Balancedness of Real-Time Hierarchical Resource Allocation

Timothy Van Zandt*
INSEAD

23 July 2003

Abstract

We take the hierarchical resource allocation model in Van Zandt (2003a) and derive a simpler, reduced-form model of balanced hierarchies. This model uses continuous approximations; we derive bounds on the errors due to these approximations. We then give results that indicate that optimal hierarchies in the general model of are approximately balanced. In particular, we show that aggregation should be balanced if the hierarchical structure is balanced and we show the hierarchical structure should be balanced if aggregation is balanced.

JEL Classifications: D83, D23
Keywords: decentralization, hierarchies, bounded rationality, resource allocation, real-time computation

Author's address:

INSEAD	Voice:	+33 1 6072 4981
Boulevard de Constance	Fax:	+33 1 6074 6192
77305 Fontainebleau CEDEX	Email:	tvz@econ.insead.edu
France	Web:	zandtwerk.insead.edu

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

Contents

1 Overview

Van Zandt (2003a) develops a model of administrative procedures, called CF hierarchies, for allocating resources in a changing environment. Decision making is decentralized along a hierarchy, that is, a rooted tree whose leaves are passive recipients of resources (called shops) and whose nonleaf nodes are multiperson administrative offices. Each office constantly aggregates information about its subordinates and disaggregates resource allocations to these subordinates. The advantage of such decentralization—compared to having a single office calculate all resource allocation—is that offices lower in the hierarchy use less aggregate and hence more recent information.

In this paper, we develop a simpler model, called BCF hierarchies, by restricting attention to CF hierarchies that are balanced (symmetric) both in their hierarchical structure and in the way each office aggregates information. The goal is to develop and justify a model that is more amenable to analysis and comparative statics than the general model of CF hierarchies. The reduced-form model of BCF hierarchies uses some continuous approximations, and we derive bounds on the resulting errors. In the main results of this paper (Section 4), we provide evidence that optimal CF hierarchies are approximately balanced.

We say "approximately balanced" because the discrete nature of the underlying model is bound to lead to some "leftover" imbalancedness. The evidence in favor of balancedness involves showing that optimal aggregation is balanced if the hierarchical structure is balanced and vice versa. One of the results relies on continuous approximations and a mix of analytic and numerical methods for demonstrating the concavity of a function. Although these results fall short of a proof that optimal CF hierarchies are balanced, they justify studying the reduced-form model of BCF hierarchies for the purpose of characterizing the scale and structure of optimally decentralized organizations. Such an exercise—with a focus on comparative statics with respect to the speed of change of the environment, managerial costs, and information technology—is carried out in Van Zandt (2003b).

2 CF hierarchies

2.1 Parameterization

Van Zandt (2003a) models the administrative apparatus of an organization as a procedure for calculating resource allocations in a changing environment, with the following general features.

- A nonstorable good is allocated each period to a set I of shops, which are indexed by i or k. The intertemporal link is that allocations are computed from past observations of the changing environment.
- Shop i's period-t payoff, as a function of its allocation x_{it}, is $u_{it}(x_{it}) = \bar{u} - (x_{it} - \gamma_{it})^2$, where γ_{it} is a parameter that follows an AR(1) process $\gamma_{it} = \sqrt{b}\gamma_{i,t-1} + \epsilon_{it}$. The variance of γ_{it} is denoted by σ^2. The autoregressive coefficient $\sqrt{b} \in (0, 1)$ is important because it inversely measures the speed at which the environment is changing. The

---

Definition 2.1 A *hierarchy* is rooted tree. It is denoted by $\langle I, J, R, \{\Theta_j\}_{j \in J}\rangle$, where I is the set of leaves (called shops), J is the set of nonleaf nodes (called offices), R is the root (called the center), and, for $j \in J$, Θ_j is the set of j's children (called subordinates).

(a) Denote the number $|I|$ of shops by n and the number $|J|$ of offices by q.

(b) For $k_1, k_2 \in I \cup J$, write $k_1 \succ k_2$ if k_1 is above k_2 in the tree (with the convention that the tree grows down from the root) and write $k_1 \succsim k_2$ if $k_1 \succ k_2$ or $k_1 = k_2$.

(c) For $j \in J$, let $\begin{cases} \text{the } \textit{span} \text{ of office } j \text{ be } s_j \equiv |\Theta_j|; \\ \textit{division } j \text{ be } \theta_j \equiv \{i \in I \mid j \succ i\}; \\ \text{the } \textit{size} \text{ of division } j \text{ be } n_j \equiv |\theta_j|. \end{cases}$

For $i \in I$, let $\theta_i \equiv \{i\}$ and $n_i \equiv 1$.

(d) For $k_1, k_2 \in I \cup J$ such that $k_1 \succsim k_2$, let $P_{k_1k_2}$ be the *path* from k_1 to k_2. ($P_{k_1k_2}$ is the set of pairs $\langle \ell_1, \ell_2\rangle$ such that $\ell_1 \in J$, $\ell_2 \in \Theta_{\ell_1}$, and $k_1 \succsim \ell_1 \succ \ell_2 \succsim k_2$.) $|P_{k_1k_2}|$ is the length of this path.

(e) Define the tier h_k of node $k \in I \cup J$ to be the length of the longest path from k to a leaf below k: $\max\{|P_{ji}| \mid i \in \theta_j\}$. Note that each shop is in tier 0, all the offices are in higher tiers, and the root is in the highest tier, which we denote by H and call the *height* of the hierarchy. Furthermore, for $j \in J$, $h_j = 1 + \max\{h_k \mid k \in \Theta_j\}$.

Table 2.1. Definitions and notation for hierarchies.

---

model is symmetric with respect to the shops in that u_{it} has the same functional form for each shop and the processes $\{\gamma_{it}\}$ are i.i.d. across shops. The constant $\bar{u}$ is normalized so that the payoff in the absence of information processing is zero.

- The computation model is a parallel random access machine, which means that the only constraint is the time it takes to perform elementary operations. The managers performing these calculations are homogeneous.

Van Zandt (2003a) defines a class of decision procedures called CF hierarchies. A decision procedure is literally a distributed computation algorithm, but each CF hierarchy is parameterized by two components.

1. The first component is a hierarchy $\langle I, J, R, \{\Theta_j\}_{j \in J}\rangle$ (a rooted tree) that represents the organizational structure (specifically, the structure of decentralized decision making). In this hierarchy, I is the set of leaves, which are the shops; J is the set of nonleaf nodes, which represent multiperson offices that are the decision-making units; R is the root or center; and, for $j \in J$, Θ_j is the set of children or subordinates of j. The span s_j (number of subordinates) of each office j must be at least 2. (See Table 2.1 for further definitions and notation regarding hierarchies.)

2. The second component is, for each $j \in J$, a binary tree T_j whose leaves are Θ_j. The binary tree T_j describes the way in which office j aggregates information. (See Table 2.2 for definitions and notation regarding binary trees.)

Definition 2.2 A *binary tree* is a tree such that each nonleaf node has exactly two children. Let T be a binary tree.

(a) The *depth* $\delta(k, T)$ of a node k in T is the length of the path from the root to k.

(b) The *depth* $\delta(T)$ of T is the maximum depth of its nodes.

(c) T is *balanced* if the depth of each leaf in T is either $\delta(T)$ or $\delta(T) - 1$.

(d) T is *serial* if no nonleaf node in T has two children that are also nonleaf nodes.

TABLE 2.2. Definitions and notation for binary trees.

Information processing within each office is decentralized, meaning that subtasks can be performed concurrently by different agents in the office. The decisions each office makes concern how to subdivide (among its subordinates) the resource allocations sent from its superior. Each office also aggregates information received from subordinates, both for its own decision making and to pass this information on to its superior.

2.2 Profit

For the policy calculated by a CF hierarchy $\mathcal{H}$, the total expected payoff of the shops is constant over time; this amount is called the *payoff* of $\mathcal{H}$ and is denoted $U(\mathcal{H})$. A CF hierarchy also has an administrative cost $C(\mathcal{H})$. The profit is defined to be $\Pi(\mathcal{H}) \equiv U(\mathcal{H}) - C(\mathcal{H})$.

The key information processing constraint represented in the model is that information processing takes time. The resulting delays can be reduced but not eliminated through parallelization. As a consequence, each office $j \in J$ adds a lag L_{jk} to the information it receives from each subordinate $k \in \Theta_j$, where

$$L_{jk} \equiv \alpha + \delta(k, T_j) + (h_j - h_k - 1)\tau. \tag{2.1}$$

The term $\alpha > 0$ is due to operations that either do not depend on the number of subordinates or can be performed concurrently. The delay that increases with the number of subordinates is due to the aggregation of information, but there is flexibility in how this delay is distributed among the data. The delay in aggregating information received from subordinate k is $\delta(k, T_j)$, which denotes the depth of leaf k in the binary tree T_j. The final term, $(h_j - h_k - 1)\tau$, is not present if k is one tier below j because then $h_j - h_k - 1 = 0$. Otherwise, k is said to skip $h_j - h_k - 1$ levels when reporting j. There is an extra lag of $\tau > 0$ per level skipped that is due to the need to synchronize resource allocations.

Each aggregate datum is a sufficient statistic of the data from which it is calculated (for the use to which it is put), so that these lags are the only loss in data that results from aggregating information up the hierarchy. It is as if office j were using disaggregate information about each shop in its division, but with a lag L_{ji} that is a sum of the lags added to the data by j and the other offices between i and j: $L_{jk} = \sum_{\langle k_1, k_2 \rangle \in P_{ji}} L_{k_1 k_2}$. The payoff can therefore be written as a function of these cumulative lags. It has the form

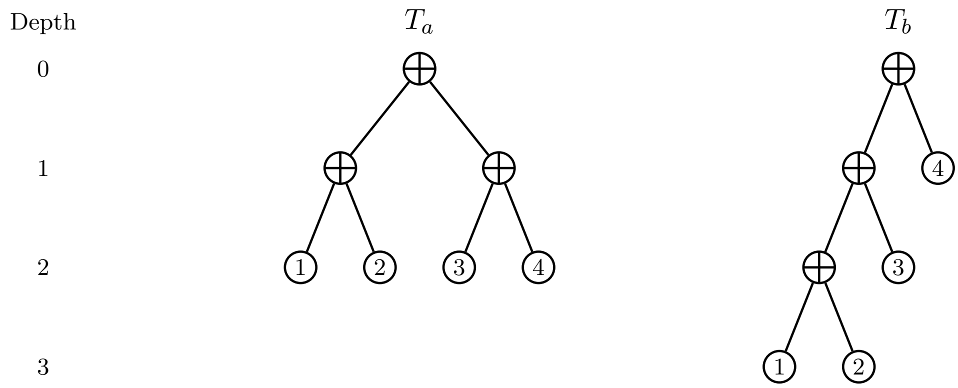

FIGURE 2.1. Two aggregation trees for four data.

