ANNALES D'ÉCONOMIE ET DE STATISTIQUE. – N° 25/26 — 1992

Information Processing in Firms and Returns to Scale

Roy RADNER, Timothy VAN ZANDT *

ABSTRACT. – What are the returns to scale in decision-making, when information processing is costly? In a parallel-processing model of a firm, we characterize efficient networks for associative operations, where efficiency is measured in terms of the number of individual processors (of given capacity) and the information-processing delay. We then embed this structure in a model of decision-making under uncertainty, and find that returns to scale can vary from increasing to sharply decreasing, depending on the intercorrelation of the data and on the loss function for incorrect decisions.

Le traitement de l'information dans les firmes et les rendements d'échelle

RÉSUMÉ. – Quels sont les rendements d'échelle associés à la prise de décisions quand le traitement de l'information est coûteux ? Dans un modèle de l'entreprise avec traitement en parallèle de l'information nous caractérisons les réseaux efficaces au sens du nombre de processeurs requis (de capacité données) et du temps de traitement. Nous incorporons cette structure dans un modèle de choix en environnement incertain, et trouvons que les rendements d'échelle peuvent se révéler croissants ou fortement décroissants selon la corrélation entre les données et la fonction de perte associée à une mauvaise décision.

* R. RADNER: AT&T Bell Laboratories, Murray Hill, New Jersey 07974; T. VAN ZANDT: Princeton University, Princeton, New Jersey 08544. The authors are grateful to A. GREENBERG, M. R. GAREY, P. B. LINHART, and J. E. MAZO for useful discussions. The views expressed here are those of the authors, and not necessarily those of AT&T Bell Laboratories. This research has been supported by National Science Foundation grant SES-9110972.

1 Introduction

The study of returns to scale in firms has traditionally focused on technological returns to scale in the production process. However, as the scale of a firm's production grows, so does its administrative apparatus. The process of managing a firm, although not as well understood nor as extensively studied as the production process itself, uses significant resources and is important to the profitability of the other operations in the firm. Therefore, this process may also have a significant impact on returns to scale. The purpose of this paper is to explore this impact.

We must first ask what managers do, how managing is related to the scale of the firm, and what resources it uses. The answer to the first question is complex, because managers, or more generally, the administrative staff, have many functions, including monitoring employees, setting goals, processing information, and making decisions (*See* RADNER [1989] for a discussion).

Current research on the internal theory of the firm emphasizes the role of incentives in managerial activities. There is already a small body of literature on how incentives, or more specifically, monitoring activities, affect returns to scale. *See*, for example, CALVO and WELLISZ [1978], MELAMUD *et al.* [1988], and MCAFFEE and MCMILLAN [1989]. The general conclusion of this literature is that the need to monitor employees leads to decreasing returns to scale, because adding new employees at the base of a hierarchy (*e.g.*, in the production process) ultimately increases the number of levels in the supervisory hierarchy or increases the number of subordinates each supervisor must monitor. This leads to a loss of control by the owner of the firm over the employees.

We focus on a different activity of managers that is unrelated to incentive problems: the processing of information involved in decision-making. This includes processing sales data, writing a letter, or keeping the books, and it is what most members of the administrative staff do most of the time (MINTZBERG [1989]). Thus, its impact on returns to scale should also be significant.

Once we model decision-making in a firm as an activity that uses significant resources, mostly personnel, we are abandoning the standard rationality paradigm that is the main-stay of economic theory. Perhaps the first economists to incorporate explicit models of information processing technology into a theory of economic decision-making were J. Marschak, T. A. Marschak, C. B. McGuire, and S. Reiter (the last with the mathematician K. Mount). [1] Bounds on the capacities of individual humans for

1. For specific citations, *see* the "Bibliographic Notes" at the end of the paper.

information processing are also a central feature of H. Simon's discussions of "bounded rationality". Statisticians, too, in their studies of sampling and sequential analysis, have had in mind that observation is costly; this was made explicit by A. Wald in his pioneering development of the theory of statistical decision (1950).

There is also a significant body of literature concerned with the optimal organization of hierarchies and with returns to scale, based on the bounded rationality of managers. *See*, for example, WILLIAMSON [1967], KEREN and LEVHARI [1979, 1983], CREMER [1980], and AOKI [1986]. WILLIAMSON [1967] contains references to earlier work as well. That literature emphasizes coordination problems as a limit to firm size. The general objectives of the present paper are similar to those of that literature, but our treatment is distinct in that we develop an explicit model of the computation and of the other information processing that is involved in managerial decision-making.

