

Organizations with an Endogenous Number of Information Processing Agents: Supplementary Notes

Timothy Van Zandt*
Princeton University

December 24, 1996

These notes supplement Van Zandt (1996) [henceforth, VZ]. They have not been subjected to proper editing and proofreading. Readers beware!

1 Associative computation by balanced hierarchies

This section studies the model of Keren and Levhari (1979) [henceforth, KL79]. This is a model of balanced hierarchies that perform an unspecified planning task. The two measures a hierarchy's performance are its delay and managerial costs. It is possible to interpret the model as one of associative computation, with a restriction to balanced hierarchies.

1.1 Introduction to KL79

Assume that the hierarchy is completely balanced.¹ Let H be the number of managerial tiers and let q_h be the number of managers in tier h , for $h = 1, \dots, H$. Processing starts when the data is distributed evenly to the q_1 managers in the lowest managerial tier. Each of these managers processes $s_1 = n/q_1$ items and sends the result to the next tier. The q_1 reports from tier 1 are distributed evenly among managers in the next tier, who then process $s_2 = q_1/q_2$ reports each and send the results to the next tier. Eventually, the single manager in tier 0 sends the answer to the output device.

The tiers are not busy concurrently, and so the total delay is the sum of the delay of each tier. The delay of tier h , with span s_h , is $s_h - 1 + \alpha$. The total delay is

$$d = H\alpha + \sum_{h=1}^H (s_h - 1) .$$

The total number q of managers is

$$q = \sum_{h=1}^H q_h .$$

Given $d_a + d_r = 1$ and $d_r + d_s = \alpha$, the formula (2) in VZ for the resource cost becomes

$$n - 1 + q\alpha .$$

Figure 1 shows a hierarchy for processing 64 items, and indicates the delay and the number of managers.

*This research supported in part by grants SES-9110973 and SBR-9223917 from the National Science Foundation. The research assistance of Archishman Chakraborty is appreciated.

¹The description of the processing and the formulas we derive will approximate those for incompletely balanced hierarchies. See Van Zandt (1995).

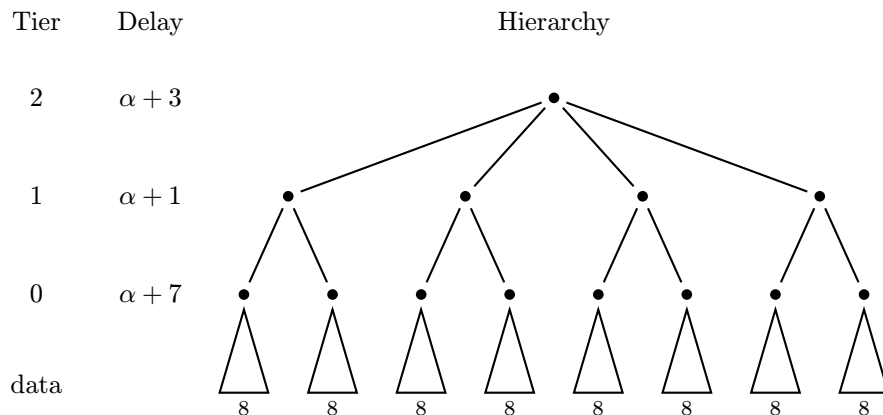


FIGURE 1. An example of a hierarchy in Keren and Levhari (1979). There are 64 reports, the total delay is $11 + 3\alpha$, the hierarchy has $8 + 4 + 1 = 13$ managers, and the total manager time is $63 + 13\alpha$. With a single manager, the delay and manager time would be $63 + \alpha$.

1.2 Some properties of efficient hierarchies in KL79

In this section, we prove a few properties of the efficient hierarchies in Keren and Levhari (1979), without using any approximations.

In what follows, a balanced hierarchy is *efficient* if there is no other balanced hierarchy with the same number of operatives that has either smaller delay and as few managers or fewer managers and no larger delay. That is, it is efficient in the KL79 model.

Proposition 1 *In efficient hierarchies:*

1. *The maximum span of control is weakly decreasing moving up the hierarchy.*
2. *For a hierarchy that has the lowest delay among those with n operatives, the maximum span of control is 3 for most tiers.*
3. *In the hierarchies that do not have the lowest delay, the maximum span of control is strictly increasing for most tiers.*

All the proofs are constructive: for hierarchies which do not have the indicated property, we construct a dominating hierarchy.

Here is an illustration of the first property. In the hierarchy in Figure 1, the span of the top tier is 4 and is greater than the span of next tier down, which is 2. If we fire a manager from tier 0 and distribute the manager's subordinates to the remaining managers on tier 1, we end up with the hierarchy in Figure 2, which also has a delay of $3\alpha + 11$ but which has one less manager. This property is illustrated again in Figure 3.

