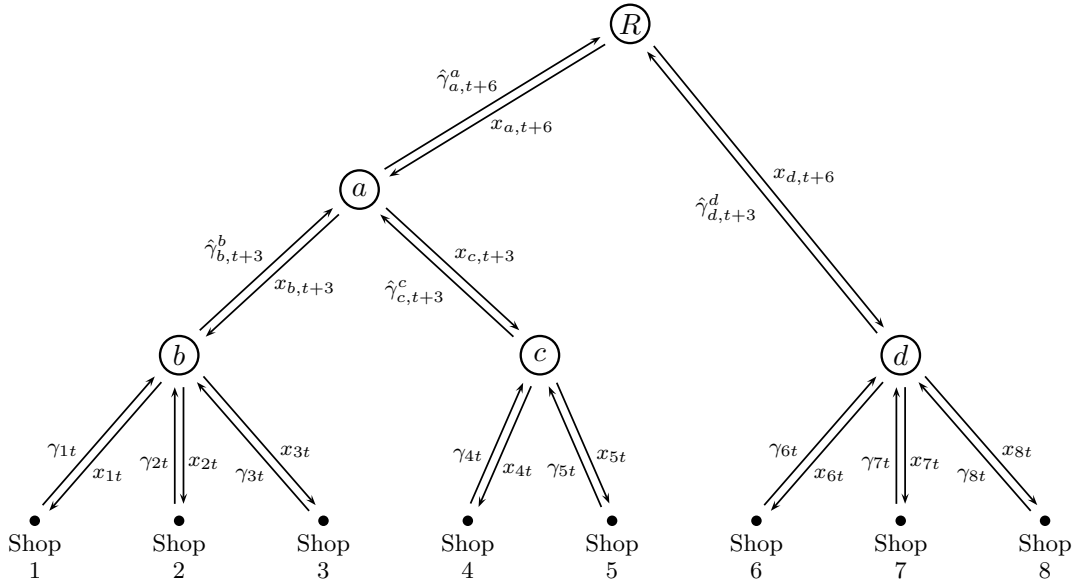
Consider the lead time τ_j for an office that is higher than tier 1. (There is a single lead time rather than a different one for each subordinate because office j finishes calculating the period- t allocations of all subordinates at the same time.) Recall that it takes time for resource allocations to be recursively disaggregated through a hierarchy, so offices further up in the hierarchy must inform their subordinates of their period- t allocations with greater lead time. Specifically, if $j \in J$ and $k \in \Theta_j \cap J$, then j must send x_{kt} to k by the beginning of period $t - \tau_k - 3$, because k uses x_{kt} in step (c) of Table 5.1. Thus, we recursively set $\tau_j = \max\{\tau_k + 3 \mid k \in \Theta_j \cap J\}$; this is equivalent to $\tau_j = 3(h_j - 1)$ for all $j \in J$.

One might expect the lead times to add to the lags so that, for offices higher in the hierarchy, $L_{jk} = d_{jk} + \tau_j$. However, there is a countervailing factor. If $k \in \Theta_j$ is an office, then, for each t , $\hat{\gamma}_{kt}^k$ is available as a partial result following step (b) of Table 5.1, three periods before k finishes calculating the period- t allocation of its subordinates and just before it needs x_{kt} as in input. As a result, typically $L_{jk} = d_{jk}$.

The exception is when subordinate k is located more than one tier below j ; then k is said to skip $h_j - h_k - 1$ levels when reporting to j . (For example, in the hierarchy in Figure 1.1, manager d skips one level when reporting to R but no other managers skip a level.) Subordinate k learns its period- t allocation three periods earlier than necessary for each level skipped, so that $L_{jk} = d_{jk} + 3(h_j - h_k - 1)$.

This “technicality” corresponds to a real need for synchronization. For example, consider a hierarchy that has many tiers but in which the root has one subordinate ℓ that is a shop. Because of the time it takes resource allocations to be disaggregated by the offices in the hierarchy, the center may have to inform its subordinates, including shop ℓ , of their May allocations in February. In February, shop ℓ must stick to its February allocation (which it learned several months before) rather than switching to its May allocation, even though the May allocation is based on more recent information. Otherwise, shop ℓ 's allocation would not be synchronized with those of other shops and the February allocation would not be balanced. (This is just one example of the synchronization that must be achieved in coordination problems. As another example, if a general transmits an order to attack through a hierarchical chain of command, soldiers may receive the order at different times but must move at the same time.)