Consider the decision-making that is involved in the management of a firm; this is a stochastic control problem. At successive dates, the managers receive information about the environment and the internal state of the firm, and then decide what actions are to be taken. The mapping from observations to responses constitutes the manager's policy or decision rule. Decision-making in a firm involves both *choosing* and *implementing* a decision rule.

This distinction between the choice of a policy and its implementation is useful, even though it is difficult to make precise in practice. The choice of decision rules is generally more complex than the implementation of decision rules, but there are several reasons that the choice of decision rules may have less impact on returns to scale than the implementation of decision rules, at least when we are measuring the long-run average performance of a firm that operates in a stationary environment:

1. The choice or modification of a policy will be infrequent compared to the occasions of its implementation, so that the costs of such choices can be amortized over a relatively long period of time.

2. Information processing and decision-making take time, in addition to using resources. The delay in choosing a decision rule only affects the performance of the firm in the short run, until the decision rule is finally chosen and any backlog of information is processed. Delay in implementing a decision rule affects the performance of the firm over and over again, since each decision is conditioned on old information.

3. The basis for a policy is usually part of the accumulated knowledge of a society. A new firm can often adopt a decision rule that is used by other firms that operate or have operated in similar environments. Thus, in a stationary environment we would expect decision rules to be at least locally optimal in the long run, even if in each period the firm or firms that operate in similar environments make only small improvements to the decision rule.

Therefore, it is a reasonable approximation in stationary environments to assume that decision rules are chosen costlessly and optimally, while

their implementation involves resources and delay; this is at least an improvement over the usual assumption that neither the choice nor the implementation of a policy is costly. Of course, real environments are not truly stationary, and it would be useful to model both aspects of decision-making, but we adopt the first approach because of its simplicity.

To implement a policy, the observations about the environment upon which the action depends must be collected, and then the action must be computed. Implementing a policy also involves storing and communicating the raw observations and the intermediate and final results. We shall group these activities under a single heading: "information processing."

These information processing activities use costly resources. What resources are required depends on the policy that is implemented. One can say without much argument that conditioning a policy on more information requires more resources. It is also generally true that a policy that is conditioned on recently observed information is more costly to implement than a policy that is conditioned on older information; this is another way of saying that information processing takes time, and greater resources are needed in order to speed up the processing. It may simply be impossible to implement some policies, even though the required observations are available, because of limits on the speed with which computations can be performed.

There is not much more that we can say about the implementation of policies without being more specific about the decision problem and the technology for processing information. The specific decision problem we study in this paper is that of a firm that must periodically predict the total demand from a number of sources, given that it observes the past individual demands and given that the demands at the sources follow some known stationary stochastic process. This is a standard prediction problem from statistical decision theory. We look at loss functions and stochastic processes where the optimal decision rules are linear functions of the observed demands and the previous prediction.

The specific information processing model we use involves the parallel computation of associative operations. We consider only associative operations because the linear decision rules that arise naturally in the decision problem outlined above involve mainly addition. One of us has argued elsewhere (RADNER [1989, 1990]) that a number of common decision-making paradigms (*e.g.*, pattern matching, finding a maximum) also involve primarily associative operations.

We use a model of parallel processing because in a firm many people are involved in the processing for any one decision; the task is so large relative to the capability of a single individual that the required time for one person to complete it would be impractical. Thus, the required computations are performed "decentrally" (in the jargon of the economics of organization) or "in parallel" (in the jargon of computer science). Our model of computation in the firm is thus that of an idealized parallel computer, whose individual processing elements are the members of the managerial organization. Indeed, in the present paper we focus on these individual processors as the only costly resource, and ignore the costs of observation, communication, and memory. We do this as a first approximation, and

there is reason to think that, in the current state of affairs, the actual processing of information by human organizations is more costly than its communication and storage.