$U(\mathcal{H}) = \sum_{j \in J} v_j$, where v_j measures the value of office j's information processing and is equal to

$$v_j \equiv \sigma^2 \sum_{k \in \Theta_j} \left(\frac{1}{n_k} - \frac{1}{n_j} \right) \sum_{i \in \theta_k} b^{L_{ji}}. \tag{2.2}$$

The amount of information processing performed each period is proportional to the number of subordinates in the hierarchy. There are $n + q - 1$ subordinates, since each shop and each office (except the root) is a subordinate. Thus $C(\mathcal{H}) \equiv (n + q - 1)w$, where $w \geq 0$ measures the managerial wage.

We thus have a fully specified formula for the profit of each CF hierarchy. A CF hierarchy is *optimal* if it has highest profit of all CF hierarchies with shops I.

2.3 Aggregation

The part of the calculations performed by a CF hierarchy that we need to understand for this paper is the aggregation of payoff information. It is an associative and commutative binary operation; denote it by $\oplus$. Suppose that one has to aggregate s data $\{X_k \mid k \in \Theta\}$ indexed by the set Θ. The calculation of $\bigoplus_{k \in \Theta} X_k$ can be represented by a binary tree in which the leaves are Θ (representing the data), the nonleaf nodes represent the $s - 1$ operations $\oplus$, and the children of any nonleaf node are the two inputs of that operation. We call such a tree an *aggregation tree*.

Different aggregation trees correspond to different ways in which the data are ordered and operations are grouped. For example, suppose $\Theta = \{1, 2, 3, 4\}$ and consider the two trees in Figure 2.1. The tree T_a represents the calculation $(X_1 \oplus X_2) \oplus (X_3 \oplus X_4)$, whereas T_b represents the calculation $((X_1 \oplus X_2) \oplus X_3) \oplus X_4$.

If the calculation is done in parallel by multiple people or processors, then the aggregation tree determines which operations can be performed concurrently. If the aggregation tree is T_a in Figure 2.1, then $(X_1 \oplus X_2)$ and $(X_3 \oplus X_4)$ can be performed concurrently. The total time it takes to complete the calculation is two periods. If instead the aggregation tree is T_b, then the operations must be performed one at a time; the calculation is completed in three periods. In general, the total delay is equal to the depth of the aggregation tree.

In a model of real-time computation in which there is a constant inflow of new data, what matters is not a single measure of delay but rather the lead time with which each data stream must be sampled before the computation is completed. For a fixed aggregation tree, these lead times (lags) are minimized by performing each operation as late as possible. Then datum $k \in \Theta$ is used $\delta(k, T)$ periods before the computation is completed. For example, suppose the calculation of $X_1 \oplus \cdots \oplus X_4$ must be completed by period t, and consider the two aggregation trees in Figure 2.1. If the aggregation tree is T_a, then all the data are used as inputs in period $t-2$, since $\delta(1, T_a) = \cdots = \delta(4, T_a) = 2$. The two partial results are then aggregated in period $t-1$. In contrast, if the aggregation tree is T_b then the data from sources 1 and 2 are used as inputs in period $t-3$; the data from source 3 is then aggregated with this partial result in period $t-2$, and finally the data from source 4 is aggregated in period $t-1$. (Observe that $\delta(1, T) = \delta(2, T) = 3$, $\delta(3, T) = 2$, and $\delta(4, T) = 1$.) When comparing these two aggregation trees, some data have lower lags and some have higher, so neither tree dominates the other.

This is why the definition of CF hierarchies allows for flexibility in the way the data are aggregated. If T_j is the aggregation tree for the way office j aggregates information about its subordinates for the calculation of each period's resource allocation, then the data about subordinate $k \in \Theta_j$ is used $\delta(k, T_j)$ periods before the aggregation is completed.

3 BCF hierarchies

3.1 Completely balanced CF hierarchies

We develop a simpler model by restricting attention to CF hierarchies that have symmetry or "balancedness" both in their hierarchical structure and in their aggregation of information. Our definition of balanced hierarchies is from Van Zandt (1995).

Definition 3.1 A hierarchy is *completely balanced* (resp., *balanced*) if it has no skip-level reporting and if offices in the same tier have the same spans (resp., have spans that differ by at most 1).

The definition of *completely* balanced hierarchies is very restrictive. Owing to integer constraints, it is typically impossible to equalize spans within a tier just by redistributing subordinates among the offices. Hence, the definition of balanced hierarchies allows for a little "leftover" imbalancedness. Nevertheless, it still precludes skip-level reporting.

By balanced aggregation, we mean that each of the aggregation trees is balanced. As defined in Table 2.2, a binary tree is balanced if the depths of its leaves differ by at most 1. For example, the aggregation tree T_a in Figure 2.1 is balanced whereas T_b is not. The depth of a balanced binary tree with s leaves is $\lceil \log_2 s \rceil$. All leaves have either this depth or a depth of $\lfloor \log_2 s \rfloor$; they have the same depth if and only if s is a power of 2, since then $\log_2 s$ is an integer.

The following definition of a balanced CF hierarchy combines both forms of balancedness.

Definition 3.2 A CF hierarchy $\mathcal{H} \equiv \langle I, J, R, \{\Theta_j\}_{j\in J}, \{T_j\}_{j\in J}\rangle$ is *balanced* if $\langle I, J, R, \{\Theta_j\}_{j\in J}\rangle$ is a balanced hierarchy and if T_j is a balanced binary tree for $j \in J$. If also $\langle I, J, R, \{\Theta_j\}_{j\in J}\rangle$ is completely balanced and if each span is a power of 2, then we say that $\mathcal{H}$ is *completely balanced*.

We will derive a simple functional form for the payoff of a completely balanced CF hierarchy and show that it provides an approximation for the payoff of a balanced CF hierarchy. We begin by introducing the following notation for any balanced hierarchy. Let q_h be the number of nodes in tier $h \in \{0, \dots, H\}$. Thus, $n = q_0 > \cdots > q_H = 1$. For $h \in \{1, \dots, H\}$:

1. let J_h be the set of offices in tier h;
2. let $s_h \equiv q_{h-1}/q_h$ be the "average" span of offices in tier h; and
3. let $n_h \equiv n/q_h$ be the "average" size of a division in tier h.

Note that $s_1 \cdots s_h = n_h$ and that s_h and n_h are integers if the hierarchy is completely balanced.

Consider a completely balanced CF hierarchy and consider an office j in tier h. Because aggregation is balanced and s_h is a power of 2, we have $\delta(k, T_j) = \log_2 s_h$ for $k \in T_j$. Since there is no skip-level reporting, $L_{jk} = \alpha + \log_2 s_h$. We denote this amount by $d_h \equiv \alpha + \log_2 s_h$ and call it the lag or delay of tier h.

Consider again an office j in tier h and let $i \in \theta_j$. Because there is no skip-level reporting, the path from j to i contains one office in each tier $\eta = 1, \dots, h$. Hence,

$$L_{ji} \;=\; \sum_{\eta=1}^{h} d_\eta \;=\; \sum_{\eta=1}^{h} (\alpha + \log_2 s_\eta) \;=\; \alpha h + \log_2(s_1 \cdots s_h) \;=\; \alpha h + \log_2 \frac{n}{q_h}.$$

We denote this quantity by $L_h \equiv \alpha h + \log_2(n/q_h)$ and call it the cumulative lag of tier h. By substituting $L_{ji} = L_h$ into equation (2.2), we obtain

$$(3.1) \qquad \begin{aligned} v_j \;&=\; \sigma^2 \sum_{k\in\Theta_j} \left(\frac{1}{n_k} - \frac{1}{n_j}\right) \sum_{i\in\theta_k} b^{L_h} \\ &=\; \sigma^2 \sum_{k\in\Theta_j} \left(\frac{1}{n_k} - \frac{1}{n_j}\right) n_k b^{L_h} \\ &=\; \sigma^2 \sum_{k\in\Theta_j} \left(1 - \frac{n_k}{n_j}\right) b^{L_h}. \end{aligned}$$

Since the hierarchy is completely balanced, n_k/n_j is equal to $1/s_h$ and there are s_h subordinates in Θ_j, so that

$$(3.2) \qquad v_j \;=\; \sigma^2 s_h (1 - 1/s_h) b^{L_h} \;=\; \sigma^2 (s_h - 1) b^{L_h}.$$

We denote this quantity by v_h.

Since there are q_h offices in tier h and since $q_h s_h = q_{h-1}$, the total value of the information processing by offices in tier h is

$$q_h v_h \;=\; q_h(\sigma^2(s_h-1)b^{L_h}) \;=\; \sigma^2(q_{h-1}-q_h)b^{L_h}.$$

Therefore, the payoff of the hierarchy is

$$U(\mathcal{H}) \;=\; \sigma^2\sum_{h=1}^{H}(q_{h-1}-q_h)b^{L_h}.$$

We thus have the following proposition.

Proposition 3.1 *Assume that $\mathcal{H}$ is a completely balanced CF hierarchy. Then the value of information processing of an office in tier h is*

$$v_h \;=\; \sigma^2(s_h-1)b^{L_h}, \tag{3.3}$$

where $L_h \equiv \alpha h + \log_2(n/q_h)$. The payoff of $\mathcal{H}$ is

$$U(\mathcal{H}) \;=\; \sigma^2\sum_{h=1}^{H}(q_{h-1}-q_h)b^{L_h}. \tag{3.4}$$

The administrative cost of $\mathcal{H}$ is $C(\mathcal{H}) = w\sum_{\eta=0}^{H-1} q_\eta$.

PROOF. See the preceding discussion. □

3.2 Balanced CF hierarchies

If a CF hierarchy is balanced but not completely balanced, then the formulae in Proposition 3.1 are well-defined but hold only approximately. They involve various "continuous approximations". First, if s_j is not a power of 2, then L_{jk} is equal to $\alpha + \lceil \log_2 s_j \rceil$ for some subordinates and to $\alpha + \lfloor \log_2 s_j \rfloor$ for others, whereas the formulae presume that $L_{jk} = \alpha + \log_2 s_j$ for all subordinates. Second, if s_h is not an integer, then $s_j = \lfloor s_h \rfloor$ for some j in tier h and $s_j = \lceil s_h \rceil$ for others, whereas the formulae presume that $s_j = s_h$ for all j in tier h. Similarly, n_j may be slightly larger than n_h for some j in tier h and slightly less than n_h for others, but the formulae presume that $n_j = n_h$ for all j in tier h. We obtain the following bounds on the errors that result from these approximations.