Whereas the first result is exact, the other two results contain the term *most*. What we prove for the second result is that, in a minimum-delay hierarchy, at most one tier can have a maximum span of 2, and fewer than

$$\min \{N \in \mathbb{N} \mid \lceil N \log_3 s \rceil 3 < Ns\}$$

consecutive tiers can have a maximum span of s , for $s > 3$. The limits are 6 tiers for $s = 4$, 1 for $s = 5$, 2 for $s = 6$, and 0 for all larger spans. Actually, a maximum span of 6 cannot

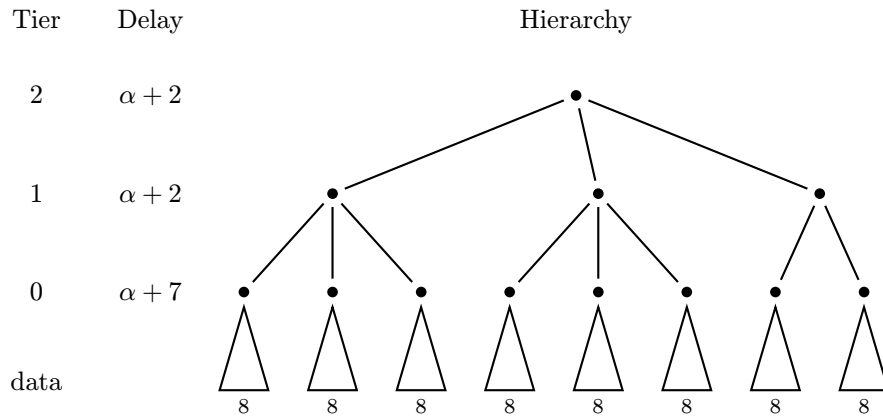


FIGURE 2. An improvement of the hierarchy in Figure 1, by making the span non-decreasing (the delay is the same but the total manager time is decreased by α).

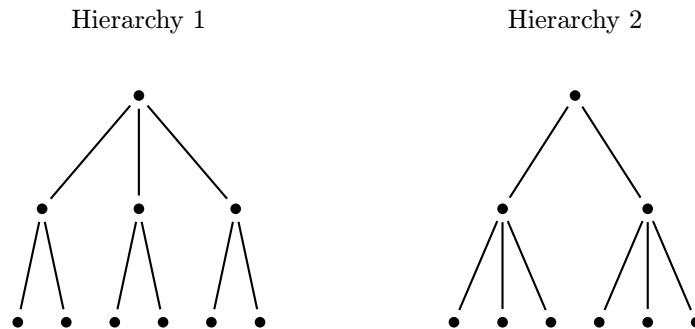


FIGURE 3. An illustration that the span in efficient hierarchies is weakly increasing. Hierarchy 1 has a span of 3 in the top tier and a span of 2 in the bottom managerial tier. It is dominated by hierarchy 2, which has spans of 2 and 3 in the top and bottom tiers, respectively. The total delay is the same (5) for both hierarchies, but hierarchy 2 has one less manager and hence α units less of manager time.

occur either because it can be divided into two tiers with maximum spans of 2 and 3. Hence, all but at most 9 tiers have a maximum span of 3, no matter how large the minimum-delay hierarchy is.

It is possible to construct examples of efficient, minimum-delay hierarchies in which some tiers have spans of 2, 4 or 5. For example, if $n = 5$, then the hierarchy with a single manager and a delay of 5 attains the minimum delay, and has fewer managers than other hierarchies with a delay of 5.

For the third result, we show that efficient hierarchies can have at most one tier of maximum span s for $s \geq 12$. It is also possible to derive bounds on the number of tiers that have maximum spans 4 to 11 on a case-by-case basis. From this and the first two results, we can conclude that in hierarchies with many tiers, either most tiers have a maximum span of 3, or for most tiers the maximum span is strictly increasing. Only low-delay hierarchies have a maximum span of 3 at most tiers.

PROOF OF EXACT: Part 1: Suppose $s_{h-1} < s_h$ for some $h \in \{2, \dots, H\}$. We can fire at least 1 (and up to q_h) managers from tier $h - 1$ and have $q_h(s_h - 1)$ managers remaining. The fired managers have a total of, at most, $q_h s_{h-1}$ direct subordinates. By assumption, $q_h s_{h-1} \leq q_h(s_h - 1)$. Thus, we can distribute these subordinates among the remaining managers on tier h without increasing any manager's span (and hence the delay of tier h) by more than 1. Because the maximum span of tier h falls by 1, to $s_h - 1$, the total delay does not change, but the new hierarchy has fewer managers. See, for example, Figure 1.

Part 2: Observe that in a minimum-delay, efficient hierarchy,

1. *No 2 consecutive tiers can have a maximum span of 2:*

If 2 tiers have a span of 2 (and thus total delay of 4), then we can eliminate the lower tier, and the maximum span (and delay) of the remaining tier is 4.

2. *Less than*

$$\min \{N \in \mathbb{N} \mid \lceil N \log_3 s \rceil 3 < Ns\}$$

consecutive tiers can have a maximum span of s , for $s > 3$:

Suppose there are N consecutive tiers with a maximum span of s and let q be the number of managers in the highest of these tiers. Then the lowest of the tiers can have up to qs^N direct subordinates. Suppose that we split the tiers into as many tiers N' with a span of 3 as are needed to have at least the qs^N direct subordinates. Then we need $3^{N'} \geq qs^N$, or $N' = \lceil N \log_3 s \rceil$. This reduces delay if the new delay, $N'3 = \lceil N \log_3 s \rceil 3$, is less than the old delay, Ns .