To derive an information-processing production function, we have first to decide what the inputs and outputs are. In economic terms, the inputs are the resources – in our case, the processors – and the output is a service: a computation is performed with a given delay on a given body of data. (Note the potential for confusion with the "inputs" and "outputs" of the computation itself!) The computational service is in turn used to implement the decision-making policy. The value of the service depends on how good the resulting decisions are, compared to how good they would be without the service (or with some alternative service). In particular, in the present analysis we shall focus on two features that influence the value: (1) the amount of data processed, and (2) the processing *delay*, *i.e.*, the amount of elapsed time between the arrival of the data and the "output" of the decision (the result of the computation). [2]

The model of information processing is presented in Section 2. We (1) characterize efficient network/algorithm combinations, (2) provide a lower bound for the number of processors needed to achieve a given delay, and (3) show that this bound is asymptotically sharp under certain conditions. These results imply, in particular, that if one holds fixed the frequency of arrival of successive cohorts, and increases the size, n, of each cohort, then (1) the number of processors must increase at least in proportion to n, and (2) the delay must increase at least as fast as $\log n$. Thus, even if processors are freely available, it is not possible to maintain a constant delay indefinitely as one increases the "size" n of the computation. The computer science literature suggests that associative operations are in some sense the most amenable to parallel processing; [3] to the extent that this is so, our results provide lower bounds on cost and delay for other types of computations.

In Section 3 we consider two examples of the statistical decision problem described above. In each example one wants to find, for each number N of sources, the optimal combination of the number and frequency of the data items used and the network/algorithm. Here "optimal" means minimizing the sum of the expected loss due to prediction error and the cost of the processors, taking due account of the relation derived in Section 2 between the amount of data actually used, the delay, and the number of processors. The two examples differ in the covariance function of the stationary process of data items and in the function that describes the loss associated with every prediction error.

In the first example, the loss function is quadratic, the demands from different sources are mutually independent, and the sequence of demands from any one source is a stationary Gaussian first-order autoregressive process. We find (eventual) sharply decreasing returns to scale in N, the number of sources; indeed, as N becomes large, the processed information

2. A more general model would also envisage computations of varying "accuracy".
3. *See*, *e.g.*, SCHWARTZ [1980].

becomes asymptotically "worthless," relative to what can be attained without conditioning on any past history of data items at all! In the limit, returns to scale are asymptotically constant, and equal to the returns when there is no information processing.

In the second example, the demand from a given source at a given date is the sum of two terms, a "common demand" and a "local disturbance." At any given date, the common demand is the same for all sources, and the sequence of common demands is a stationary Gaussian autoregressive process. The local disturbances are independent and identically distributed Gaussian variables, *i.e.*, they are independent across sources and dates. The common demand processes and the local disturbances are mutually independent. In addition, the loss due to an error in predicting the total demand is proportional to the error; the constants of proportionality may depend on the sign of the error. In this example there are increasing returns to scale with N, although in the limit returns to scale are asymptotically constant, but greater than the returns without information processing.

The example with decreasing returns to scale is of particular interest. In the study of the returns to scale of production processes, it is generally held that decreasing returns to scale only arise when some input is held fixed; otherwise, a firm could duplicate a production process and thereby achieve at least constant returns to scale. In this paper, there is no resource in fixed supply. Why cannot the same argument be made about replicating a firm? For a large firm to truly be like two small firms, the two subentities must be informationally disassociated, with information shared only indirectly through markets. Our interest in this paper has been the returns to scale of firms as unified organizations, not the returns to scale of holding companies.

One conclusion that emerges from this work is that the impact of information processing in organizations on returns to scale depends critically on the organization's stochastic environment and on the loss from incorrect decisions. This work should be viewed as preliminary, because we do not give an extensive classification of conditions under which returns to scale are increasing, decreasing or constant. That is left for future research.

2 A Model of Computation in Firms

The purpose of this section is to present a model of computation that will allow us to explicitly determine what policies are feasible and what computing resources are required to implement them. As mentioned in the introduction, we will only be considering linear decision rules whose implementation primarily involves periodically adding a fixed number of data items. Therefore, this is the type of computation modelled in this

section. Addition could be replaced in the model by any associative binary operation, such as pattern matching or finding a maximum.

2.1. Definitions and Building Blocks

Call the unit of time in this section a cycle; it is the time it takes a manager (processor) to perform one operation, as described below. Every b cycles, a list of n data items (called a cohort) is received by the managerial organization, and the job of this organization is to compute the sum.

The atomic processing element in the managerial organization is called a processor. Each processor has an infinite addressed buffer or "in-box" and a register. Clearing the register (setting it to zero) can be performed instantaneously. It takes a processor exactly one cycle to read a single number from its in-box and add the value to its register, thereby setting its register to the resulting sum. The time is the same whatever are the values of the data that are added, including when a datum is added to a cleared register.