Further details can be found in the proof of Proposition 5.3. The flow of information in period t is illustrated in Figure 5.4.


 FIGURE 5.4. Flow of information in period t .

Proposition 5.3 Suppose, for $j \in J$, that j 's calculations are as shown in Table 5.1, with the summation $\sum_{k \in \Theta_j} \hat{\gamma}_{kt}^j$ represented by T_j . Suppose further that, for $j \in J$ and $k \in \Theta_j$, the integers τ_j , d_{jk} , and L_{jk} are set to the smallest values such that the set of statements is consistent. Then

$$\begin{aligned}\tau_j &= 3(h_j - 1), \\ d_{jk} &= 4 + \delta(k, T_j), \\ L_{jk} &= d_{jk} + 3(h_j - h_k - 1).\end{aligned}$$

PROOF. We showed that $\tau_j = 3(h_j - 1)$ in the first paragraph of this section; we explained that $d_{jk} = 4 + \delta(k, T_j)$ in Section 5.6.

Let $j \in J$ and $k \in \Theta_j$. Recall that office j needs $\hat{\gamma}_{k,t-L_{jk}}^k$ in period $t - \tau_j - d_{jk}$. If k is a shop then we can set $L_{jk} = \tau_j + d_{jk}$, since $\gamma_{k,t-\tau_j-d_{jk}}$ is available at the beginning of period $t - \tau_j - d_{jk}$. Since $\tau_j = 3(h_j - 1)$ and $h_k = 0$, it follows that $L_{jk} = d_{jk} + 3(h_j - h_k - 1)$.

If instead k is an office, then we set L_{jk} so that k completes the calculation of $\hat{\gamma}_{k,t-L_{jk}}^k$ just before period $t - \tau_j - d_{jk}$. From Table 5.1, we see that k finishes calculating $\hat{\gamma}_{k,t-L_{jk}}^k$ in step (b), three periods before it finishes calculating the period- $(t - L_{jk})$ allocation and hence just before period $(t - L_{jk}) - \tau_k - 3$. Therefore, L_{jk} is the solution to

$$\begin{aligned}t - L_{jk} - \tau_k - 3 &= t - \tau_j - d_{jk} \\ L_{jk} + 3(h_k - 1) + 3 &= 3(h_j - 1) + d_{jk} \\ L_{jk} &= d_{jk} + 3(h_j - h_k - 1).\end{aligned}$$

□

5.8 Summary

This concludes our definition of the organization \mathcal{P} and hierarchical decomposition $\langle J, R, \{\Theta_j\}_{j \in J}, \{\mathcal{P}_k\}_{k \in I \cup J}, X \rangle$ that corresponds to the CF hierarchy \mathcal{H} . We may summarize as follows.

1. For $j \in J$: $\varphi_{jt} = \{\hat{\gamma}_{k,t-L_{jk}}^k\}_{k \in \Theta_j}$, where $L_{jk} = 4 + \delta(k, T_j) + 3(h_j - h_k - 1)$.
2. For $i \in I$: \mathcal{P}_i contains only i 's DATA and ALLOCATION statements.
3. For $j \in J$: \mathcal{P}_j contains the operations listed in Table 5.1 for all $t \in \mathbb{Z}$ as well as CONSTANT statements for any constants used in these operations.
4. For $j \in J$, $k \in \Theta_j$, and $t \in \mathbb{Z}$: the initial vertex of $X(k, t)$ (the message from j specifying k 's period- t allocation) is the ADD operation $\hat{\gamma}_{kt}^j + \Delta_{kt}$ in step (e) of Table 5.1.
5. For $j \in J$ and $k \in \Theta_j$: $\{\hat{\gamma}_{kt}^k\}_{t \in \mathbb{Z}}$ are the only messages sent from k to j and $\{x_{kt}\}_{t \in \mathbb{Z}}$ are the only messages sent from j to k .

Recall that the number of offices in J is denoted by q .

Theorem 5.1 *The per-period administrative load of the CF hierarchy \mathcal{H} is $4(q + n - 1)$. The payoff is $\sum_{j \in J} v_j$, where v_j is defined by equations (5.5) and (5.2) and by $L_{jk} = 4 + \delta(k, T_j) + 3(h_j - h_k - 1)$. Therefore, the profit is*

$$\Pi(\mathcal{H}) \equiv \sum_{j \in J} v_j - 4(n + q - 1)w.$$

PROOF. From Table 5.1, the number of operations per period for office j is $4s_j$. Since $\sum_{j \in J} s_j = n + q - 1$, the administrative load is $4(n + q - 1)$. The formulae for the payoff and the lags come from Propositions 5.2 and 5.3. \square

6 Returns to scale

6.1 Nature of the exercise

As a simple extension to our model, we can allow allocations to be coordinated within multiple independent hierarchies, thereby foregoing coordination and gains from trade between shops in different hierarchies. Call such a collection of CF hierarchies a *CF forest*.