Part 3: we will only show that efficient hierarchies can have at most one tier of maximum span s for $s \geq 12$. It is also possible to derive bounds on the number of tiers with maximum spans of 4 to 11 on a case-by-case basis.

Suppose there are two tiers with a maximum span of s . We will try to make a better hierarchy by replacing these two tiers with three tiers that have maximum spans of 2, $\lceil s/2 \rceil - 1$, and slightly more than s , respectively.

Let q be the number of managers in the top tier. There are more than $q(s - 1)$ managers in the lower tier, and at most qs^2 direct subordinates of the lower tier. Replace the two tiers by three tiers with q , $2q$ and $q(s - 1) - 2q = q(s - 3)$ managers, respectively, so that

the total number of managers does not increase. The old delay was $2s$. The new delay is at most

$$2 + \left\lceil \frac{q(s-3)}{2q} \right\rceil + \left\lceil \frac{qs^2}{q(s-3)} \right\rceil .$$

The q 's cancel. This delay is less than $2s$ when $s \geq 12$.

□

1.3 Stationary balanced KL79 hierarchies

In this section, we examine how the results of Keren and Levhari (1979) change in the periodic case, when procedures must be stationary. A tier with span s can handle one problem every s cycles. The tier with maximum span is the processing bottleneck. If \bar{s} is the highest span of the tiers in a hierarchy, then the hierarchy's throughput is $1/\bar{s}$.

A managers whose span is s , with $s < \bar{s}$, will be idle $\bar{s} - s$ cycles for each problem. The resource cost per problem if managers are salaried is q/θ and the resource cost if managers are hourly is $n - 1 + q\alpha$. In either case, for fixed throughput, the only way to reduce the resource cost is to eliminate a manager. Although managerial costs and hence the optimal hierarchy may depend on whether managers are hourly and salaried, the set of efficient hierarchies is the same in both cases.

An efficient one-shot hierarchy in Keren and Levhari (1979) is also efficient in the periodic case, unless there happens to be another hierarchy with the same number of managers and delay but with higher throughput, which typically is not the case. However, because throughput is an additional criterion in the periodic case, there are also efficient periodic hierarchies that have higher throughput, but higher delay or resource costs, than the efficient one-shot hierarchies. These hierarchies are similar to the efficient one-shot hierarchies, but the spans of the lower tiers are flattened so as to increase throughput. They are likely to be optimal when throughput is important relative to delay. As an extreme example, the uniform hierarchies, which have the highest throughput for given managerial costs, are all efficient. (If a hierarchy's span is not constant, then throughput can be increased, without hiring more managers, by transferring managers from a low-span tier to a high-span tier.) Note that the throughput objective is enough to induce parallelization under the stationarity assumption (lower span increases parallelization and managerial costs, but also increases throughput), whereas in the one-shot model the delay objective was needed to induce parallelization.

2 The proof of Proposition 1

Proposition 1 *If delay d and throughput θ can be achieved by a stationary hierarchy with q managers, and if*

$$(3) \quad \left\lceil \frac{q+n-1}{q} \right\rceil \geq \lceil \log_2 q \rceil ,$$

then there is a stationary Rad93-type hierarchy with q managers and with delay of at most d and throughput of at least θ .

PROOF: By a *Rad93-type* hierarchy, we mean the MS networks in Van Zandt (1996a). In these networks, the postprocessing is the same as in an efficient one-shot Rad93 network, except that if there is some slack because $q+n \bmod q$ is not a power of 2, then some managers

start processing late (rather than finishing early, as in Rad93). We add the condition that if q is not a power of 2, then we design the network so that the root processor is one of the managers that starts postprocessing late. The root processor is thus busy postprocessing for $\lfloor \log_2 q \rfloor$ cycles, and this is the maximum postprocessing time of all managers.

Let \mathcal{N} be a network with q managers that has delay d and throughput θ , and maximum workload $\bar{w} = 1/\theta$. Let w_j be the workload of manager j , and index the managers in \mathcal{N} so that $w_1 \geq \dots \geq w_q$. Let c_j be the time between when manager j finishes and the root manager finishes in \mathcal{N} .

We will design a Rad93-type network \mathcal{N}' with q managers, so that its delay is not more than d and its throughput is at least θ . The postprocessing in \mathcal{N}' was described above. Number the managers in \mathcal{N}' in the reverse order in which they finish (e.g., the root is manager 1). Let c'_j and w'_j be defined for \mathcal{N}' as c_j and w_j were defined for \mathcal{N} . To complete the design of \mathcal{N}' , We need to assign data to the managers.

Let c_j (resp., c'_j) be the time between when manager j finishes and the root manager finishes in \mathcal{N} (resp., in \mathcal{N}'). Note that $d = \max\{c_j + w_j | j = 1, \dots, q\}$ and similarly for the delay in \mathcal{N}' . Note also that the Rad93-type networks are such that $c_j \geq c'_j$ for all j (see Van Zandt (1996a)). It follows that if we can distribute the data in \mathcal{N}' such that $w'_j = w_j$, then the delay for \mathcal{N}' is not greater than d , and of course $\bar{w}' = \bar{w}$, so that the two networks have the same throughput.