Each processor can send the value of its register to the buffer of any other processor or to an output device via one-way communication channels, and data can be sent from the input device to the buffers of all the processors. We thus implicitly assume that, like memory, communication channels are not a scarce resource. We also assume that communication is instantaneous. This does not mean that communication has no implicit cost. A processor has to read a number into its register before communicating it to another processor. If a processor has a partial sum, the processor can add one more datum from its in-box to the partial sum in one cycle, but if that datum is in the buffer of another processor, then two cycles are needed to add it to the partial sum.

This depiction of communication delay is fairly realistic; it is easy to send a report to another manager, but it takes time for the manager to read the report. On the other hand, direct communication channels may be costly in firms, just as they are in parallel computers. For example having a direct communication channel with another member of the managerial staff may require proximity. Therefore, we will keep track of the communication channels used by our algorithms, even though in the formal model these channels are assumed to be costless.

Each addition, communication and clearing of a register is called an *operation*. A processor may add a number that was received from the input device (a raw datum) or from another processor (a partial sum). The addition is called a ***preprocessing*** operation in the former case, and in the latter is called a *postprocessing* operation. A processor is said to be *busy* during a cycle if it performs an addition; otherwise, it is *idle*. An operation together with a time when it is to be performed is called an *instruction*. Formally, a *network* is a set P of processors and a list I of instructions; we lump the processors and the algorithm together in the definition of a network in order to ease the exposition. The *size* of a network is the number of processors, and is denoted by p. The *architecture* of a network $\langle$ P, I $\rangle$ is the directed graph whose vertices are the elements

of P, and where an edge connects processor j to processor j' if and only if j sends the value of its register to j' at some time. That is, the edges are the communication channels that are actually used.

The only (scarce) economic inputs in the information processing are the processors. The computation services are parameterized by the size n of each cohort, the time b between cohorts, and the delay d in calculating the sum of each cohort, assuming that each cohort is processed with the same delay.

A network is *feasible* if each processor performs at most one addition operation in each cycle. A feasible network is said to be *functional*, given its workload, if, for each cohort the network is to process and for any values the data may have, a single output is produced that is equal to the sum of the data. Since our processors do not make errors and we are free to specify what computation the processors are performing, nothing would be gained by allowing for errors in the computations.

A feasible and functional network is said to be as good as another such network if the former uses as few processors and has as short a delay as the latter. It is said to be better if it also uses fewer processors or has a shorter delay. A feasible and functional network is efficient if there is no other feasible and functional network that is better. We use these qualifiers for performances as well; in particular, a performance is said to be efficient if it is the performance of an efficient network.

Because the same computation is repeated over and over again in our model, it is useful to be able to carefully describe how a single cohort of data is processed. The terminology we have defined so far can be applied with only minor modification to such a one-shot computation. Call a network that performs such a computation a *one-shot* network. When some confusion would otherwise arise, we will call a network that processes periodically arriving data a *renewal* network. The term "renewal" refers to the fact that in such a network, when a processor's task in the processing of one cohort is completed, the processor will join the processing of a new cohort. Denote the number of processors in a one-shot network by r; this symbol will also be used to denote the number of processors involved in the processing of a single cohort in a renewal network.

Call the processing of each cohort of data in a renewal network a *project*. The processors that are involved in a project form a *team*. Each project can be represented by a one-shot network, with the clock starting with the arrival of the cohort ("local time").

The processors in the one-shot representation of a project represent *tasks*, not specific processors in the renewal network. In fact, if in a renewal network, a processor in a team that is processing a cohort passes on a partial sum, clears her register, and then later continues processing the same cohort, we can treat the second batch of processing as a new task in the one-shot representation; there need not be a one-to-one correspondence between the teammates involved in a project in a renewal network and the tasks in the one-shot representation of that project. A task is said to finish in the cycle that it passes on a partial sum or the final output. It is said

to begin in the first cycle in which it performs an addition. The length of the task is the time from when the task begins to when it finishes.