We can then ask whether it is always optimal to group all shops in the same hierarchies, in which case we say that there are uniformly increasing returns to scale, or whether instead there is an upper bound \bar{n} on the number of shops in any CF hierarchy within an optimal CF forest, in which case we say that \bar{n} is a limit to firm size.

If there were no information processing constraints then full integration would be optimal, because larger organizations can take advantage of greater gains from trade and risk sharing. (Under the statistical assumptions, the full-information maximized payoff is $\sigma^2(n - 1)/n$ per

shop.) Thus, it would be significant if this conclusion is reversed by the presence of information processing constraints.

Our tool for answering this question is the net value V_R^{net} of the root of a CF hierarchy, which is the difference between the profit of the CF hierarchy and the total profit of the sub-hierarchies below the root if they were independent. Specifically, consider a CF hierarchy $\langle I, J, R, \{\Theta_j\}_{j \in J}, \{T_j\}_{j \in J} \rangle$. For $j \in J$, let \mathcal{H}_j be the CF hierarchy $\langle \theta_j, J_j, j, \{\Theta_k\}_{k \in J_j}, \{T_k\}_{k \in J_j} \rangle$, where $J_j = \{\ell \in J \mid \ell \preceq j\}$. That is, $\{\mathcal{H}_j\}_{j \in \Theta_R}$ denotes the ‘‘CF subhierarchies’’ coordinated by the root. Then

$$V_R^{\text{net}} \equiv \Pi(\mathcal{H}) - \sum_{j \in \Theta_R} \Pi(\mathcal{H}_j);$$

it is equal to the value of the root’s information minus the root’s administrative cost. That is,

$$(6.1) \quad V_R^{\text{net}} = v_R - 4ws_R,$$

where

$$(6.2) \quad v_R = \sigma^2 \sum_{j \in \Theta_R} \left(\frac{1}{n_j} - \frac{1}{n} \right) \sum_{i \in \theta_j} b^{L_{ji}}.$$

We conclude this subsection with two remarks, as follows.

1. If V_R^{net} is always positive then returns to scale must be uniformly increasing, because we can improve on a CF forest with multiple CF hierarchies by adding a new root who coordinates the allocations to these hierarchies.
2. If there is a size \bar{n} of shops such that V_R^{net} is always negative in a CF hierarchy with more than \bar{n} shops, then \bar{n} is a limit to firm size. This is because we can improve on a CF forest that contains a CF hierarchy with more than \bar{n} shops by eliminating the root of that hierarchy and thus dividing the hierarchy into smaller parts.

6.2 Benchmark: Zero managerial cost

As a benchmark, suppose $w = 0$. The models presented by Keren and Levhari (1983), Radner (1993), and Van Zandt and Radner (2001) do not allow for internal decentralization, which means that all decisions must be made with the same highly lagged information when the organization is large. As a consequence, information processing delay by itself is enough to limit the size of organizations. Yet for the model presented here, if there are multiple independent CF hierarchies then the payoff can necessarily be increased by adding a new center above them—a center that coordinates allocations between these hierarchies without disrupting the existing decentralized decision making. Hence, there are uniformly increasing returns to scale, as stated in Proposition 6.1.

Proposition 6.1 *If $w = 0$ then any optimal CF forest must be fully integrated.*

PROOF. If $w = 0$ then $V_R^{\text{net}} = v_R > 0$. □

6.3 Benchmark: Limit on decentralization

Proposition 6.1 depends on the potential for internal decentralization of decision making. If instead the number of tiers is fixed, then there would be a limit to firm size, as we now show.

For numbers H and n , let $AV_H(n)$ be the maximum per-shop payoff for CF hierarchies of height H with n shops and let $AV_H^* \equiv \sup\{AV_H(n) \mid n = 1, 2, \dots\}$. Note that AV_0^* is defined to be 0, which is trivially the maximum payoff of hierarchies of height 0, because such a hierarchy is a single shop and has no administrative staff.