If we cannot distribute the data in this way, then instead start by assigning $\max\{0, w_j - y'_j\}$ items to each manager j in \mathcal{N}' , where y'_j is the number of postprocessing operations performed by j . From (3), $\bar{w} \geq y'_1 \geq y'_j$ for all j . For managers j such that $w_j < y'_j$, the workload in \mathcal{N}' is y'_j . Also, the Rad93 networks are such that j is not the first manager to start processing. Hence, manager j is not a constraint on throughput or delay. For managers j such that $w_j \geq y'_j$, the argument given in the previous paragraph applies. We may have assigned more than n items to the managers in \mathcal{N}' , but taking away the extra data simply improves the performance of \mathcal{N}' . \square

3 The returns to specialization in BD94

Here is a more extensive review of the returns to specialization in Bolton and Dewatripont (1994).

The authors are actually looking for stationary networks that minimize the managerial costs per problem. This is roughly the design problem of an organization with a large flow of problems where delay is not important and the organization can replicate stationary networks in order to attain the necessary throughput. If that were the end of the story, then the best stationary network to replicate is the one with a single manager, since parallelization just introduces communication costs. However, BD94 assume that the speed at which operations are executed in a stationary network increases with the network's throughput, because the throughput is the rate at which managers repeat the same operations. Specifically, the time each operation takes when the throughput is θ is $\tau(\theta)$ times the gross operation times given above; τ is strictly decreasing in θ . BD94 call these productivity gains "returns to specialization". Thus, while parallelization increases communication costs, it also increases returns to specialization, and may result in a net decrease in resource costs. A necessary condition for a network to minimize managerial costs per problem is that it be efficient with respect to gross managerial costs and throughput. Which of these efficient networks is optimal depends on τ .

For example, the gross operations for a network with a single manager is n . The throughput θ_1 is given by:

$$\theta_1 = \frac{1}{\tau(\theta_1)n} .$$

If $\tau(\theta) = \theta^{-\alpha}$, for $0 < \alpha < 1$, then $\theta_1 = n^{\frac{-1}{1-\alpha}}$, and managerial costs are $n^{\frac{1}{1-\alpha}}$. In a two-manager network, in which the root manager and the subordinate have equal workloads (which may or may not be optimal), the number n^* of items the subordinate processes, which is also the gross workload of each manager, is given by:

$$\begin{aligned} n^* &= n - n^* + \lambda + an^* \\ n^* &= \frac{n + \lambda}{2 - a} . \end{aligned}$$

The throughput θ_2 is then defined by:

$$\theta_2 = \frac{1}{\tau(\theta_2)n^*} .$$

The managerial costs are thus $\tau(\theta_2)(2n^*)$. When $\tau(\theta) = \theta^{-\alpha}$,

$$\theta_2 = n^{*\frac{1}{1-\alpha}} ,$$

and managerial costs are $2n^{*\frac{1}{1-\alpha}}$. This is greater than for the one-manager network if

$$\lambda < M((2 - a)2^{\alpha-1} - 1) .$$

This condition tends to hold when:

1. communication costs (a and λ) are low,
2. returns to specialization (α) are high, and
3. the problem size (n) is large.

4 Returns to scale of batch processing with exogenous cost of delay

4.1 Overview

A characterization of the returns to scale in information processing cannot, by itself, tell us what the impact of computational complexity is on the returns to scale of decision problems, such as those faced by organizations. One reason is that we must look at the decision problems in order to determine the cost of delay. Another, deeper, reason is that the size and type of computation is a choice variable in a decision problem, and the optimal size may not be proportional to the size of the decision problem. Hence the decision-theoretic approach taken in Section 3.3.2.

However, the curvature of the information processing production function will still play a role in the returns to scale of firms, and so its characterization is a useful warm-up exercise.

The economic inputs in information processing are the computational resources, such as managers and computers. The output is the computation of a given problem with a given delay and precision. If we were interested in the returns to scale of the information

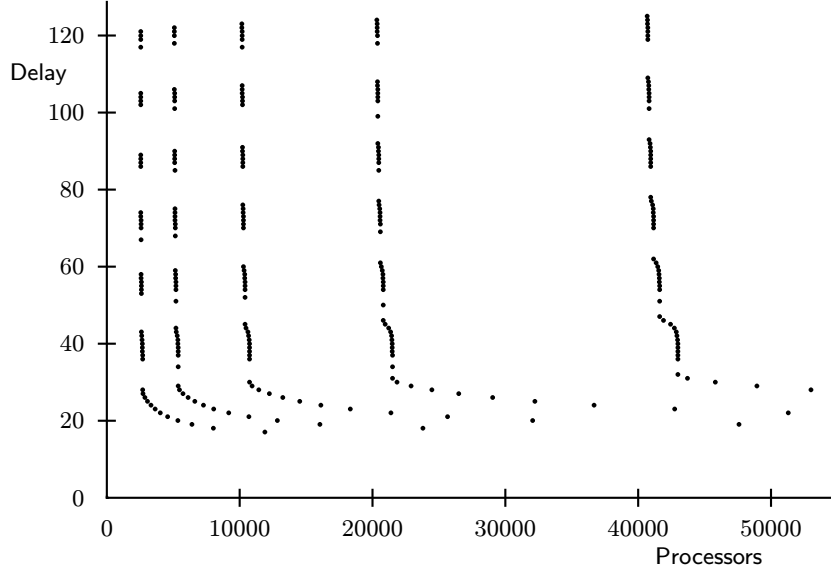


FIGURE 4. Sample isoquants of $n(r, d; \theta)$ for the periodic case. Problems arrive every 10 cycles. The isoquants are for problem sizes of 4, 8, 16, 32 and 64 times 10^4 .