Proposition 3.2 *Assume that $\mathcal{H}$ is a balanced CF hierarchy. Let $h \in \{1,\ldots,H\}$ and let $j \in J_h$. Then, for $i \in \theta_j$,*

$$|L_{ji} - L_h| \;<\; h \tag{3.5}$$

and

$$b^h \;<\; \frac{v_j}{\sigma^2(s_j-1)b^{L_h}} \;<\; b^{-h}. \tag{3.6}$$

Furthermore,

$$b^h \;<\; \frac{U(\mathcal{H})}{\sigma^2\sum_{h=1}^{H}(q_{h-1}-q_h)b^{L_h}} \;<\; b^{-h}. \tag{3.7}$$

PROOF. See Appendix A. □

The formulae for the payoff, cost, and profit of completely balanced CF hierarchies, as shown in Proposition 3.1, depend only on the parameters $\langle n, q_1, \ldots, q_{h-1} \rangle$. We define a reduced-form model, called BCF hierarchies, in terms of these parameters. Based on Proposition 3.2, we view it as an approximate model of balanced CF hierarchies.

Definition 3.3 The *reduced-form model of BCF hierarchies* is as follows. A BCF hierarchy of height H is specified by $\boldsymbol{q} = \langle q_0, q_1, \ldots, q_{H-1} \rangle \in \mathbb{R}_+^H$ such that $q_h/q_{h+1} \geq 2$ for $h = 1, \ldots, H-1$ (where $q_H = 1$). The payoff $U_H(\boldsymbol{q})$ of such a BCF hierarchy is given by equation (3.4), with $n = q_0$. The administrative cost is $C_H(\boldsymbol{q}) \equiv w \left(\sum_{h=0}^{H-1} q_h \right)$ and the profit is $\Pi_H(\boldsymbol{q}) \equiv U_H(\boldsymbol{q}) - C_H(\boldsymbol{q})$.

Given Definition 3.3, it follows that the comparison of BCF hierarchies (to quantify the benefits and costs of decentralization), the characterization of optimal BCH hierarchies, and the returns to scale of BCH hierarchies are well-posed problems; these are taken up in Van Zandt (2003b).

4 Evidence in favor of balanced CF hierarchies

4.1 Overview

Having first defined CF hierarchies and then derived balanced CF hierarchies (or the reduced form BCF hierarchies) as a subclass, a natural question is whether optimal CF hierarchies are balanced. An answer is not required to justify the study of BCF hierarchies; few eyebrows would have been raised if we had, for simplicity, imposed balancedness as a restriction throughout the definition of CF hierarchies. Nevertheless, it would be an interesting characterization of CF hierarchies. Besides telling us when restricting attention to balanced CF hierarchies is nonbinding, it would also tell us when the asymmetries we observe in organizations are due to asymmetries—among managers, activities, and recipients of resources—that are not present in the model of CF hierarchies.

We cannot expect balancedness to be a trivial consequence of the model's symmetry. The literature on organizations contains several examples in which hierarchies are imbalanced even though the underlying model is symmetric. For example, in the batch processing models of Radner (1993) and Bolton and Dewatripont (1994), the optimal hierarchies are highly irregular. Van Zandt (1998, Section 3.2.3) shows that the optimal hierarchies in a symmetric version of Geanakoplos and Milgrom (1991) may also be imbalanced. We will provide yet another example by showing that optimal CF hierarchies are not balanced if $b < 1/2$.

We do not have a calibration of the model that tells us what are plausible values of b, but one would say casually that $b < 1/2$ for an AR(1) process implies that the environment is changing extremely quickly. For the range $b > 1/2$, which we consider more relevant, we have evidence that optimal CF hierarchies are approximately balanced. We say "approximately" because the discreteness of hierarchies will likely lead to some "leftover" skip-level reporting, which is not allowed by our definition of balanced hierarchies.

The evidence consists of two parts:

1. in Section 4.2, we show that optimal aggregation is balanced if we restrict the hierarchy to be completely balanced;

2. in Section 4.4, we show that optimal hierarchies are balanced if we restrict the aggregation to be balanced and prohibit skip-level reporting.

The main gap is that we do not show simultaneously that balanced hierarchies and balanced aggregation are optimal. Furthermore, in the second part, we use continuous approximations and rely on a mix of analytic and numerical methods to check the concavity of a key function. We take a critical look at these gaps in Section 5.

The arguments involve changes to CF hierarchies that do not affect the managerial costs and hence depend only on the formula for the payoff. The variance σ^2 merely scales the payoff; we normalize $\sigma^2 = 1$ for the rest of this paper.

4.2 Balanced aggregation

Optimal aggregation is defined as follows.

Definition 4.1 Two CF hierarchies are *structurally equivalent* if their hierarchies are the same. A CF hierarchy has *optimal aggregation* if there is no structurally equivalent CF hierarchy with a higher payoff.

Note that the managerial cost does not depend on the aggregation trees, which is why this cost does not enter into the definition of optimal aggregation.

Consider first a centralized CF hierarchy, meaning that it has a single office R whose subordinates are thus all shops. We use the notation Θ_R (though this is the same as θ_R and I) to ease the extension of the argument to general CF hierarchies. The payoff of the hierarchy is v_R. Substituting $j = R$, $n_k = 1$, $n_j = n$, $\theta_k = \{k\}$, and $L_{jk} = \alpha + \delta(k, T_R)$ into equation (2.2) yields

$$v_R \;=\; b^\alpha \left(1 - \frac{1}{n}\right) \sum_{k \in \Theta_j} b^{\delta(k,T_R)}.$$

Therefore, if the aggregation tree T_R is chosen to maximize the payoff then it should maximize $\sum_{k \in \Theta_j} b^{\delta(k,T_R)}$.

What aggregation tree maximizes this quantity? Let us rephrase this question abstractly. For any binary tree T, let T° be the leaves of T and, for $b \in (0, 1)$, let $V(T, b) = \sum_{k \in T^\circ} b^{\delta(k,T)}$. For $s \in \{1, 2, \ldots\}$, let $\mathcal{T}_s$ be the set of binary trees with s leaves, and for

$b \in (0,1)$, let $V_s(b) = \max\{V(T,b) \mid T \in \mathcal{T}_s\}$.[1] What binary trees T with s leaves maximize $V(T,b)$ so that $V(T,b) = V_s(b)$?

Note that $d \mapsto b^d$ is a convex function. Hence, if the average depth of the leaves were the same for all binary trees with s leaves, then the trees that maximize $V(T,b)$ could not be balanced because it would be better for the leaves to have diverse depths. However, the average depth of the leaves is lowest for a balanced binary tree. Since $d \mapsto b^d$ is a decreasing function (given $b \in (0,1)$), this factor works in favor of balanced trees. As long as $d \mapsto b^d$ is not too convex—that is, as long as b is not too low—the trees that maximize $V(T,b)$ are balanced, as shown in Lemma 4.1. Otherwise, the trees that maximize $V(T,b)$ are the "opposite" of balanced; they are *serial*, meaning (see Table 2.2) that each nonleaf node has at least one child that is a leaf. For example, the aggregation tree T_b in Figure 2.1 is serial.

Lemma 4.1 *Let $s \in \{1,2,\ldots\}$ and $b \in (0,1)$.*

1. *If $b > 1/2$, then T solves $\max_{T' \in \mathcal{T}_s} V(T',b)$ if and only if T is balanced.*
2. *If $b < 1/2$, then T solves $\max_{T' \in \mathcal{T}_s} V(T',b)$ if and only if T is serial.*
3. *If $b = 1/2$, then $V(T,b) = 1$ for all $T \in \mathcal{T}_s$.*

PROOF. See Lemmas B.3 and B.4 in Appendix B. □

Corollary 4.1 *If $b > 1/2$, then a centralized hierarchy has optimal aggregation if and only if T_R is balanced; if $b < 1/2$, then it has optimal aggregation if and only if T_R is serial; if $b = 1/2$, then any aggregation tree generates the same profit and hence is optimal.*

Note that balanced aggregation entails maximal decentralization of information processing. Serial aggregation entails minimal decentralization; the aggregation can be performed by a single person, since no two operations are concurrent. The model of computation in Van Zandt (2003a) has no communication or other managerial costs to decentralization; hence, in a batch processing model, maximal decentralization is always optimal. Yet in this model decentralized information processing is not optimal (during the aggregation of information) if $b < 1/2$. This is another example of the decision-theoretic cost of decentralizing information processing that was pointed out in Van Zandt (1999): aggregating reports precludes processing current information.

Consider optimal aggregation in an arbitrary CF hierarchy. We will ultimately obtain a result only for completely balanced hierarchies, but we will proceed as far as possible with arbitrary hierarchies in order to show where the difficulty arises.

The following notation will be useful. For any $\ell_1, \ell_2 \in I \cup J$ such that $\ell_1 \succsim \ell_2$, define $L_{\ell_1,\ell_2} \equiv \sum_{\langle k_1,k_2\rangle \in P_{\ell_1\ell_2}} L_{k_1k_2}$. That is, L_{ℓ_1,ℓ_2} is the cumulative lag from ℓ_2 to ℓ_1. This is

[1] To make $\mathcal{T}_s$ well-defined, we should first fix a set of potential nodes for the binary trees. Note that $V_s(b)$ is well-defined because $V(T_1,b) = V(T_2,b)$ if T_1 and T_2 are isomorphic, and hence $\{V(T,b) \mid T \in \mathcal{T}_s\}$ contains finitely many values.

a generalization of our definition of L_{ji} for $j \in J$ and $i \in \theta_j$. If $\ell_1 = \ell_2$, then trivially $L_{\ell_1\ell_2} = 0$. If $\ell_1, \ell_2, \ell_2 \in I \cup J$ and $\ell_1 \succsim \ell_2 \succsim \ell_3$, then $L_{\ell_1\ell_3} = L_{\ell_1\ell_2} + L_{\ell_2\ell_3}$.

Suppose that we modify $\mathcal{H}$ by changing the aggregation tree T_j of a single office $j \in J$. This modification does not affect the sizes of the divisions, which appear in the formulae (2.2) for the values of the offices' information. All it changes are the values of $\{L_{jk}\}_{k\in\Theta_j}$, which affect the value of information for office j and j's superiors but not for any other office. Thus, a necessary condition for $\mathcal{H}$ to have optimal aggregation is that (for each $j \in J$) it is impossible to increase $\sum_{\ell \succsim j} v_\ell$ merely by changing the aggregation tree of office j.