A renewal network is fully described by a one-shot representation of the processing of each cohort, together with an assignment of the processors to the tasks in the representations. A renewal network is *stationary* if there is a single one-shot representation $\langle P, I \rangle$ of the processing of all the cohorts and if the assignment of processors is stationary. The last condition means that there is a function $\iota : P \to \mathbb{N}$ and a permutation $\alpha : P \to P$ such that a processor with task j in a project remains idle for $\iota(j)$ cycles after it finishes and then, in the next cycle, takes up task $\alpha(j)$ in the processing of the unique cohort in which $\alpha(j)$ begins that cycle, and so on. Such a renewal network is completely characterized by $\langle P, I \rangle$, ι and α, since all initial assignments of processors to tasks are equivalent, up to a renaming of the processors. The network is said to be a replication of $\langle P, I \rangle$. It is said to be *strongly stationary* if the assignment rule α is the identity, so that each processor has the same task for each cohort that she processes. This does not mean, however, that each processor has the same teammates in each project.

2.2. Parallelization: Speed-up and Overhead

A single processor can add up the n items in a cohort in exactly n cycles. If $n > b$, then a new cohort arrives before this processor finishes the current cohort, and so a single processor cannot possibly keep up with the flow of data. Although any feasible network must then contain more than one processor, this does not mean that the data are processed in parallel. Parallel processing means that more than one processor add up the items in a single cohort (*i.e.*, that each team has more than one processor).

Parallelization is said to increase when the number of processors on each team increases. It is easy to see how increasing parallelization lowers the delay per cohort. For example, suppose that $n = 100$ and initially each cohort is processed by a single processor, with a delay of 100. Now let the items in each cohort be added by two processors. Each processor can add 50 of the items, and then one processor adds on the partial sum obtained by the other processor. The total delay is 51 cycles, little more than half the delay when there is one processor per cohort.

Note, however, that the speed-up that can be obtained from parallelization is bounded. The smallest delay is obtained by dividing the data for each cohort among $\lfloor n/2 \rfloor$ processors. (For a real number z, $\lfloor z \rfloor$ denotes z rounded down, and $\lceil z \rceil$ denotes z rounded up.) Each processor adds two data items, and then the partial sums are aggregated by having half the processors pass their partial sums to the other half, and then repeating this each cycle with the remaining processors until a single processor ends up with the overall sum. The resulting delay is $1 + \lceil \log_2 n \rceil$. The fact that this minimum delay rises with n will be important in the next section.

Between this minimum delay, which requires $\lfloor n/2 \rfloor$ processors per cohort, and the maximum delay of n cycles, using 1 processor per cohort, it is possible to obtain a delay of

$$\lfloor n/2 \rfloor + \lceil \log_2 (r + n \bmod r) \rceil \equiv \hat{d}(r, n), \tag{1}$$

using r processors per cohort. (*See*, e.g., RADNER [1990] for a proof and references.)

How parallelization increases the number of processors in a network is less obvious. In a one-shot network, doubling the number of processors that add the items by definition doubles the number of processors in the network. In a renewal network, however, doubling the number of processors *per cohort* does not double the number of processors in the network. Because the delay falls when there are more processors on a team, each processor is ready to work on a new cohort sooner. In a renewal network, we are interested in the number w of processor-cycles (man-hours) devoted to each cohort. The number p of processors that are employed in the network satisfies the identity $pb = w$, and so $p = w/b$. The number of processors in a network rises with parallelization if parallelization causes w to rise.

One addition is performed for each data item, and so the number of processor-cycles per cohort is at least n. Recall the implicit communication cost discussed earlier. Each processor in a team, except for the one that produces the final sum, passes on a partial sum that is added by one of its teammates. Thus, there are $r - 1$ more addition operations that must be performed, where r is the number of tasks in a stationary network. This is part of the overhead of parallelization.

A cycle when a processor is idle also counts as a processor-cycle devoted to some cohort. Let i denote the total number of idle processor-cycles for each cohort. Then the number w of processor-cycles, including idle time, devoted to each cohort is

$$n + r - 1 + i. \tag{2}$$

With increased parallelization, it is more difficult to schedule processors to tasks without much idle time. Thus, i increases with r. This is the other part of the overhead of parallelization.

Therefore, for a stationary network with r tasks per cohort, we see from (2) and (1) that

$$p \geqslant \left\lceil \frac{n + r - 1}{b} \right\rceil \equiv \hat{p}(r, n, b), \tag{3}$$

$$d \geqslant \hat{d}(r, n). \tag{4}$$

As we let r vary from 1 to $\lfloor n/2 \rfloor$, we trace out a lower bound on the efficient frontier of performances for the workload (n, b), at least out of the set of performances for stationary networks. RADNER [1990] shows that even if we allow non-stationary networks and define the delay of a renewal network to be the long-run average delay for each cohort, these bounds continue to hold except without the rounding up and down.