processing service sector, then we would treat each computation as a different product, and measure scale by the number of times a computation is repeated (with varying data). However, we are studying the returns to scale of the end users of the computations—the organizations that compute decisions. Therefore, we fix the class of computation that is to be performed, and measure scale by the problem size. For example, if the problem is to sort or to sum a list of numbers, then the scale is the number of items in the list. For a fixed class of problems and fixed precision (and, in the case of periodic computation, for a fixed throughput θ), let $n(r, d)$ be the largest problem size that can be computed with delay d using computational resources r .

Suppose that the problem size n is proportional to the firm's output and that total costs are

$$cn + C(d) + wr ,$$

where c is the per-unit production cost when there are no computational constraints, $C(d)$ is the cost of delay (assumed to be independent of problem size), and w is the managerial wage. Then returns to scale depend on the slope of the per-unit administrative costs $(C(d)+wr)/n$. Sample isoquants of $n(r, d)$ for the Radner (1993) model of periodic associative computation with salaried managers are shown in Figure 4; observe that $n(\lambda r, \lambda d)$ is roughly convex in λ for fixed r and d . Hence, per-unit administrative costs are decreasing if $C(d)$ is linear. In fact, as long as $C(d)$ is polynomial, per-unit administrative costs converge to the lower bound $w\theta$ for any of the computation models studied in Section 2, because delay is logarithmic in problem size. (See Section 4 for details.)

However, since delay is not an economic input, but rather is costly because it degrades the decision-theoretic value of the computation, it is more likely that the cost of delay depends on the problem size. Suppose that n is proportional to the production inputs, and productivity is degraded by administrative delay, so that output is $f(d)n$, where $f(d)$ is a positive function that decreases to zero. Then per-unit production plus managerial costs for

input level n are

$$\frac{cn + wr}{f(d)n} .$$

Therefore, even if managers are not costly, unit costs grow without bound as long as delay grows without bound. This is a reasonable assumption about computation in human organizations for any class of problems.² For the models of associative computation discussed in this section, delay is bounded below by $\log_2 n$. The cost of delay can overcome even strongly increasing technological returns to scale. Keren and Levhari (1983, p. 481) show that the unit costs are unbounded for the Keren and Levhari (1979) model of computation (this would also be true for the other models) if $f(d) = e^{-d}$, output is $f(d)n^\alpha$, and $\alpha < 2$.

4.2 Some technical results

Section 2 of VZ describes various models of associative computation in which the computational costs are delay and managers. Here is an outline of the relationship between delay (d), managers (r) and problem size (n) for several of these models. This substantiates some claims made in the previous Section 4.1. In each case, we are dropping some floor and ceiling operators that are not relevant to Proposition 3.

1. The one-shot Keren and Levhari (1979) model: For uniform hierarchies with a span of $s > 2$ (which are not necessarily efficient),

$$\begin{aligned} d &= s \log_s n \\ r &= (n - 1)/(s - 1) . \end{aligned}$$

2. The one-shot Radner (1993) model: For an efficient hierarchy with r managers,

$$d = n/r + \log_2 r .$$

3. The periodic Radner (1993) model: For throughput x and an efficient network with q managers per cohort:³

$$\begin{aligned} d &\leq 1/x + n/q + \log_2 q \\ r &\leq nx + q . \end{aligned}$$

4. The periodic PRAM model: There is no substitution between delay and managers. For a throughput of x ,

$$\begin{aligned} d &= \log_2 n \\ r &= nx . \end{aligned}$$

Proposition 3 *If the total cost of delay and managers is*

$$C(d) + wr ,$$

where C is polynomial, then the ratio of minimized total costs to problem size converges to 0 in models 1 and 2, and converges to the lower bound wx in models 3 and 4.

²For a few problems, computer scientists have found constant-time, efficient algorithms, but these require unrealistic computation models. (*Constant-time* means that the computation time is bounded as the size of the problem varies, and *efficient* means that the number of processor cycles is linear in problem size.)

³To add slack for the sake of reducing idle time, delay is increased from the efficient one-shot level by at most $(1/x) - 1$ cycles. Per manager, there are at most $(1/x) - 1$ cycles of idle time plus one message; thus total work per cohort is at most $n + q/x$. See Van Zandt (1996a) for details.

PROOF: In the following, we use the fact that if C is polynomial, then $\lim_{n \rightarrow \infty} C(\log n)/n \rightarrow 0$.