Proposition 6.2

1. For each $H \in \{1, 2, \dots\}$, $\max\{AV_H(n) \mid n = 1, 2, \dots\}$ exists.
1. The sequence $\langle AV_0^*, AV_1^*, AV_2^*, \dots \rangle$ is strictly decreasing.
1. For $H \in \{1, 2, \dots\}$, $\lim_{n \rightarrow \infty} AV_H(n) = AV_{H-1}^*$.

PROOF. See Appendix. □

What Proposition 6.2 tells us is that, even if $w = 0$, there is a limit to firm size when we fix the number of tiers (part 1). Furthermore, the maximum payoff that can be achieved is higher the more tiers we allow, as this enables greater internal decentralization (part 2). If we force more shops into a hierarchy containing a fixed number of tiers, then the information the center uses is so highly aggregate and hence so old on average that the per-shop value of the center's information processing goes to zero (part 3). These conclusions, by standing in contrast to Proposition 6.1, show that decentralization of decision making (by adding more tiers) is important in order for large organizations to avoid the inexorable degradation of decision-making that results from computational delay.

6.4 Positive managerial cost

Suppose, however, that $w > 0$. As already indicated, in a large firm the information used by the center is necessarily rather old, on average. As a consequence, for a large enough firm, the value of the center's information processing is lower than the cost of its information processing. The profit is then increased by disbanding the center and creating independent hierarchies.

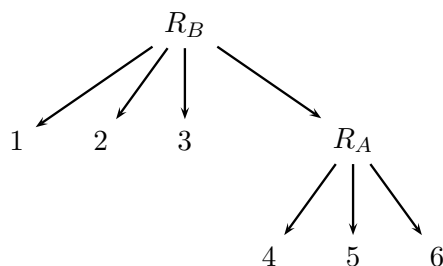
Proposition 6.3 *If $w > 0$ then there is a limit to firm size.*

PROOF. From the definition of v_R and since $L_{Ri} \geq 3H$ for $i \in I$, we have

$$v_R < \sigma^2 \sum_{j \in \Theta_R} \frac{1}{n_j} \sum_{i \in \theta_j} b^{3H} = \sigma^2 b^{3H} s_R.$$

Therefore, the condition, $V_R^{\text{net}} \geq 0$ implies $\sigma^2 b^{3H} s_R > 4ws_R$ and hence $b^{3H} > w/\sigma^2$, or $H < \log(w/\sigma^2)/(3 \log b)$. Thus, there is a bound on the height of a CF hierarchy in an optimal CF forest. According to Proposition 6.2, for $H \in \{1, \dots, \bar{H}\}$ there is also a bound on the size of hierarchies of height H in an optimal CF forest. \square

Compare this result with Geanakoplos and Milgrom (1991), in which returns to scale are uniformly increasing.⁶ In their model, if there are two independent hierarchies A and B , then the payoff can be increased by making the root R_A of hierarchy A a subordinate of the root R_B of hierarchy B , as in the following example,



This is true even when R_B acquires no information about the shops in hierarchy A . The new subordinate does not increase R_B 's administrative costs, and R_B can make advantageous transfers to R_A based only on information about the shops in hierarchy B .

A similar argument may seem, at first, to work in the current model. After merging, R_B 's aggregation tree can be set so that R_A 's aggregate payoff information is processed first and R_B 's processing lags for data from its original immediate subordinates do not change. However, such merging does not necessarily increase the profit in the current model. First, the additional subordinate increases office R_B 's administrative costs. Second, unless the heights H_A and H_B of the two hierarchies are such that $H_A < H_B$, the lag of the data that R_B uses about the shops in hierarchy B increases by $3(H_A - H_B + 1)$ because R_B 's immediate subordinates in hierarchy B skip $H_A - H_B + 1$ additional tiers when reporting to R_B in the new hierarchy.⁷

7 Robustness

The most important and basic theme of this model is that decentralized decision making can be advantageous in group decision problems because it allows for small coordination problems to use recent information without foregoing coordination at a larger scale, which must use more aggregate and hence less recent information. This robust principle is also demonstrated in an abstract version of this hierarchical resource allocation model (see Van Zandt (2003b)).

However, the particular hierarchical procedures defined in the current paper and the formulae for their profits depend on various restriction and assumptions.

⁶In the absence of a restriction to balanced hierarchies; see Van Zandt (1998a).

⁷Both of these effects are present even if we allow R_B to allocate resources to R_A without processing information about the shops in hierarchy A .