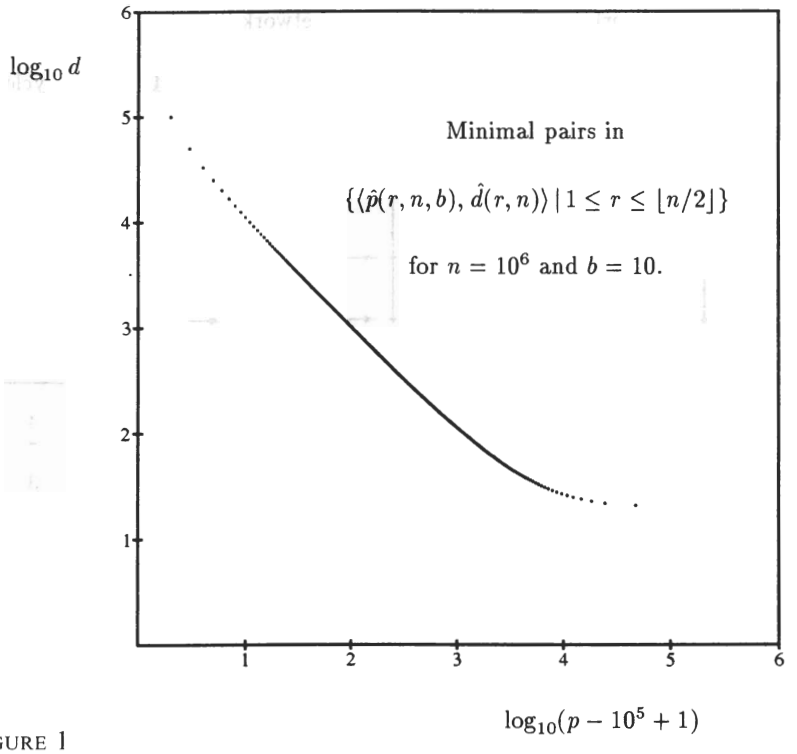

FIGURE 1

The lower bound on the performance efficiency frontier for a workload of 10^6 data arriving every 10 cycles.

Figure 1 depicts the lower bound defined by (3) and (4) for the case where $n = 10^6$ and $b = 10$. The plot is log-log, and $10^5 - 1$ is first subtracted from the number of processors because any functional network has at least 10^5 processors.

2.3. Efficient Networks

The lower bound takes into account the overhead from communication (what is plotted in Figure 3 is the overhead, plus 1), but not the idle time (other than the minimum number of cycles i per cohort that makes $(n+r-1+i)/b$ an integer). VAN ZANDT [1990] characterizes the true efficiency frontier when idle time is accounted for. A rough description of how to construct efficient networks follows; *see* VAN ZANDT [1990] for details.

Suppose that to achieve a delay $\hat{d}(r, n)$, we use a strongly stationary network which replicates an efficient one-shot network with r processors and delay $\hat{d}(r, n)$. It is not possible to achieve this delay with fewer processors per cohort, and so the number of processor-cycles per cohort, not including idle time, is the lowest for this delay. However, the idle time might be large.