In model 1, a uniform hierarchy with span $s = \log n$ has delay

$$d = \log n \log_{\log n} n = (\log n)^2 / \log^2 n .$$

Per unit costs of delay and managers are, respectively,

$$\begin{aligned} \frac{C((\log n)^2 / \log^2 n)}{n} &\rightarrow 0 \\ w \frac{n-1}{n} \frac{1}{\log n} &\rightarrow 0 . \end{aligned}$$

In model 2, with $r = n / \log n$ managers the per-unit costs of delay and managers are, respectively,

$$\begin{aligned} \frac{C(\log n + \log n / \log 2 - \log^2 n / \log 2)}{n} &\rightarrow 0 \\ \frac{w}{\log n} &\rightarrow 0 . \end{aligned}$$

In model 3, with $q = n / \log n$ managers per cohort, the cost of delay and managers are, respectively, at most

$$\begin{aligned} \frac{C(1/x + \log n + \log n / \log 2 - \log^2 n / \log 2)}{n} &\rightarrow 0 \\ w(x + 1 / \log n) &\rightarrow wx . \end{aligned}$$

In model 4, costs of delay and managers are, respectively,

$$\begin{aligned} \frac{C(\log n / \log 2)}{n} &\rightarrow 0 \\ wx & . \end{aligned}$$

□

In the rest of this section, we further characterize $f(r, d)$ for the one-shot and periodic Radner (1993) models.

Dropping some floor, ceiling and mod operators, which arise from the discrete nature of the problem and which are not significant asymptotically, in the one-shot case:

$$f(r, d) \approx rd - r \log_2 r$$

(Radner (1993)). This is asymptotically homogeneous of degree two.

The following is an implicit approximate solution for f in the periodic case:

$$f(r, d) \approx (rb + 1) \frac{d}{1+d} - \frac{\log_2(rb - f(r, d) + 1)}{1+d}$$

(Radner (1993) and Van Zandt (1996a)). Observe that as d and r increase proportionately,

$$d/(1+d) \uparrow 1 \quad \text{and} \quad \frac{\log_2(rb - f(r, d) + 1)}{1+d} \downarrow 0$$

(since $f(r, d)$ is increasing in r and d). The returns are thus increasing but asymptotically constant. The ratio $\lambda r / f(\lambda r, \lambda d)$ converges (as $\lambda \rightarrow \infty$) to xb (x processors per data item is the minimum that can keep up with the flow of data). The ratio $\lambda d / f(\lambda r, \lambda d)$ converges to dx/r .

5 An example of the aggregation of cost functions

First, the example without explicit derivations:

Suppose that there is one good, which we can think of as an input for this example, and shop i 's profit function is

$$\pi_i(x_i) = \alpha_i + \beta_i \log(x_i) .$$

The solution to

$$(2) \quad \begin{aligned} & \max \quad \sum_{i \in \theta_j} \alpha_i + \beta_i \log(x_i) \\ \text{subj. to:} & \quad \sum_{i \in \theta_j} x_i = x_j \end{aligned}$$

is

$$(3) \quad x_i^* = \frac{\beta_i}{\sum_{i' \in \theta_j} \beta_{i'}} x_j ,$$

and the maximized profit is

$$(4) \quad \pi_j(x_j) = \alpha_j + \beta_j \log(x_j) ,$$

where

$$\begin{aligned} \alpha_j &= \sum_{i \in \theta_j} \left(\alpha_i - \beta_i \log \left(\beta_i / \sum_{i' \in \theta_j} \beta_{i'} \right) \right) \\ \beta_j &= \left(\sum_{i \in \theta_j} \beta_i \right) . \end{aligned}$$

Hence, in the aggregation phase, each manager adds up the β 's of her immediate subordinates, and passes this sum to her immediate superior (the α 's are irrelevant). In the disaggregation phase, each manager j allocates $(\beta_k / (\sum_{k' \in \theta_j} \beta_{k'})) x_j$ to subordinate k . Manager j only needs to compute $(\sum_{k' \in \theta_j} \beta_{k'})$ once.

Now the derivation of formulae (3) and (4):

The first-order conditions for

$$\begin{aligned} & \max \quad \sum_{i \in \theta_j} \alpha_i + \beta_i \log(x_i) \\ \text{subj. to:} & \quad \sum_{i \in \theta_j} x_i = x_j \end{aligned}$$

are $\beta_i/x_i = \lambda$ for $i \in \theta_j$. Then

$$\sum_{i \in \theta_j} \beta_i = \lambda \sum_{i \in \theta_j} x_i = \lambda x_j .$$

This implies that $\lambda = \left(\sum_{i \in \theta_j} \beta_i \right) x_j$ and

$$x_i^* = \beta_i / \lambda = \frac{\beta_i}{\sum_{i' \in \theta_j} \beta_{i'}} x_j .$$

The value of the solution is

$$\begin{aligned} & \sum_{i \in \theta_j} (\alpha_i - \beta_i \log(x_i^*)) \\ &= \sum_{i \in \theta_j} \left(\alpha_i - \beta_i \log \left(\left(\beta_i / \sum_{i' \in \theta_j} \beta_{i'} \right) x_j \right) \right) \\ &= \sum_{i \in \theta_j} \left(\alpha_i - \beta_i \log \left(\beta_i / \sum_{i' \in \theta_j} \beta_{i'} \right) \right) + \left(\sum_{i \in \theta_j} \beta_i \right) \log(x_j) . \end{aligned}$$

6 Optimal Allocation Rules in GM91

The second half of Section 3.2.2 in VZ sketches the derivation of the optimal allocation rules in the Geanakoplos and Milgrom (1991) model, for a single resource. In this section, we give a more complete derivation, for multiple resources.