First, note that the particularly tractable formulae for the team statistically optimal decision rules and for the profit in this model are entirely dependent on the quadratic payoff functions and on the assumption that the payoff parameters follow AR(1) processes that are independent across shops. Under other assumptions, it may be analytically intractable to derive team statistically optimal decision rules, or the computational complexity of the team statistically optimal decision rules may make them poor candidates for decision rules of hierarchically decomposed decision procedures.⁸ Without the Markovian assumption, there may be no single statistic that an office can send to its superior that adequately summarizes the office's information about its aggregate payoff parameter. Without the statistical independence of payoff parameters of different shops, it may be useful for the upper-level offices to provide their subordinates with statistics that help them estimate their current payoff parameters, or for offices in the same level of the hierarchy to share payoff information directly.

Second, note that—after defining (in Section 2) a real-time decentralized information processing model that could be used to represent a wide variety of decision procedures—we exogenously restricted attention to a class of hierarchical procedures. The set of CF hierarchies is rich enough to exhibit a nexus of decision making, as offices in lower tiers have fewer shops below them and hence control smaller coordination problems using less aggregate and more recent information, while offices in higher tiers coordinate further gains from trade between the subordinate divisions using more aggregate but older information. Furthermore, the decomposition of the decision problem and the flow of information resembles that of various hierarchical decision procedures observed in organizations (e.g., budgeting in firms), and the identification of nodes of the hierarchy with offices rather than individual managers is also consistent with the structure of actual organizations. However, absolutely no claim is being made that these hierarchal procedures have higher profits than all other decision procedures that could have been defined.

That we do not show CF hierarchies dominating all other decision procedures is attributable more to a feature of the model than to a deficiency in the analysis. It is common in the study of organizations to begin with a model that permits the representation of only a limited range of organizational forms. In contrast, the model in this paper is rich and can represent an enormous range of decision procedures. We have been able to derive a reduced-form model of hierarchies with interesting properties, so we have not entirely forsaken simplicity. On the other hand, as an extension to the current research we could also compare these hierarchical procedures with others, such as those mentioned previously or drastically different ones that resemble various market mechanisms or networks of decentralized bilateral trade. Because this resource allocation problem without externalities is the easiest problem for markets to solve, it seems likely that bilateral-trade procedures could have higher profits than the CF hierarchies for some parameter values. We might therefore address the question of when markets are better than hierarchies for allocating resources. It could also be possible to define decision procedures in which the inferred organizational structures change over time.

⁸Nevertheless, in Van Zandt (1996) it is shown that, with logarithmic utility functions, the aggregate payoff functions have simple formulae and certainty equivalence holds.

8 Related literature

These bibliographic notes complement those in Van Zandt (2003b).

8.1 Overview

This paper builds on several strands of the literature on information processing in organizations. We study a resource allocation problem with quadratic payoffs and no externalities, which is a dynamic version of the problem studied by Crémer (1980), Aoki (1986), and Geanakoplos and Milgrom (1991). We use the methodology of dynamic control with real-time parallel computation introduced by Radner and Van Zandt (1992) and Van Zandt (1999) to the study of organizations.

Also related are the models of parallel batch processing, such as Mount and Reiter (1990), Reiter (1996), Radner (1993), Bolton and Dewatripont (1994), Friedman and Oren (1995), Van Zandt (1998b), and Meagher and Van Zandt (1998). As in this paper, managers in these models are described by certain elementary operations and communication capabilities, and the sequentiality of operations and computational delay are important. However, those papers examine the optimal procedures for computing exogenously given computation problems, rather than embedding the computation model in a temporal decision problem. Some ways in which real-time computation is different include: decisions are computed on an ongoing basis; the decision in each period may be computed from data of heterogeneous lags; and partial results may be used in the computation of decisions in multiple periods. All these features can be seen in the hierarchical procedures studied in this paper.

These features of real-time computation were first illustrated in Radner and Van Zandt (1992) and Van Zandt (1999). The decision problem in those papers is to predict the sum of a family of stochastic processes. Information processing is decentralized so that more recent data can be used in the prediction. However, because a single prediction is made each period, there is no room for decentralized decision making. In contrast, the resource allocation problem studied in this paper permits the decentralized decision making that is our main theme.

8.2 Comparison with Geanakoplos and Milgrom (1991)

Crémer (1980) studied a version of the quadratic resource allocation problem that is more general than the one in this paper. He derived the decision rules and expected payoff for a single manager who allocates resources directly to a group of shops. This was extended by Geanakoplos and Milgrom (1991) to the hierarchical disaggregation of resource allocations.