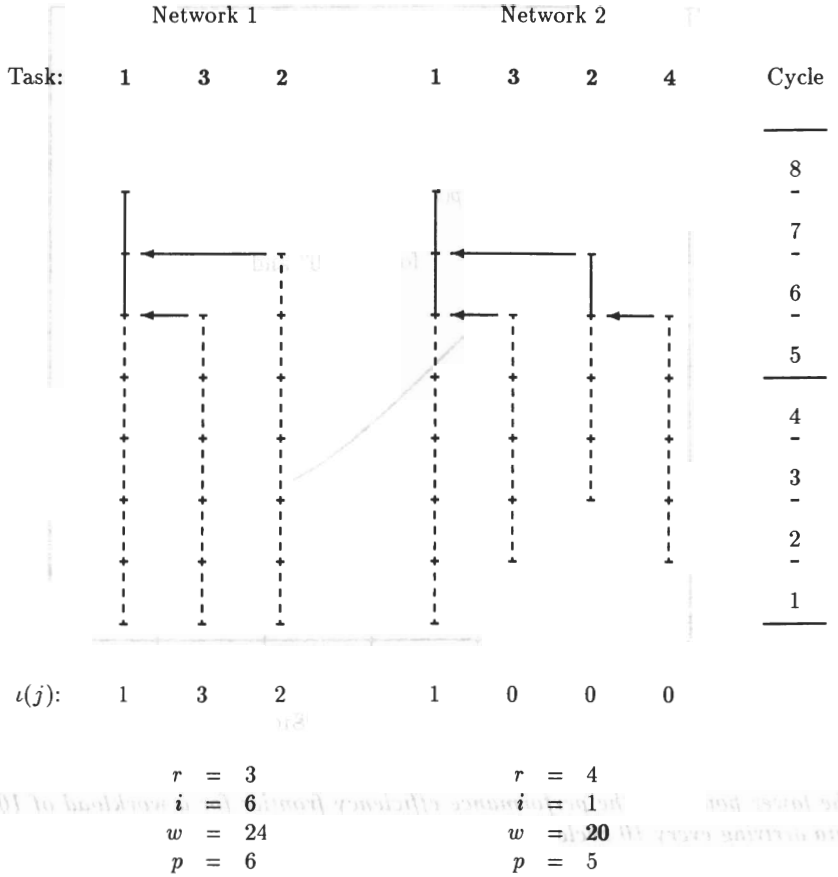

FIGURE 2

Two strongly stationary networks for a workload of 16 data arriving every 4 cycles. Network 1 has 3 tasks and 6 processors and Network 2 has 4 tasks and 5 processors. The activity of each processor is represented by a vertical line that is solid when the processor is postprocessing and that is broken when the processor is preprocessing. The arrows indicate the communication of partial sums.

Now add an extra processor per cohort. The overhead from communicating partial sums increases by 1 processor-cycle per cohort, but if we can change the scheduling of processors so that the number of idle cycles per cohort falls by at least 2, then the total number of processors in the network falls.

Arrange the $r+1$ processors in each project like an efficient one-shot network. The delay is lower than with r processors, but rather than finishing earlier, allow some processors to start later. The starting time of each task depends on how many raw data items it must process. Because of the slack in the project, there is some freedom in how the data items are distributed to the processors. If the total length of a task is a multiple of

b, then a processor with this task finishes processing one cohort just in time to begin the same task with the last cohort that has arrived, and is not idle at all.

Distribute the data items among the tasks so as to minimize the idle time per cohort. If the resulting idle time is lower than with r tasks, repeat the above exercise with $r+2$ tasks. Continue until adding a task does not reduce the idle time. As shown in VAN ZANDT [1990], the resulting strongly stationary network uses as few processors as any stationary network with the same delay, and any efficient performance can be achieved by such a network.

There is no closed-form description of the actual efficiency frontier. However, we can show, as stated in Proposition 1, that the lower bound given by equations (3) and (4) is often a good approximation of the actual efficiency frontier. Specifically, Proposition 1 states that $p(d, n, b)$ is close to

$$p_L(d, n, b) \equiv \min \{\hat{p}(r, n, b) \mid 1 \leqslant r \leqslant \lfloor n/2 \rfloor, \qquad d \leqslant \hat{d}(r, n)\},$$

which is the lower bound on the number of processors that can achieve a delay of d for the workload (n, b), as defined by (3) and (4). In fact, the minimum number of processors that achieve a given delay is of the same order as the lower bound (part 1 of Proposition 1), in one sense it is asymptotically close to the lower bound (part 3), and it is often within two of the lower bound (part 4). For the proof of Proposition 1, *see* VAN ZANDT [1990].

PROPOSITION 1: Let $\hat{r}(d, n)$ be the minimum number of tasks with which a delay of d is achievable when processing a single cohort of n data.

1. For all d and n such that $1 + \lceil \log_2 n \rceil \leqslant d \leqslant n$,

$$p(d, n, b) \leqslant \left(\frac{b+2-2/n}{3-2/n}\right) p_L(d, n, b) \leqslant b \cdot p_L(d, n, b).$$

2. If $n/\hat{r}(d, n) \geqslant b+1$, then

$$p(d, n, b) - p_L(d, n, b) < \frac{\hat{r}(d, n)}{n/\hat{r}(d, n) - b - 2} + 1.$$

3. If $n \to \infty$ and $\hat{r}(d, n)/n \to 0$, then

$$\frac{p(d, n, b) - p_L(d, n, b)}{p_L(d, n, b)} \to 0$$

at a rate $o((\hat{r}(d, n)/n)^2)$.