We can decompose the value of information of an office $\ell_1 \in J$ above j so that it is written as a function of $\{L_{ji}\}_{i\in\theta_j}$. Let ℓ_2 be the unique element of Θ_{ℓ_1} such that $\ell_2 \succsim j$. Then

$$\begin{aligned} v_{\ell_1} &= \sum_{k\in\Theta_{\ell_1}\setminus\{\ell_2\}} \left(\frac{1}{n_k} - \frac{1}{n_{\ell_1}}\right) \sum_{i\in\theta_k} b^{L_{\ell_1 i}} \\ &\quad + \left(\frac{1}{n_{\ell_2}} - \frac{1}{n_{\ell_1}}\right) \sum_{i\in\theta_{\ell_2}\setminus\theta_j} b^{L_{\ell_1 i}} + \left(\frac{1}{n_{\ell_2}} - \frac{1}{n_{\ell_1}}\right) \sum_{i\in\theta_j} b^{L_{\ell_1 i}}. \end{aligned} \tag{4.1}$$

The first two terms in equation (4.1) do not depend on $\{L_{jk}\}_{k\in\Theta_j}$ and hence do not depend on T_j. They are denoted by "constant" in equation (4.2). Since $L_{\ell_1 i} = L_{\ell_1 j} + L_{ji}$ for $i \in \theta_j$, we have $\sum_{i\in\theta_j} b^{L_{\ell_1 i}} = b^{L_{\ell_1 j}} \sum_{i\in\theta_j} b^{L_{ji}}$. Hence

$$v_{\ell_1} = \text{constant} + \left(\frac{1}{n_{\ell_2}} - \frac{1}{n_{\ell_1}}\right) b^{L_{\ell_1 j}} \sum_{i\in\theta_j} b^{L_{ji}}. \tag{4.2}$$

Thus, T_j maximizes v_{ℓ_1} if and only if it maximizes $\sum_{i\in\theta_j} b^{L_{ji}}$.

However, this conclusion may not hold for v_j. According to equation (2.2), v_j is a *weighted* sum of $\left\{\sum_{i\in\theta_k} b^{L_{ji}}\right\}_{k\in\Theta_j}$, where the weights are $\left\{\left(\frac{1}{n_k} - \frac{1}{n_j}\right)\right\}_{k\in\Theta_j}$. Maximizing this weighted sum is the same as maximizing $\sum_{i\in\theta_j} b^{L_{ji}}$ if and only if the weights are the same, which in turn holds if and only if n_k is the same for all $k \in \Theta_j$. This is true for an office in tier 1 (since then $n_k = 1$ for all $k \in \Theta_j$) and is true if the hierarchy is completely balanced, but it may not be true otherwise. As a consequence, the payoff-maximizing T_j may depend on the aggregation trees of all the offices superior and inferior to j, and there is typically no *recursive* procedure for selecting the aggregation trees of the offices in a fixed hierarchy in order to maximize the payoff.

Assuming, then, that n_k is the same for all $k \in \Theta_j$, we have that T_j should maximize $\sum_{i\in\theta_j} b^{L_{ji}}$. Observe that $\sum_{i\in\theta_j} b^{L_{ji}} = \sum_{k\in\Theta_j} b^{L_{jk}} \sum_{i\in\theta_k} b^{L_{ki}}$. If also $\sum_{i\in\theta_k} b^{L_{ki}}$ is the same for all $k \in \Theta_j$, then (denoting the common value by B) we have $\sum_{i\in\theta_j} b^{L_{ji}} = B \sum_{k\in\Theta_j} b^{\alpha+\delta(k,T_j)} = Bb^\alpha V(T_j, b)$. Thus, T_j should be chosen to maximize $V(T_j, b)$. This proves the following lemma.

Lemma 4.2 *Fix a CF hierarchy $\mathcal{H}$. Let $j \in J$ and suppose that n_k is the same for each $k \in \Theta_j$. Suppose that T_j is unilaterally modified in order to maximize the payoff. Then T_j should maximize $\sum_{i\in\theta_j} b^{L_{ji}}$ (over the set of aggregation trees for office j). If also $\sum_{k\in\Theta_j} b^{ki}$ is the same for all $k \in \Theta_j$, then T_j should maximize $V(T, b)$.*

Corollary 4.2 *In any CF hierarchy with optimal aggregation, if office j is in tier 1 then $V(T_j, b) = V_{s_j}(b)$.*

We now use Lemma 4.2 to recursively show that, for completely balanced hierarchies, aggregation is optimal if and only if T_j maximizes $V(T_j, b)$, just as we have already shown to hold for a centralized hierarchy and for any tier-1 office in an arbitrary hierarchy.

Lemma 4.3 *Suppose $\langle I, J, R, \{\Theta_j\}_{j\in J}\rangle$ is completely balanced and $V(T_j, b) = V_{s_j}(b)$ for all offices in tiers $1, \ldots, h$. Let j be an office in tier h. Then $\sum_{i\in\theta_j} b^{L_{ji}} = b^{\alpha h} \prod_{\eta=1}^{h} V_{s_\eta}(b)$ and*

$$v_j = \left(\frac{1}{n_{h-1}} - \frac{1}{n_h}\right) b^{\alpha h} \prod_{\eta=1}^{h} V_{s_\eta}(b). \tag{4.3}$$

Proof. Equation (4.3) follows directly from $\sum_{i\in\theta_j} b^{L_{ji}} = b^{\alpha h} \prod_{\eta=1}^{h} V_{s_\eta}(b)$. We prove the latter condition, which we denote by $\mathbf{P}(h)$, by induction. Let j be an office in tier 1. Since $\sum_{k\in\Theta_j} b^{L_{jk}} = b^{\alpha} V(T_j, b)$ and we assume $V(T_j) = V_{s_j}(b)$, it follows that $\mathbf{P}(1)$ holds. Let $h > 1$ and suppose $\mathbf{P}(h-1)$ holds. Let j be an office in tier h. Then

$$\sum_{i\in\theta_j} b^{L_{ji}} = \sum_{k\in\Theta_j} b^{L_{jk}} \sum_{i\in\theta_k} b^{L_{ki}} = b^{\alpha(h-1)} \prod_{\eta=1}^{h-1} V_{s_\eta}(b) \sum_{k\in\Theta_j} b^{L_{jk}} = b^{\alpha h} \prod_{\eta=1}^{h} V_{s_\eta}(b).$$

Thus, $\mathbf{P}(h)$ holds. □

Proposition 4.1 *Suppose $\langle I, J, R, \{\Theta_j\}_{j\in J}\rangle$ is completely balanced. Then $\mathcal{H}$ has optimal aggregation if and only if $V(T_j, b) = V_{s_j}(b)$ for all $j \in J$.*

Proof. We show by induction on the tiers that $V(T_j, b) = V_{s_j}(b)$ for all $j \in J$ is a necessary condition for optimality. Sufficiency then follows because, according to equation (4.3) in Lemma 4.3, all structurally equivalent hierarchies with that property have the same payoff.

The basis of induction is Corollary 4.2. The inductive step is as follows. Let $h \in \{2, \ldots, H\}$, and suppose $V(T_j, b) = V_{s_j}(b)$ for all offices in tiers lower than h. Then, by Lemma 4.3, $\sum_{i\in\theta_k} b^{L_{ki}}$ is the same for all offices k in tier $h-1$. By Lemma 4.2, $V(T_j, b) = V_{s_j}(b)$ for all offices j in tier h. □

Corollary 4.3 *Suppose that $\mathcal{H} \equiv \langle I, J, R, \{\Theta_j\}_{j\in J}, \{T_j\}_{j\in J}\rangle$ is a CF hierarchy such that $\langle I, J, R, \{\Theta_j\}_{j\in J}\rangle$ is completely balanced.*

1. *If $b > 1/2$ (resp., $b < 1/2$), then $\mathcal{H}$ has optimal aggregation if and only if T_j is balanced (resp., serial) for all $j \in J$.*
2. *If $b = 1/2$, then all CF hierarchies that are structurally equivalent to $\mathcal{H}$ have the same payoff.*

Proof. The corollary follows from Proposition 4.1 and Lemma 4.1. □

4.3 Subhierarchies that maximize the value of superiors' information

We approach the question of whether optimal hierarchies are balanced. We begin with a step that ends up closely related to our characterization of optimal aggregation.

Suppose that, given a CF hierarchy, we "redesign" one of the subhierarchies—keeping fixed its root and shops—in order to maximize the total profit. Modifying the subhierarchy affects the value of the information of the offices in the subhierarchy and also their administrative cost; the impact outside the subhierarchy is only on the value of information of offices superior to the subhierarchy. We characterize this impact, under the restrictions that the subhierarchy has no skip-level reporting and that the height of the subhierarchy remain fixed.

Let j be the root of the subhierarchy. Let ℓ_1 be an office above j. Let ℓ_2 be the unique subordinate of ℓ_1 such that $\ell_2 \succsim j$. Recall from equation (4.2) in Section 4.2 that we can write

$$v_{\ell_1} \;=\; \textit{constant} + \left(\frac{1}{n_{\ell_2}} - \frac{1}{n_{\ell_1}}\right) b^{L_{\ell_1 j}} \sum_{i \in \theta_j} b^{L_{ji}}, \tag{4.4}$$

where the constant term does not depend on the subhierarchy with root j. Hence, this subhierarchy maximizes v_{ℓ_1} if and only if it maximizes $\sum_{i\in\theta_j} b^{L_{ji}}$.

There is a nonrecursive way to write the cumulative lag L_{ji} that will help us to see how $\sum_{i\in\theta_j} b^{L_{ji}}$ depends on the subhierarchy rooted at j. We restrict the subhierarchy at j to have no skip-level reporting and to have height h_j. Then

$$L_{ji} \;=\; \sum_{\langle k_1,k_2\rangle \in P_{ji}} \alpha + \delta(k_2, T_{k_1}) \;=\; \alpha h_j + \sum_{\langle k_1,k_2\rangle \in P_{ji}} \delta(k_2, T_{k_1}).$$

Define $\hat{T}_j$ to be the tree constructed, starting with T_j, by recursively replacing each leaf k that is not a shop with T_k. The binary tree $\hat{T}_j$ is called j's *cumulative aggregation tree*; it represents the way in which, collectively, j and the offices that are subordinate to j aggregate cost information about the shops in θ_j. (Figures 4.1 and 4.2 show two examples. At the top of each figure is a hierarchy; at the bottom is the cumulative aggregation tree of the root, presuming that each office's aggregation tree is balanced.) It is straightforward that $\sum_{\langle k_1,k_2\rangle\in P_{ji}} \delta(k_2, T_{k_1}) = \delta(i, \hat{T}_j)$ and hence that $L_{ji} = \alpha h_j + \delta(i, \hat{T}_j)$.

We can therefore rewrite equation (4.4) as

$$v_{\ell_1} \;=\; \text{constant} + \left(\frac{1}{n_{\ell_2}} - \frac{1}{n_{\ell_1}}\right) b^{L_{\ell_1 j}} b^{\alpha h_j} V(\hat{T}_j, b).$$

In order to maximize v_{ℓ_1}, the subhierarchy should be designed to maximize $V(\hat{T}_j, b)$. For example, if $b > 1/2$ then v_{ℓ_1} is maximized by making $\hat{T}_j$ balanced. This result is summarized Proposition 4.2.