In the general version studied by these authors, x_i and γ_i are vectors and each payoff function may have multiplicative constants (which take the form of matrices in the quadratic form). The heterogeneity of these constants and of the distributions of the payoff parameters play a role in Crémer (1980), who characterizes how differences among shops affect the optimal grouping of shops into divisions. The multiplicity of goods is important in Aoki (1986), who considers suboptimal but simpler decision rules that break up the multivariable optimization problem into single-variable problems.

Heterogeneity of the distributions of payoff functions and multiplicity of goods could be incorporated into the model in this paper. Those generalizations would not affect the basic message about the value of decentralized decision making, but the complementary themes studied by Crémer (1980) and Aoki (1986) would then also arise. Specifically, the asymmetries should affect, within the class of CF hierarchies, which shops should be grouped under the same superior offices. Furthermore, with multiple goods, it would be possible to compare the CF hierarchies with analogous procedures that ignore the cross-partials between different goods. The latter procedures would not be team statistically optimal, but they would use more recent information and would have lower administrative costs.

The model of Geanakoplos and Milgrom (1991) was outlined in Section 3.3. Recall that multilevel hierarchies compute resource allocations in that model, but the computation is based on costly information acquisition (reading of external information) as in team theory. Each node in the hierarchy is called a “manager”. Managers are exogenously restricted from aggregating information. Instead, the only communication in the hierarchy consists of the downward disaggregation of allocations. Managers acquire information about the shops’ payoffs from outside the hierarchy. As is usual in team theory, the managers can compute any functions of their information. However, an important simplifying assumption is that managers do not draw inferences from the allocations they receive, even though these allocations reveal some of their superiors’ information. (Because the managers’ information is endogenous, we cannot simply assume that a manager’s information is a sufficient statistic for her superior’s information for the purpose of predicting the payoff parameters in the manager’s division.) The decision rules and payoffs are then the same as those given in Section 3.3.

The two models have some important common features. First, in very general terms, the value of decentralization is the same in both models: It enables managers or offices lower in the hierarchy to allocate resources within small groups of shops using high-quality, specialized information, while managers or offices higher in the hierarchy can still take advantage of gains from trade between large groups using aggregate information. Second, the model in Geanakoplos and Milgrom (1991) is used in this paper to represent team statistically optimal decentralized decision making and to derive the decision rules and expected payoffs.

However, the details of their static limited-information/unlimited-computation model are quite different from our temporal unlimited-information/limited-computation model. Furthermore, theirs is not a reduced form of ours because the information available to an office in our model depends on the structure of the hierarchy below it and on the calculations of subordinate offices and because an office’s administrative cost depends on how many subordinates it has. In contrast, for the Geanakoplos and Milgrom (1991) model, each manager who may be employed has a fixed set of feasible signals and a fixed wage that do not depend on the structure of the hierarchy in which the manager is employed.

An example of how these differences affect the results of the models is given in Section 6. Another example is given in Van Zandt (1998a, Section 3.4.4), where it is shown that—with the restriction to balanced hierarchies—the statistical assumptions on managers’ information that are needed to obtain a limit to firm size in Geanakoplos and Milgrom (1991) are not satisfied in our model. Hence, there would be no limit to firm size in our model if each office’s cost were independent of the number of subordinates.

Appendix: Proof of Proposition 6.2

Let $\mathbb{N} \equiv \{1, 2, \dots\}$ be the whole numbers.

Lemma A.1 *There is a sequence $\{a_n\}_{n \in \mathbb{N}}$ such that $a_n \rightarrow 0$ and, if $n \in \mathbb{N}$ and v_R is the value of the root's information in a CF hierarchy with n shops, then $a_n \geq v_R/n$.*

PROOF. Let $n \in \mathbb{N}$ and let v_R be the value of the root's information in a CF hierarchy \mathcal{H} with n shops. Then

$$v_R < \sigma^2 \sum_{k \in \Theta_R} \sum_{i \in \theta_k} b^{L_{Ri}} = \sigma^2 \sum_{i \in I} b^{L_{Ri}}.$$

For each lag $L \in \mathbb{N}$, there is a uniform bound on the number of shops about which an office can use data whose lag is L or less. Formally, there is a function $B: \mathbb{N} \rightarrow \mathbb{N}$ such that, for any office j in any CF hierarchy, $\#\{i \in \theta_j \mid L_{ji} \leq L\} \leq B(L)$. The bound is due to the delay in aggregating information. In particular, since in any binary tree at most 2^d nodes can have a depth of d or less, $B(L) = 2^L$ is such a bound. (See Van Zandt (2003a, Appendix B) for further details.)