For multiple resources, the quadratic cost functions can be written

$$(5) \quad C_i(x_i, \gamma_i) = (\gamma_i - x_i)' B_i (\gamma_i - x_i) .$$

γ_i is a random vector. B_i is a known, symmetric and positive semi-definite matrix.

The optimal allocation by manager j to each shop i in division θ_j is:⁴

$$(6) \quad x_i^* = \hat{\gamma}_i^j - B_i^{-1} B_j (\hat{\gamma}_j^j - x_j) ,$$

where

$$(7) \quad \gamma_j = \sum_{i \in \theta_j} \gamma_i \quad B_j = \left(\sum_{i \in \theta_j} B_i^{-1} \right)^{-1} \quad \hat{\gamma}_i^j = E[\gamma_i | m_j] \quad \hat{\gamma}_j^j = E[\gamma_j | m_j] .$$

We will now show that

$$(8) \quad \tilde{C}_j(x_j, \{\gamma_i\}_{i \in \theta_j}, \{\hat{\gamma}_i^j\}_{i \in \theta_j}) = C_j(x_j, \gamma_j) + L_j(\{\gamma_i\}_{i \in \theta_j}, \{\hat{\gamma}_i^j\}_{i \in \theta_j}) ,$$

where

$$C_j(x_j, \gamma_j) = (\gamma_j - x_j)' B_j (\gamma_j - x_j)$$

and

$$L_j(\{\gamma_i\}_{i \in \theta_j}, \{\hat{\gamma}_i^j\}_{i \in \theta_j}) = \sum_{i \in \theta_j} (\gamma_i - \hat{\gamma}_i^j)' B_i (\gamma_i - \hat{\gamma}_i^j) - (\gamma_j - \hat{\gamma}_j^j)' B_j (\gamma_j - \hat{\gamma}_j^j) .$$

$C_j(\cdot)$ is j 's aggregate cost function, i.e., the full-information minimized cost for the division under j , as discussed in the previous section. $L_j(\cdot)$ is the loss due to prediction error.

Note that

$$(9) \quad B_i (\gamma_i - x_i^*) = B_j (\hat{\gamma}_j^j - x_j) + B_i (\gamma_i - \hat{\gamma}_i^j)$$

$$(10) \quad (\gamma_i - x_i^*)' = (\hat{\gamma}_j^j - x_j)' B_j B_i^{-1} + (\gamma_i - \hat{\gamma}_i^j)'$$

Then, premultiplying (9) by (10), shop i 's realized costs are

$$(11) \quad (\hat{\gamma}_j^j - x_j)' B_j B_i^{-1} B_j (\hat{\gamma}_j^j - x_j) + 2(\hat{\gamma}_j^j - x_j)' B_j (\gamma_i - \hat{\gamma}_i^j) + (\gamma_i - \hat{\gamma}_i^j)' B_i (\gamma_i - \hat{\gamma}_i^j) .$$

Noting that $\sum_{i \in \theta_j} B_i^{-1} = B_j^{-1}$, the sum of the costs over $i \in \theta_i$ is:⁵

$$(12) \quad (\hat{\gamma}_j^j - x_j)' B_j (\hat{\gamma}_j^j - x_j) + 2(\hat{\gamma}_j^j - x_j)' B_j (\gamma_j - \hat{\gamma}_j^j) + \sum_{i \in \theta_i} (\gamma_i - \hat{\gamma}_i^j)' B_i (\gamma_i - \hat{\gamma}_i^j) .$$

⁴The allocation rule (7) is part of Proposition 1 in Geanakoplos and Milgrom (1991) and is analogous to equation (A8) in Crémer (1980).

⁵If we take the expectation of (12) conditional on j 's information, the middle term disappears. If we also subtract $-\gamma_i' B_i \gamma_i$ from each shop's cost function, as in Crémer (1980) and Geanakoplos and Milgrom (1991), the last term becomes $\sum_{i \in \theta_i} \hat{\gamma}_i^j' B_i \hat{\gamma}_i^j$. This gives equation (9) in GM91 and equation (A12) in Cr80. Because j 's beliefs are not additively separated from j 's allocation, this expression is not useful for solving recursively for the optimal allocation rules of high-level managers.

Take the first two terms in (12) and subtract $(\gamma_j - x_j)B_j(\gamma_j - x_j)$:

$$(13) \quad (\hat{\gamma}_j^j - x_j)'B_j(\hat{\gamma}_j^j - x_j) + 2(\hat{\gamma}_j^j - x_j)'B_j(\gamma_j - \hat{\gamma}_j^j) - (x_j - \gamma_j)'B_j(x_j - \gamma_j)$$

Expand (13) and simplify to obtain

$$(14) \quad -(\gamma_j - \hat{\gamma}_j^j)'B_j(\gamma_j - \hat{\gamma}_j^j) .$$

Therefore, we can replace the first two terms in (12) by $(\gamma_j - x_j)B_j(\gamma_j - x_j)$ plus (14). This gives the total costs as in (8):

$$(15) \quad (\gamma_j - x_j)'B_j(\gamma_j - x_j) - (\gamma_j - \hat{\gamma}_j^j)'B_j(\gamma_j - \hat{\gamma}_j^j) + \sum_{i \in \theta_i} (\gamma_i - \hat{\gamma}_i^j)'B_i(\gamma_i - \hat{\gamma}_i^j) .$$

From this decomposition (8), we see that:

- Manager j chooses her information to minimize the expected loss $E[L_j(\cdot)]$. Hence, this choice does not depend on the allocation j receives.
- Only $C_j(x_j, \gamma_j)$ is relevant to the decision problem of j 's immediate superior, and hence this superior does not need to draw inferences about j 's choice of information or beliefs.