Proposition 4.2 *Consider a CF hierarchy. Let j be an office other than the root and let ℓ be an office above j. Suppose that the subhierarchy with root j is designed in order to maximize v_ℓ, keeping fixed the number of shops and the height of the subhierarchy and precluding skip-level reporting. Then it should be designed to maximize $V(\hat{T}_j, b)$.*

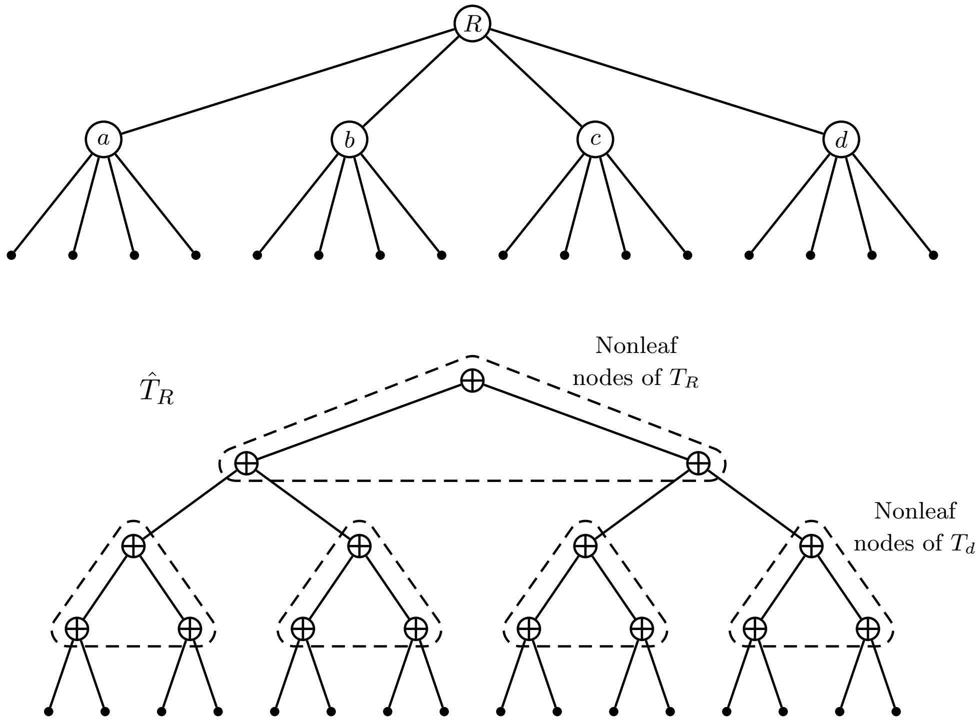

FIGURE 4.1. A hierarchy is shown at the top. If the aggregation tree of each office is balanced, then the cumulative aggregation tree of the root is isomorphic to the binary tree at the bottom. Because both the hierarchy and the aggregation trees are balanced, the cumulative aggregation tree is balanced.

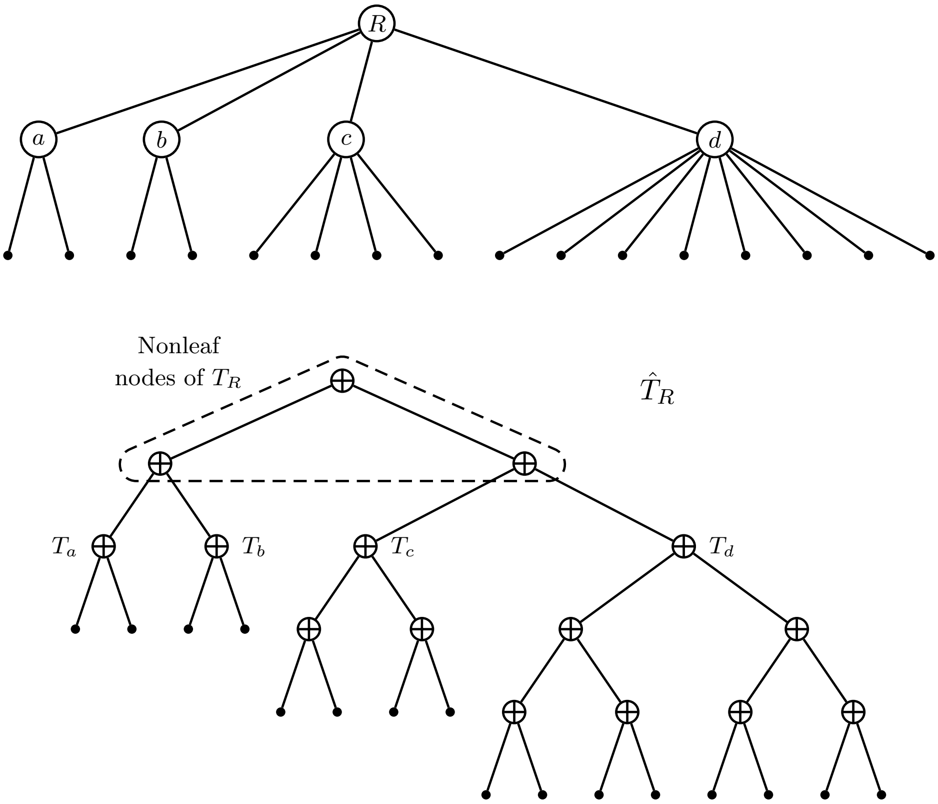

FIGURE 4.2. A hierarchy is shown at the top. If the aggregation tree of each office is balanced, then the cumulative aggregation tree of the root is isomorphic to the binary tree at the bottom. With this unbalanced hierarchy, it is impossible to adjust the aggregation trees of the offices to make the cumulative aggregation tree balanced.

Consider the implications for the subhierarchy that maximizes v_ℓ. Figure 4.1 shows a balanced hierarchy with five offices and sixteen shops, in which each office has a span of 4. It also shows $\hat{T}_R$, assuming that T_j is a balanced binary tree for all j. Observe that $\hat{T}_R$ is balanced. Consider, in contrast, the example in Figure 4.2; this hierarchy differs in that it is not balanced. As a consequence, even though aggregation by each office is balanced, the cumulative aggregation tree $\hat{T}_R$ is not balanced. Moreover, there is no way to change the aggregation trees of the individual offices so that $\hat{T}_R$ becomes balanced. These two examples illustrate that the hierarchy constrains the ways in which data are collectively aggregated by offices. A sufficient condition for $\hat{T}_j$ to be balanced is that both the hierarchy and the aggregation be balanced.

4.4 Balanced hierarchies

Next we show that optimal hierarchies are balanced if $b > 1/2$, under the ad hoc restrictions that there be no skip-level reporting and that the aggregation trees be balanced. Let the CF hierarchies that satisfy these restrictions be called CF* hierarchies. Our proof is by induction on the number of tiers, and it uses a continuous approximation in the inductive step.

We henceforth assume $b > 1/2$. This assumption is not merely to justify the restriction that aggregation be balanced. Even with that restriction, there are several places in the proofs where the results would change if $b < 1/2$. The only clear idea we have of the structure of optimal CF hierarchies when $b < 1/2$ is that they are not balanced.

We make use of the following notation. Define $a \equiv b^\alpha \in (0, 1)$. For $s > 0$, let $g(s) \equiv b^{\log_2 s} = s^{\log_2 b}$. Since $\log_2 b < 0$, it follows that g is strictly convex. Furthermore, the function $s \mapsto sg(s) = s^{\log_2 b+1}$ is concave because $b > 1/2$ and hence $\log_2 b + 1$ is between 0 and 1.

We first consider 3-tier hierarchies ($H = 2$) and then generalize the argument to arbitrary H. Consider, then, the CF hierarchies with $H = 2$ when there is no skip-level reporting. (We delay the restriction that the aggregation trees be balanced so that the reader can see where it is needed.) We fix n and the number s_R of offices in tier 1. How should the shops be distributed among the tier-1 offices? In particular, should they be distributed unevenly (as in Figure 4.1) or evenly (as in Figure 4.2)?

The distribution of shops affects the payoff but not the administrative cost. We want to show that, if we chose the $\{s_j\}_{j \in \Theta_R}$ that sum to n in order to maximize the payoff, then the spans should be equal to each other. Our strategy is to rearrange $U(\mathcal{H}) = v_R + \sum_{j \in \Theta_R} v_j$ so that it can be written $\sum_{j \in \Theta_R} F(s_j)$, where F is a strictly concave function. The result then follows.

According to Corollary 4.2 and Lemma 4.1, since $b > 1/2$ it is optimal for the aggregation of a tier-1 office to be balanced (in any CF hierarchy). As a continuous approximation, we let $L_{ji} \equiv \alpha + \log_2 s_j$ for $j \in \Theta_R$ and $i \in \theta_j$. Then, for $j \in \Theta_R$, we have $v_j = (s_j - 1) b^{\alpha + \log_2 s_j} = a(s_j - 1) g(s_j)$. The function $s \mapsto (s - 1)g(s)$ is strictly concave (since g is strictly convex and $s \mapsto sg(s)$ is strictly concave), so the total value of the tier-1 offices is maximized by distributing the shops evenly among them.

However, consider the value of the root:

$$
\begin{aligned}
v_R &= \sum_{j \in \Theta_R} \left(\frac{1}{s_j} - \frac{1}{n} \right) s_j b^{2\alpha + \log_2 s_j + \delta(j, T_R)} \\
&= a^2 \sum_{j \in \Theta_R} \left(1 - \frac{s_j}{n} \right) g(s_j) b^{\delta(j, T_R)}.
\end{aligned}
$$

The value v_R is maximized by distributing the shops unevenly, because then the root performs more direct allocation of resources. We need to show that this fact is outweighed by the concavity of $s \mapsto (s - 1)g(s)$. However, we can write $v_R + \sum_{j \in \Theta_R} v_j = \sum_{j \in \Theta_j} F(s_j)$ only if $\delta(j, T_R)$ is the same for each $j \in \Theta_j$—that is, only if T_R is balanced. Although we know from Section 4.2 that T_R should be balanced if the hierarchy is balanced, we can also see that T_R should *not* be balanced if the hierarchy is *not* balanced: T_R should give a lower depth (delay) for those subordinates for which $(1 - s_j/n)g(s_j)$ is higher (i.e., for which s_j is smaller). To proceed with our method of proof, we henceforth impose the ad hoc restriction that aggregation be balanced; that is, we restrict attention to 3-tier CF* hierarchies.