Let $\{L_i\}_{i=1}^\infty$ be the sequence such that $L_i = 1$ for the first $B(1)$ terms, $L_i = 2$ for the next $B(2)$ terms, and so on. Let $a_n = \sigma^2(1/n) \sum_{i=1}^n b^{L_i}$. Then $v_R/n \leq a_n$. Since the sequence $\{L_i\}$ increases monotonically to infinity, the sequence $\{b^{L_i}\}$ decreases monotonically to 0 and hence $\{a_n\}$ decreases monotonically to 0. \square

For $H \in \mathbb{N}$, let $AV_{\leq H}^* \equiv \max\{AV_{H'}^* \mid 0 \leq H' \leq H\}$.

Lemma A.2 *For $H \in \mathbb{N}$, $\limsup_{n \rightarrow \infty} AV_H(n) \leq AV_{\leq H-1}^*$.*

PROOF. Let $H \in \mathbb{N}$. We show that the statement holds if either $H = 1$ or if $H > 1$ and it holds for H' such that $1 \leq H' < H$.

Let $n \in \mathbb{N}$ and let \mathcal{H} be a CF hierarchy of height H with n shops such that $U(\mathcal{H})/n = AV_H(n)$. For $j \in \Theta_R$, let \mathcal{H}_j be the CF subhierarchy of \mathcal{H} with root j , as defined in Section 6.2. Then $U(\mathcal{H}) = \sum_{j \in \Theta_R} U(\mathcal{H}_j) + v_R$. Since also $n = \sum_{j \in \Theta_R} n_j$, we have

$$AV_H(n) \leq \frac{U(\mathcal{H}_j)}{n} = \frac{\sum_{j \in \Theta_R} U(\mathcal{H}_j)}{\sum_{k \in \Theta_R} n_k} + \frac{v_R}{n}.$$

For any positive numbers $\langle a_1, \dots, a_n, b_1, \dots, b_n \rangle$,⁹

$$\frac{\sum_i a_i}{\sum_i b_i} \leq \max\{a_i/b_i \mid i = 1, \dots, n\}.$$

⁹Suppose $n = 2$ and $a_1/b_1 \geq a_2/b_2$ and hence $a_1 b_2 \geq a_2 b_1$. Then

$$\frac{a_1 + a_2}{b_1 + b_2} = \frac{1}{b_1} \frac{a_1 b_1 + a_2 b_1}{b_1 + b_2} \leq \frac{1}{b_1} \frac{a_1 b_1 + a_1 b_2}{b_1 + b_2} = \frac{a_1}{b_1}.$$

The proof for $n > 2$ then follows by induction.

Therefore,

$$(A.1) \quad AV_H(n) \leq \max_{j \in \Theta_R} \frac{U(\mathcal{H}_j)}{n_j} - \frac{v_R}{n}.$$

For $j \in \Theta_R$, the height of \mathcal{H}_j is at most $H - 1$ and hence $U(\mathcal{H}_j)/n_j \leq AV_{\leq H-1}^*$. Let a_n be the term in the sequence given in Lemma A.1, which is such that $v_R/n \leq a_n$. Combining these two inequalities and equation (A.1), we obtain

$$AV_H(n) \leq AV_{\leq H-1}^* + a_n.$$

Since $\lim_{n \rightarrow \infty} a_n = 0$, we have $\limsup_{n \rightarrow \infty} AV_H(n) \leq AV_{\leq H-1}^*$. \square

PROOF OF PROPOSITION 6.2. By definition, AV_0^* is the maximum payoff of CF hierarchies of height 0. We prove statements 1–3 for $H \in \mathbb{N}$ by induction, assuming that the sequence $\langle AV_0^*, \dots, AV_{H-1}^* \rangle$ is well-defined and strictly increasing. Note that this assumption implies that $AV_{\leq H-1}^* = AV_{H-1}^*$.

The main steps are to prove (i) that $\limsup_{n \rightarrow \infty} AV_H(n) \leq AV_{H-1}^*$, which was accomplished in Lemma A.2 (given also the fact that $AV_{\leq H-1}^* = AV_{H-1}^*$), and (ii) that $\liminf_{n \rightarrow \infty} AV_H(n) \geq AV_{H-1}^*$, which is accomplished below. Then $\lim_{n \rightarrow \infty} AV_H(n) = AV_{H-1}^*$, which is statement 3 in the proposition. That AV_H^* is well-defined and less than AV_{H-1}^* then follows from the fact that there are CF hierarchies of height H whose per-shop payoff is more than AV_{H-1}^* . An example is a CF hierarchy such that the subhierarchies under the root are all isomorphic to a CF hierarchy \mathcal{H}_{H-1} of height $H - 1$ whose per-shop payoff is AV_{H-1}^* .