Note that for each manager j

$$\begin{aligned} \gamma_j &\equiv \sum_{i \in \Theta_j} \gamma_i = \sum_{i \in \theta_j} \gamma_i \\ B_k &\equiv \left(\sum_{k \in \Theta_j} B_k^{-1} \right)^{-1} = \left(\sum_{i \in \theta_j} B_i^{-1} \right)^{-1} . \end{aligned}$$

By induction, each manager j chooses an allocation to minimize the expected value of $\sum_{k \in \Theta_k} C_k(x_k, \gamma_k)$. The solution is

$$x_k = \hat{\gamma}_k^j - B_k^{-1}B_j(\hat{\gamma}_j^j - x_j) \quad \text{for } k \in \Theta_j .$$

The minimized value of $\sum_{k \in \Theta_k} C_k(x_k, \gamma_k)$ is analogous to (8), but in computing the minimized total cost of the division we need to carry along also the losses from the prediction errors of the subordinate managers. The total costs for the whole organization are thus

$$(\gamma_R - x_R)'B_R(\gamma_R - x_R) + \sum_{j \in J} L_j \left(\{\gamma_k \mid k \in \Theta_j\}, \{\hat{\gamma}_k^j \mid k \in \theta_j\} \right) .$$

7 Increasing returns to scale in GM91

To see that there is no optimal firm size in Geanakoplos and Milgrom (1991), we shall describe how to join two GM91 hierarchies in a way that decreases the expected total costs of the shops and does not change the managerial costs. Let A and B be the root managers of the two hierarchies, with total resources x_A and x_B , respectively. Merge the hierarchies by making manager B a subordinate of manager A . As a worse case, assume that manager B is the recipient of x_B , and hence manager A only makes additional net transfers to B that do not depend on x_B . (If x_B is random, then A treats x_B as part of B 's random cost parameters; if x_B is constant, then it would be equivalent to assume that A receives x_B directly.)

Manager A has the option of not making any transfers to B , in which case the total costs of the two divisions are the same as when they are independent. This does not qualify as informational integration, but typically it will be optimal for A to make transfers, which results in strictly lower expected total costs for the two divisions. Even if A has no information about the shops under B , A will make transfers to B contingent on A 's information: (i) the marginal expected cost of any transfer to B is deterministic; (ii) the marginal expected cost of transfers to A 's other subordinates depends on A 's information; (iii) A 's allocation equalizes the marginal expected costs of transfers across subordinates. If A does have information about B 's shops, the merger results in further cost reductions.

A simple example is possible with the following symmetry assumptions:

1. $B_i = 1$ for all i (which implies that $B_j = 1/|\theta_j|$).
2. $\{\gamma_i \mid i = 1, \dots, n\}$ are i.i.d., with mean 0 and variance σ^2 .
3. For any collection of shops, total resources are 0. (Coordination involves only transfers between shops.)

Suppose that there are two firms, A and B , each of size n and each managed by a single manager. Suppose each manager can observe the costs of his n shops, so that expected total costs for each firm is

$$(1/n) \text{Var}(\gamma_j) = \sigma^2 .$$

When manager B reports to A , and both managers acquire the same information as when they are independent, total costs are

$$(16) \quad (1/2n)(\gamma_A + \gamma_B)^2 - (1/2n)(\gamma_A + \gamma_B - \hat{\gamma}_A^A - \hat{\gamma}_B^A)^2 \\ + \sum_{i \in \theta_A} (\gamma_i - \hat{\gamma}_i^A)^2 + (1/n)(\gamma_B - \hat{\gamma}_B^A)^2 + (B\text{'s prediction errors}) .$$

Since A and B know their own shops costs, (16) equals

$$(17) \quad (1/2n)(\gamma_A + \gamma_B)^2 - (1/2n)(\gamma_A + \gamma_B - \hat{\gamma}_B^A)^2 + (1/n)(\gamma_B - \hat{\gamma}_B^A)^2 \\ = (1/2n)(\gamma_A + \gamma_B)^2 + (1/2n)(\gamma_B - \hat{\gamma}_B^A)^2 .$$

Since $E[(\gamma_A + \gamma_B)^2] = 2n\sigma^2$ and

$$E[(\gamma_B - \hat{\gamma}_B^A)^2] \leq E[\gamma_B^2] = n\sigma^2 ,$$

expected total costs are strictly less than $2\sigma^2$, which were the expected total costs of the two firms when independent.

References

- [BD94] 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.
- [GM91] 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. (1979). The optimum span of control in a pure hierarchy. *Management Science*, 11, 1162–1172.
- 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.
- [Rad93] 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.
- [VZ] Van Zandt, T. (1996). Organizations with an endogenous number of information processing agents. Princeton University.
- Van Zandt, T. (1996a). The scheduling and organization of periodic associative computation. Princeton University.