With this restriction and the continuous approximation that $\delta(j, T_R) = \log_2 s_R = g(s_R)$

for $j \in \Theta_R$, we have

$$v_R = a^2 g(s_R) \sum_{j \in \Theta_R} \left(1 - \frac{s_j}{n}\right) g(s_j). \tag{4.5}$$

Since $v_j = a(s_j - 1)g(s_j)$, we have $U(\mathcal{H}) = \sum_{j \in \Theta_R} F(s_j)$, where

$$F(s) \equiv \left(a(s-1) + a^2 g(s_R)\left(1 - \frac{s}{n}\right)\right) g(s) \tag{4.6}$$

$$= -a\left(1 - ag(s_R)\right) g(s) + a\left(1 - ag(s_R)/n\right) s g(s). \tag{4.7}$$

The first term in equation (4.7) is strictly concave because $-a(1 - ag(s_R)) < 0$ and g is strictly convex. The second term is strictly concave because $a(1 - ag(s_R)/n) > 0$ and $s \mapsto sg(s)$ is strictly concave. Hence F is strictly concave, as desired, and we have shown that optimal 3-tier CF* hierarchies are balanced.

We now give an inductive proof that optimal CF* hierarchies are balanced for arbitrary H. The inductive step has two components. First, as a corollary to Proposition 4.2, we show that, if optimal CF* hierarchies of height H are balanced, then subhierarchies of height H in an optimal CF* hierarchy of height greater than H are balanced.

Proposition 4.3 *Assume $b > 1/2$ and let $H \in \{1, 2, \ldots\}$. Suppose that optimal CF* hierarchies of height H are balanced. Consider an arbitrary CF hierarchy of height greater than H and suppose that a subhierarchy of height H is set to the CF* hierarchy that maximizes the profit (leaving fixed the number of shops in the subhierarchy, the height of the subhierarchy, and the other components of the CF hierarchy). Then the subhierarchy should be balanced.*

Proof. Let $\mathcal{H}$ be a CF hierarchy of height greater than H. Let $j \in J_H$ and consider the subhierarchy with root j. The total value and cost of the offices in the subhierarchy are the same as if the subhierarchy were an independent CF hierarchy. By assumption, we know that the CF* hierarchy of height H that maximizes the profit of this subhierarchy is balanced. The other effect that the subhierarchy has on the profit of the CF hierarchy is through the value of information processing of each office superior to j. According to Proposition 4.2 and Lemma 4.1, a balanced CF* hierarchy of height H also maximizes this value. □

The second component of the inductive step is as follows. Consider all CF* hierarchies of height $H+1$ in which the subhierarchies of height H are balanced. We show that, among this class, the optimal CF* hierarchies are balanced. Observe how the two components complete the proof. If optimal CF* hierarchies of height H are balanced then, by Proposition 4.3, the height-H subhierarchies of optimal CF* hierarchies of height $H+1$ are balanced. Therefore, from the second component, the optimal CF* hierarchies of height $H+1$ are balanced.

Consider a CF* hierarchy $\mathcal{H}$ of height $H+1$ such that each subhierarchy of height H is balanced. For $j \in \Theta_R$ and $h \in \{0, 1, \ldots, H\}$, let q_h^j be the number of nodes in tier h of subhierarchy j. (In particular, $q_0^j = n_j$.) Denote the vector $\langle q_0^j, q_1^j, \ldots, q_{H-1}^j \rangle$ by $\boldsymbol{q}^j$. We use the model of BCF hierarchies, with its continuous approximations, to measure the total value of information processing of the offices in subhierarchy j: $U_H(\boldsymbol{q}^j)$. Analogous to equation (4.5), we have

$$v_R \;=\; a^{H+1} g(s_R) \sum_{j \in \Theta_R} \left(1 - \frac{q_0^j}{n}\right) g(q_0^j).$$

Therefore, $U(\mathcal{H})$ is approximately equal to $\sum_{j \in \Theta_R} F_H(\boldsymbol{q}^j; s_R, n)$, where

$$F_H(\boldsymbol{q}^j; s_R, n) \;=\; U_H(\boldsymbol{q}^j) + a^{H+1} g(s_R) \left(1 - \frac{q_0^j}{n}\right) g(q_0^j).$$

Suppose $\mathcal{H}$ is not balanced because the BCF hierarchies under the root are not identical. Suppose then we "smooth" out the offices across the subhierarchies so that each has $\bar{q}_h \equiv (1/s_R) \sum_{j \in \Theta_R} q_h^j$ offices in tier h. The resulting balanced CF hierarchy has the same administrative cost as $\mathcal{H}$, but its payoff is $s_R F_H(\bar{\boldsymbol{q}}; s_R, n)$ rather than $\sum_{j \in \Theta_R} F_H(\boldsymbol{q}^j; s_R, n)$. The balanced CF hierarchy has a higher payoff if F_H is strictly concave in $\boldsymbol{q}$, which we show to hold if $b > 1/2$.

Proposition 4.4 *For $H \in \{1, 2, \ldots\}$ and $n \in \{2, 3, \ldots\}$ such that $n \geq 2^H$ and for $s_R \geq 2$, $F_H(\boldsymbol{q}; s_R, n)$ is strictly concave in $\boldsymbol{q}$ if $b > 1/2$.*

Proof. In Appendix C, we show analytically that $\partial F_H / \partial q_h^2 < 0$ for $h \in \{0, 1, \ldots, H-1\}$ and that

$$\frac{\partial^2 F_H}{\partial q_h^2} \frac{\partial^2 F_H}{\partial q_{h'}^2} \;>\; \left(\frac{\partial^2 F_H}{\partial q_h \partial q_{h'}}\right)^2$$

for $h, h' \in \{1, \ldots, H\}$ such that $h \neq h'$. These are necessary but not sufficient conditions for the Hessian matrix of F_H to be negative definite.

The strict concavity condition

$$F_H\left(\lambda \boldsymbol{q} + (1-\lambda)\boldsymbol{q}'; s_R, n\right) \;>\; \lambda F_H(\boldsymbol{q}; s_R, n) + (1-\lambda) F_H(\boldsymbol{q}'; s_R, n)$$

was tested and confirmed in 10^8 trials, with parameters and variables chosen randomly as follows (in each case, selection is with a uniform distribution on the indicated range):[2] (i) $\alpha \in (0, 10)$, (ii) $b \in (1/2, 1)$, (iii) $n \in \left\{16, \ldots, 10^8\right\}$, (iv) $H \in \{2, \ldots, \lfloor \log_2 n \rfloor - 1\}$, (v) $s_R \in \left\{2, \lfloor n/2^H \rfloor\right\}$, (vi) $\lambda \in (0, 1)$, (vii) $q_0 \in (2^H, n - (s_R - 1)(2^H))$, (viii) $q_0' \in (2^H, n - q_0 - (s_R - 2)2^H)$, and (ix) for $h \in \{1, \ldots, H-1\}$ given q_{h-1} and q_{h-1}', $q_h \in (2^{H-h}, q_{h-1}/2)$ and $q_h' \in (2^{H-2}, q_{h-1}'/2)$. These ranges reflect a bound on n of 10^8 as well as the restrictions that spans be at least 2 and that the total number of units among the s_R subhierarchies be n. □

[2] 10^8 trials is effectively exhaustive search, but this test can be framed probabilistically as follows: The probability that the strict concavity conditions fails on a set of parameters and variables whose measure (with respect to the distribution with which they are drawn) is greater than $1.6 \cdot 10^{-7}$ is less than $1.6 \cdot 10^7$.

5 Conclusions

In Section 3, we showed how the formulae for the profit of a CF hierarchy are simplified when the hierarchy is completely balanced. We then used these formulae as an approximation for balanced CF hierarchies.

Section 4 presented a series of results and arguments that are meant to show that optimal CF hierarchies are approximately balanced when $b > 1/2$. These results fall short of their target for several reasons.

Some of the shortcomings are due to the discreteness of CF hierarchies. To start with, we are trying to "prove" a result that is not even precisely defined: that optimal CF hierarchies are *approximately* balanced. Furthermore, at various points in which we establish some balancedness property of optimal CF hierarchies, we then use continuous approximations in subsequent steps of the proof.

A relatively minor point is that we rely partially on numerical simulations to test the concavity of a function. However, the simulation test is quite exhaustive and is complemented by analytic verification of certain necessary conditions.

More important is that we assume away skip-level reporting. Yet it is likely that there would be some skip-level reporting if optimal CF hierarchies were imbalanced. We note that the optimal hierarchies in Radner (1993) have skip-level reporting; so also may the optimal hierarchies in the symmetric model of Geanakoplos and Milgrom (1991), as explained in Van Zandt (1998). Nevertheless, the reasons for the optimality of skip-level reporting in those models do not arise here. Furthermore, one pays a price for skip-level reporting in our model because of the need to synchronize resource allocations.

The most significant gap is the following. In Section 4.2 we showed that, taking as given a completely balanced hierarchy, optimal aggregation is balanced. In Section 4.4 we showed that, if we restrict aggregation to be balanced, then optimal hierarchies are balanced. However, we do not jointly establish the optimality of balanced aggregation and balanced hierarchies.

Nevertheless, these results are not needed for the mathematical validity of the model of BCF hierarchies. Rather, their purpose is to allow one to judge how much effort and importance should be given to the characterization of optimal BCF hierarchies versus optimal CF hierarchies. Thus, they are useful even if incomplete.

Appendices: Proofs

A Proof on bounds on errors due to continuous approximations

Lemmas A.1 and A.2 are used in the proof of Proposition 3.2.

Lemma A.1 *Let* $s \geq 1$. *Then* $\lfloor \log_2 s \rfloor = \lfloor \log_2 \lfloor s \rfloor \rfloor$ *and* $\lceil \log_2 s \rceil = \lceil \log_2 \lceil s \rceil \rceil$.