Thus, it remains to be shown that $\liminf_{n \rightarrow \infty} AV_H(n) \geq AV_{H-1}^*$. Let \mathcal{H}_{H-1} be a CF hierarchy of height $H - 1$ such that $U(\mathcal{H}_{H-1})/n_{H-1} = AV_{H-1}^*$, where n_{H-1} is the number of shops in \mathcal{H}_{H-1} . Let $n \geq n_{H-1}$ and let \mathcal{H} be a CF hierarchy with n shops and height H such that $\lfloor n/n_{H-1} \rfloor$ of the root's subordinates head subhierarchies isomorphic to \mathcal{H}_{H-1} and the remaining $n \bmod n_{H-1}$ subordinates are shops. Then

$$AV_H(n) \geq \frac{U(\mathcal{H})}{n} = \frac{\lfloor n/n_{H-1} \rfloor U(\mathcal{H}_{H-1})}{n}.$$

The limit of the last expression, as $n \rightarrow \infty$, is $U(\mathcal{H}_{H-1})/n_{H-1} = AV_{H-1}^*$. \square

References

- Aoki, M. (1986). Horizontal vs. vertical information structure of the firm. *American Economic Review*, 76, 971–983.
- Bolton, P. and Dewatripont, M. (1994). The firm as a communication network. *Quarterly Journal of Economics*, 109, 809–839.
- Crémer, J. (1980). A partial theory of the optimal organization. *Bell Journal of Economics*, 11, 683–693.
- Friedman, E. J. and Oren, S. S. (1995). The complexity of resource allocation and price mechanisms under bounded rationality. *Economic Theory*, 6, 225–250.

- Geanakoplos, J. and Milgrom, P. (1991). A theory of hierarchies based on limited managerial attention. *Journal of the Japanese and International Economies*, 5, 205–225.
- Keren, M. and Levhari, D. (1983). The internal organization of the firm and the shape of average costs. *Bell Journal of Economics*, 14, 474–486.
- Meagher, K. and Van Zandt, T. (1998). Managerial costs for one-shot decentralized information processing. *Review of Economic Design*, 3, 329–345.
- Mookherjee, D. and Reichelstein, S. (1996). Incentives and decentralization in hierarchies. Department of Economics, Boston University and Haas School of Business, University of California at Berkeley.
- Mount, K. and Reiter, S. (1990). A model of computing with human agents. Discussion Paper No. 890, Center for Mathematical Studies in Economics and Management Science, Northwestern University.
- Mount, K. R. and Reiter, S. (1996). A lower bound on computational complexity given by revelation mechanisms. *Economic Theory*, 7, 237–266.
- Mount, K. R. and Reiter, S. (1998). On modeling computing with human agents. In M. Majumdar (Ed.), *Organizations with Incomplete Information*. Cambridge: Cambridge University Press.
- Radner, R. (1993). The organization of decentralized information processing. *Econometrica*, 62, 1109–1146.
- Radner, R. and Van Zandt, T. (1992). Information processing in firms and returns to scale. *Annales d'Economie et de Statistique*, 25/26, 265–298.
- Reiter, S. (1996). Coordination and the structure of firms. Northwestern University.
- Van Zandt, T. (1996). Organizations with an endogenous number of information processing agents: Supplementary notes. Princeton University.
- Van Zandt, T. (1998a). Organizations with an endogenous number of information processing agents. In M. Majumdar (Ed.), *Organizations with Incomplete Information*. Cambridge: Cambridge University Press.
- Van Zandt, T. (1998b). The scheduling and organization of periodic associative computation: Efficient networks. *Economic Design*, 3, 93–127.
- Van Zandt, T. (1999). Real-time decentralized information processing as a model of organizations with boundedly rational agents. *Review of Economic Studies*, 66, 633–658.
- Van Zandt, T. (2003a). Balancedness of real-time hierarchical resource allocation. INSEAD.
- Van Zandt, T. (2003b). Real-time hierarchical resource allocation. INSEAD.
- Van Zandt, T. (2003c). Structure and returns to scale of real-time hierarchical resource allocation. INSEAD.

Van Zandt, T. and Radner, R. (2001). Real-time decentralized information processing and returns to scale. *Economic Theory*, 17, 497–544.