PROOF. For $s \geq 1$, $\log_2 s$ is an integer only at integer values of s. Hence, $\log_2 s$ cannot leave an interval between two integers by rounding s up or down. □

Lemma A.2 *Suppose $\mathcal{H}$ is a balanced CF hierarchy. For $h \in \{1, \ldots, H\}$, $j \in J_h$, and $k \in \Theta_j$, we have*

$$-1 + \log_2 s_h \; < \; \delta(k, T_j) \; < \; 1 + \log_2 s_h. \tag{A.1}$$

Proof. Since T_j is balanced, $\delta(k, T_j) \geq \lfloor \log_2 s_j \rfloor$. Since $\langle I, J, R, \{\Theta_j\}_{j \in J} \rangle$ is balanced, $s_j \geq \lfloor s_h \rfloor$. Hence, $\delta(k, T_j) \geq \lfloor \log_2 \lfloor s_h \rfloor \rfloor$. According to Lemma A.1, $\lfloor \log_2 \lfloor s_h \rfloor \rfloor = \lfloor \log_2 s_h \rfloor$. Since $\lfloor \log_2 s_h \rfloor > -1 + \log_2 s_h$, we have $\delta(k, T_j) > -1 + \log_2 s_h$. A mirror argument shows that $\delta(k, T_j) < 1 + \log_2 s_h$. □

Proof of Proposition 3.2. Let $h \in \{1, \ldots, H\}$, $j \in J_h$, and $i \in \theta_j$. Since the hierarchy has no skip-level reporting, the path from j to i has one office in each tier $\eta \in \{1, \ldots, h\}$. By summing equation (A.1) along the path from j to i, we obtain

$$\sum_{\eta=1}^{h} -1 + \log_2 s_\eta \; < \sum_{\langle k_1, k_2 \rangle \in P_{ji}} \delta(k_2, T_{k_1}) \; < \; \sum_{\eta=1}^{h} 1 + \log_2 s_\eta.$$

Since $\sum_{\eta=1}^{h} \log_2 s_\eta = n/q_h$, the left-hand expression equals $L_h - \alpha h - h$ and the right-hand expression equals $L_h - \alpha h + h$. The center expression equals $L_{ji} - \alpha h$. Hence, we obtain equation (3.5) of the proposition.

We obtain the bound on v_j, equation (3.6), from the same calculation shown in equation (3.1) for the value of information in a completely balanced CF hierarchy, but substituting $L_h^- \equiv L_h - h$ or $L_h^+ \equiv L_h + h$ for L_h.

From equation (3.6) we have $v_j > b^h \sigma^2 (s_j - 1) b^{L_h}$. The total value of information of offices in tier h is therefore greater than $\sum_{j \in J_h} b^h \sigma^2 (s_j - 1) b^{L_h}$. The latter amount equals $b^h \sigma^2 (q_{h-1} - q_h) b^{L_h}$ because $\sum_{j \in J_h} s_j = q_{h-1}$ and $\sum_{j \in J_h} 1 = q_h$. By a mirror argument, we obtain that the total value of information of offices in tier h is less than $b^{-h} \sigma^2 (q_{h-1} - q_h) b^{L_h}$. By summing over h to obtain bounds on $U(\mathcal{H})$ and then rearranging, we derive equation (3.7).

□

B Some properties of binary trees

For any binary tree T, define the depth $\delta(T_1, T)$ of a subtree T_1 to be the depth in T of the root of T_1. Observe that, if T_1 is a subtree of a binary tree T, then for a node k in T_1 we have $\delta(k, T) = \delta(T_1, T) + \delta(k, T_1)$.

Lemma B.1 *Let T be a binary tree, and let $\{T_1, \ldots, T_M\}$ be a collection of subtrees of T such that $\{T_1^\circ, \ldots, T_M^\circ\}$ is a partition of T°. Then, for $b \in (0, 1)$,*

$$V(T, b) \; = \; \sum_{j=1}^{M} b^{\delta(T_m, T)} V(T_m, b).$$

PROOF.

$$V(T,b) = \sum_{m=1}^{M} \sum_{k \in T_m^\circ} b^{\delta(T_m,T)+\delta(k,T_m)} = \sum_{m=1}^{M} b^{\delta(T_m,T)} \sum_{k \in T_m^\circ} b^{\delta(k,T_m)} = \sum_{m=1}^{M} b^{\delta(T_m,T)} V(T_m,b).$$

□

Definition B.1 Let T be a binary tree that has disjoint subtrees T_1 and T_2 with roots r_1 and r_2, respectively. Let p_1 and p_2 be the parents of r_1 and r_2, respectively. The tree T' *obtained from T by switching T_1 and T_2* is the tree whose nodes and edges are the same as those of T except that edges $\langle p_1, r_1\rangle$ and $\langle p_2, r_2\rangle$ in T are replaced by $\langle p_2, r_1\rangle$ and $\langle p_1, r_2\rangle$ in T'.

Lemma B.2 *Let T be a binary tree that has disjoint subtrees T_1 and T_2, and let T' be the tree obtained from T by switching T_1 and T_2. Then*

$$V(T',b) - V(T,b) = (b^{\delta(T_2,T)} - b^{\delta(T_1,T)})(V(T_1,b) - V(T_2,b)). \tag{B.1}$$

PROOF. For $k \in T^\circ \setminus \{T_1^\circ \cup T_2^\circ\}$, we have $\delta(k,T) = \delta(k,T')$. Furthermore, $\delta(T_1,T') = \delta(T_2,T)$ and $\delta(T_2,T') = \delta(T_1,T)$. Treating each leaf in $T^\circ \setminus \{T_1^\circ \cup T_2^\circ\}$ as a subtree, it follows from Lemma B.1 that

$$V(T,b) = b^{\delta(T_1,T)}V(T_1,b) + b^{\delta(T_2,T)}V(T_2,b) + \sum_{k \in T^\circ \setminus \{T_1^\circ \cup T_2^\circ\}} b^{\delta(k,T)}$$

$$V(T',b) = b^{\delta(T_2,T)}V(T_1,b) + b^{\delta(T_1,T)}V(T_2,b) + \sum_{k \in T^\circ \setminus \{T_1^\circ \cup T_2^\circ\}} b^{\delta(k,T)}.$$

Subtracting $V(T,b)$ from $V(T',b)$ yields equation (B.1). □

Lemma B.3 $V(T,1/2) = 1$ *for all binary trees T. If $b < 1/2$, then $V(T,b) < 1$ for all binary trees T with at least two leaves.*

PROOF. If T is a binary tree with one leaf, then it contains just a single node that is length 0 from the root (itself). Hence, $V(T,b) = b^0 = 1$.

Inductive step: Let $s \in \{2,3,\ldots\}$. Assume that $V(T,b) \le 1$ if $b < 1/2$ (resp., $V(T,b) = 1$ if $b = 1/2$) for any binary tree with fewer than s leaves. Let T be a binary tree with s leaves. Since $s > 1$, the root of T has two children; let T_1 and T_2 be the subtrees under these children. By Lemma B.1, $V(T,b) = b(V(T_1,b) + V(T_2,b))$. Because T_1 and T_2 each have fewer than s leaves, it follows that if $b < 1/2$ then $V(T_1,b) \le 1$ and $V(T_2,b) \le 1$ and hence $V(T,b) < 1$ (resp., if $b = 1/2$ then $V(T_1,b) = V(T_2,b) = 1$ and hence $V(T,b) = 1$). □

Lemma B.4 *Let $s \in \{1,2,\ldots\}$ and $b \in (0,1)$.*

1. *Suppose $b > 1/2$. For $T \in \mathcal{T}_s$, $V(T,b) = V_s(b)$ if and only if T is balanced. Furthermore,*

$$V_s(b) = (2^{\lceil \log_2 s \rceil} - s)b^{\lfloor \log_2 s \rfloor} + (2s - 2^{\lceil \log_2 s \rceil})b^{\lceil \log_2 s \rceil}. \tag{B.2}$$

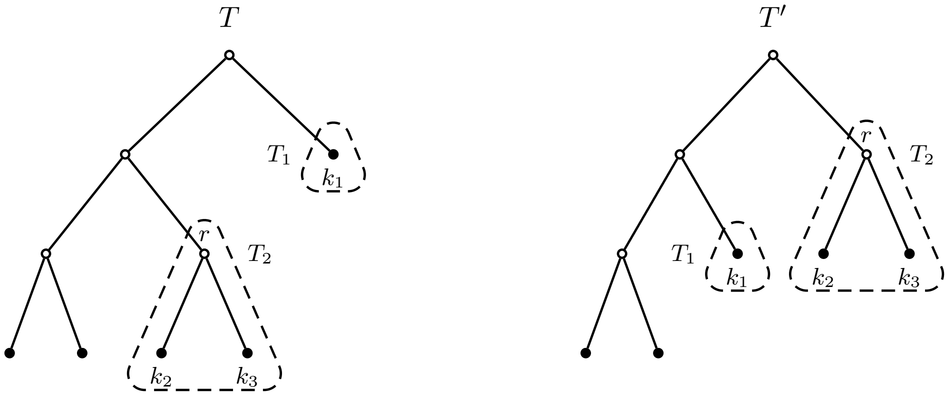

FIGURE B.1. The construction for Case 1 of the proof of Lemma B.4.

2. *Suppose $b < 1/2$. For $T \in \mathcal{T}_s$, $V(T,b) = V_s(b)$ if and only if T is serial. Furthermore,*

$$V_s(b) \;=\; \frac{b + b^{s-1} - 2b^s}{1-b}. \tag{B.3}$$

PROOF. Let $s \in \{1,2,\ldots\}$ and $b \in (0,1)$.

Case 1: Suppose that $b > 1/2$. Let $T \in \mathcal{T}_s$. We first show that $V(T,b) < V_s(b)$ if T is not balanced. Then we show that $V(T,b)$ is given by the r.h.s. of equation (B.2) if T is balanced.

Suppose that T is not balanced. Then there is a leaf k_1 such that $\delta(k_1,T) < \delta(T) - 1$. This implies that $s > 1$, and hence there are two leaves k_2 and k_3 such that $\delta(k_2,T) = \delta(k_3,T) = \delta(T)$ and such that k_2 and k_3 have the same parent r. Let T_1 be the subtree consisting just of k_1, and let T_2 be the subtree with root r and leaves $\{k_2,k_3\}$. Note that T_1 and T_2 are disjoint. Let T' be the tree obtained from T by switching T_1 and T_2. An example is shown in Figure B.1.

We have $V(T_1,b) = 1$ and $V(T_2,b) = 2b$. Thus, by Lemma B.2,

$$V(T',b) - V(T,b) \;=\; (b^{\delta(r,T)} - b^{\delta(k_1,T)})(1-2b).$$

Since $\delta(k_1,T) < \delta(T) - 1 = \delta(r,T)$ and $b \in (0,1)$, it follows that $b^{\delta(r,T)} < b^{\delta(k_1,T)}$. Since $b > 1/2$, we have $1 < 2b$. Hence $V(T',b) > V(T,b)$.

Suppose that T is balanced. Then every leaf has depth $\delta(T)$ or $\delta(T) - 1$. Let m be the number of leaves that have depth $\delta(T) - 1$, so that $s - m$ have depth $\delta(T)$. Adding two children below each of the m leaves whose depths are $\delta(T)$ would increase the number of leaves by m, to $s + m$. Furthermore, all leaves would have depth $\delta(T)$ and the number of leaves would be $2^{\delta(T)}$. Therefore, $s+m = 2^{\delta(T)}$, or $m = 2^{\delta(T)} - s$. We have thus shown that $2^{\delta(T)} - s$ leaves have depth $\delta(T) - 1$ and $s - (2^{\delta(T)} - s) = 2s - 2^{\delta(T)}$ have depth $\delta(T)$, so

$$V(T,b) \;=\; (2^{\delta(T)} - s)b^{\delta(T)-1} + (2s - 2^{\delta(T)})b^{\delta(T)}. \tag{B.4}$$

Recall that $\delta(T) = \lceil \log_2 s \rceil$ and $\delta(T) - 1 = \lfloor \log_2 s \rfloor$, except when s is a power of 2, and hence $2^{\delta(T)} - s = 0$. Therefore, the r.h.s. of (B.4) is equal to the r.h.s. of (B.2).

Case 2: Suppose that $b < 1/2$. Let $T \in \mathcal{T}_s$. We first show that $V(T,b) < V_s(b)$ if T is not serial. Then we show that $V(T,b)$ is given by the r.h.s. of equation (B.3) if T is serial.

Suppose that T is not serial. Then there is a node p with two children, r_1 and r_2, that are also nonleaf nodes. Let T_1 be a subtree consisting only of a leaf that is inferior to r_1, and let T_2 be the subtree whose root is r_2. Let T' be the tree obtained from T by switching T_1 and T_2. (This construction is also illustrated by Figure B.1, in reverse, with the following mapping from this construction to the notation in Figure B.1. The original nonserial tree T is T' in the figure; the node p with two nonleaf children is the root; r_1 is the left child and r_2 is the right child of the root; T_1 and T_2 are as marked in the figure; and the new tree after the switch is T.)

Then

$$V(T',b) - V(T,b) \;=\; (b^{\delta(T_2,T)} - b^{\delta(T_1,T)})(V(T_1,b) - V(T_2,b)).$$

Since (i) $\delta(T_2,T) < \delta(T_1,T)$, (ii) $b \in (0,1)$, (iii) $V(T_2,b) < 1$ (by Lemma B.3), and (iv) $V(T_1,b) = 1$ (since T_1 contains a single node), we conclude that $V(T',b) > V(T,b)$.

Suppose that T is serial. Then $\delta(T) = s-1$ and only two leaves have depth $\delta(T)$. Thus,

$$V(T,b) \;=\; b^{s-1} + \sum_{d=1}^{s-1} b^d \;=\; \frac{b + b^{s-1} - 2b^s}{1-b}.$$

□

C Proofs of necessary conditions for concavity in Proposition 4.4

We check certain necessary conditions for concavity of F_H. Recall that $g(s) \equiv b^{\log_2 s} = s^{\log_2 b}$ and $a \equiv b^\alpha$. Let $\hat{g}(s) \equiv b^{-\log_2 s} = s^{-\log_2 b}$ so that, in the formula for $U_H(\boldsymbol{q})$, $L_h = b^{\alpha h + \log_2(q_0/q_h)} = a^h g(q_0)\hat{g}(q_h)$. Then

$$U_H(\boldsymbol{q}) \;=\; g(q_0) \sum_{h=1}^{H} a^h \,(q_{h-1} - q_h)\, \hat{g}(q_h)$$

and

$$F_H(\boldsymbol{q}; s_R, n) \;=\; U_H(\boldsymbol{q}) + a^{H+1} g(s_R) \left(1 - \frac{q_0}{n}\right) g(q_0).$$

Consider first the second-order conditions that involve only $q_1, \ldots, q_{H-1}$. These depend only on

$$G_H(\boldsymbol{q}) \;\equiv\; \sum_{h=1}^{H} a^h \,(q_{h-1} - q_h)\, \hat{g}(q_h).$$

For $h \in \{1, \ldots, H-1\}$,

$$\begin{aligned}
\frac{\partial G}{\partial q_h} &= -a^h \hat{g}(q_h) + a^{h+1}\hat{g}(q_{h+1}) + a^h \,(q_{h-1} - q_h)\, \hat{g}'(q_h), \\
\frac{\partial^2 G_H}{\partial q_h^2} &= a^h \left(-2\hat{g}'(q_h) + (q_{h-1} - q_h)\, \hat{g}''(q_h)\right).
\end{aligned}$$

Since $b \in (1/2, 1)$, we have $-1 < \log_2 b < 0$. Let $\kappa \equiv \log_2 b + 1$, so that $0 < \kappa < 1$. Then

$$\text{(C.1)} \qquad \hat{g}'(s) \;=\; -(\log_2 b)s^{-\kappa} \;>\; 0$$

$$\text{(C.2)} \qquad \hat{g}''(s) \;=\; (\log_2 b)\kappa s^{-\kappa-1} \;<\; 0.$$

Since $q_{h-1} > q_h$, it follows that $\partial^2 G_H/\partial q_h^2 < 0$.

Note that, for $h, h' \in \{1, \ldots, H-1\}$ such that $|h - h'| \geq 2$, we have $\partial^2 G_H/\partial q_h \partial q_{h'} = 0$ and hence

$$\text{(C.3)} \qquad \frac{\partial^2 G_H}{\partial q_h^2}\frac{\partial^2 G_H}{\partial q_{h'}^2} \;>\; \left(\frac{\partial^2 G_H}{\partial q_h \partial q_{h'}}\right)^2.$$

Next we show that equation (C.3) holds for $h \geq 2$ and $h' = h - 1$. Let $h \geq 2$. Then

$$\frac{\partial^2 G_H}{\partial q_h \partial q_{h-1}} \;=\; a^h \hat{g}'(q_h).$$

Let

$$\begin{aligned}
\Gamma \;&\equiv\; \frac{1}{a^h a^{h-1}\hat{g}'(q_h)^2}\left(\frac{\partial^2 G_H}{\partial q_h^2}\frac{\partial^2 G_H}{\partial q_{h-1}^2} - \left(\frac{\partial^2 G_H}{\partial q_h \partial q_{h-1}}\right)^2\right) \\
&=\; \left(-2 + (q_{h-1} - q_h)\frac{\hat{g}''(q_h)}{\hat{g}'(q_h)}\right)\left(-2\frac{\hat{g}'(q_{h-1})}{\hat{g}'(q_h)} + (q_{h-2} - q_{h-1})\frac{\hat{g}''(q_{h-1})}{\hat{g}'(q_h)}\right) - a.
\end{aligned}$$

We show that Γ zFrom equations (C.1) and (C.2) we have

$$\begin{aligned}
\frac{\hat{g}''(q_h)}{\hat{g}'(q_h)} \;&=\; -\kappa q_h^{-1}, \\
\frac{\hat{g}'(q_{h-1})}{\hat{g}'(q_h)} \;&=\; \frac{q_{h-1}^{-\kappa}}{q_h^{-\kappa}} \;=\; s_h^{-\kappa}, \\
\frac{\hat{g}''(q_{h-1})}{\hat{g}'(q_h)} \;&=\; \frac{\kappa q_{h-1}^{-\kappa-1}}{-q_h^{-\kappa}} \;=\; -\kappa q_{h-1}^{-1} s_h^{-\kappa}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
s_h^{\kappa}\Gamma \;&=\; \left(2 + (q_{h-1} - q_h)\,\kappa q_h^{-1}\right)\left(2 + (q_{h-2} - q_{h-1})\,\kappa q_{h-1}^{-1}\right) - a s_h^{\kappa} \\
&=\; (2 + \kappa\,(s_h - 1))\,(2 + \kappa\,(s_{h-1} - 1)) - a\, s_h^{\kappa} \\
&=\; 4 + 2\kappa\,(s_h + s_{h-1} - 2) + \kappa^2\,(s_h - 1)\,(s_{h-1} - 1) - a\, s_h^{\kappa}
\end{aligned}$$

All terms but the last are positive. Since $a < 1$ and $s_{h-1} \geq 2$,

$$s_h^{\kappa}\Gamma \;\geq\; 4 + 2\kappa s_h - s_h^{\kappa} \;=:\; \gamma(\kappa).$$

We see that $\gamma(\kappa) > 0$ and hence $\Gamma > 0$ for all $\kappa \in (0, 1)$, as follows. Note that $\gamma(0) = 3 > 0$. Furthermore, $\gamma'(\kappa) = 2s_h - \kappa s_h^{\kappa-1} = (2 - \kappa/s_h)s_h > 0$, since $\kappa < 1$ and $s_h > 1$.

Finally, we show that F_H is concave in q_0. We can write

$$\text{(C.4)}\quad \begin{aligned} F_H(\boldsymbol{q}; s_R, n) &= a q_0 g(q_0)\hat{g}(q_1) - \left(\sum_{h=1}^{H-1} q_h \left(a^h g(q_0)\hat{g}(q_h) - a^{h+1} g(q_0)\hat{g}(q_{h+1}) \right) \right) \\ &\qquad - a^H g(q_0)\hat{g}(1) + a^{H+1} g(s_R) g(q_0) - a^{H+1} g(s_R)\frac{q_0}{n} g(q_0) \\ &= \underbrace{\left(a^H \left(a g(s_R) - 1 \right) - \sum_{h=1}^{H-1} a^h q_h \left(\hat{g}(q_h) - a\hat{g}(q_{h+1}) \right) \right)}_{\text{(a)}} g(q_0) \\ &\qquad + \underbrace{\left(a\hat{g}(q_1) - a^{H+1} g(s_R)/n \right)}_{\text{(b)}} q_0 g(q_0). \end{aligned}$$

Since $\hat{g}$ is an increasing function, $q_h > q_{h+1}$, and $a < 1$, we have $\hat{g}(q_h) - a\hat{g}(q_{h+1}) > 0$. Since $g(s_R) < 1$ and $a < 1$, we have $a g(s_R) - 1 < 0$. Therefore, term (a) in equation (C.4) is strictly negative and, since g is strictly convex, term (a) times $g(q_0)$ is strictly concave in q_0. Since $\hat{g}(q_1) > 1$, $a < 1$, and $g(s_R) < 1$, term (b) in equation (C.4) is strictly positive. Since $q_0 \mapsto q_0 g(g_0)$ is strictly concave, so is term (b) times $q_0 g(g_0)$. Hence, F_H is strictly concave in q_0.

References

Bolton, P. and Dewatripont, M. (1994). The firm as a communication network. *Quarterly Journal of Economics*, 109, 809–839.

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.

Radner, R. (1993). The organization of decentralized information processing. *Econometrica*, 62, 1109–1146.

Van Zandt, T. (1995). Continuous approximations in the study of hierarchies. *RAND Journal of Economics*, 26, 575–590.

Van Zandt, T. (1998). Organizations with an endogenous number of information processing agents. In M. Majumdar (Ed.), *Organizations with Incomplete Information*. Cambridge: Cambridge University Press.

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). Real-time hierarchical resource allocation with quadratic payoffs. INSEAD.

Van Zandt, T. (2003b). Structure and returns to scale of real-time hierarchical resource allocation. INSEAD.