4. If $\hat{r}(d, n) < \sqrt{n}$ and $b \leqslant (\sqrt{n}/2) - 2$, then $p(d, n, b) \leqslant p_L(d, n, b) + 2$.

Figure 3 shows the actual efficiency frontier for $n = 10^6$ and $d = 10$. Compare with Figure 1. For each efficient performance $\langle p, d \rangle$, $(p - p_L(d, n, b))/p_L(d, n, b)$ is plotted in Figure 4. According to part 4 of Proposition 1, $p_L(d, n, b)$ should be within two of $p(d, n, b)$ when $\hat{r}(d, n) < \sqrt{n}$ (roughly, when $d > 10^3$); in this example, $p_L(d, n, b) = p(d, n, b)$

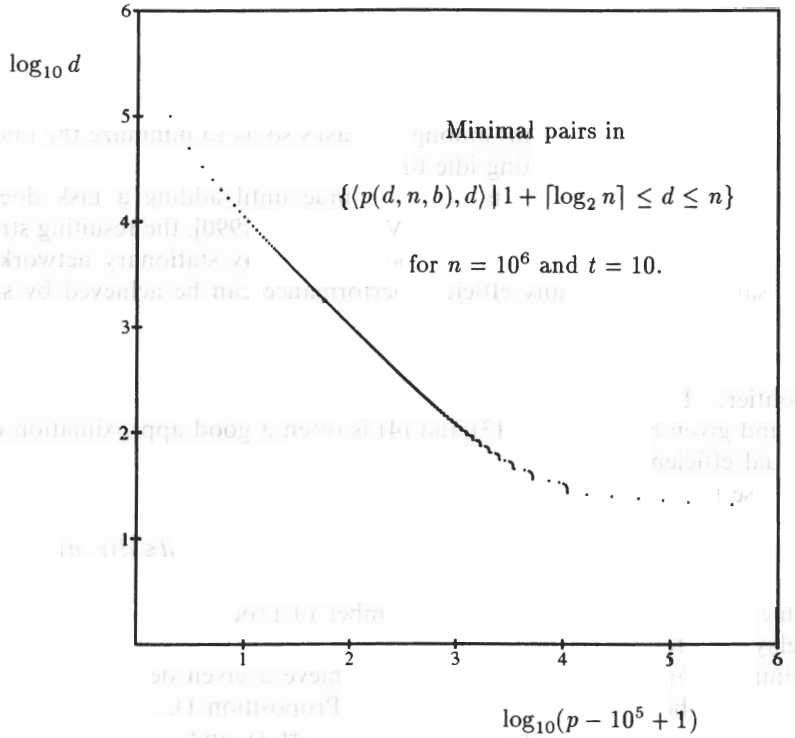

FIGURE 3

The efficient performance values for a workload of 10^6 data arriving every 10 cycles.

for these values of d. $p(d, n, b)$ is within 5% of $p_L(d, n, b)$, except for the six lowest delays. However, for the six lowest delays, $p_L(d, n, b)$ and $p(d, n, b)$ differ substantially. Part 1 of Proposition 1 states that $p(d, n, b)$ should be no more than 4 times $p_L(d, n, b)$. In fact, in this example, $p(d, n, b)$ is close to 3.5 times $p_L(d, n, b)$. More generally, as the number of data and the time between cohorts both become large, the number of processors in the efficient network that achieves the lowest delay d gets arbitrarily close to $b.p_L(d, n, b)$. Thus, the upper bound on $p(d, n, b)$ given in part 1 is tight.

The architectures of the efficient networks described above are not trees, even though they are replications of one-shot networks whose architectures are trees. Although the architectures have no cycles, each processor may communicate to more than one other processor. Also, the architectures are not rooted since each processor whose task passes on the final output is a maximal element in its architecture, and there is more than one such processor as long as the delay is much longer than the time between cycles.

RADNER [1990] describes some class of networks which he calls PPO networks and whose architectures are trees. These networks are not efficient, but they do fairly well at least for delays that are not too close to

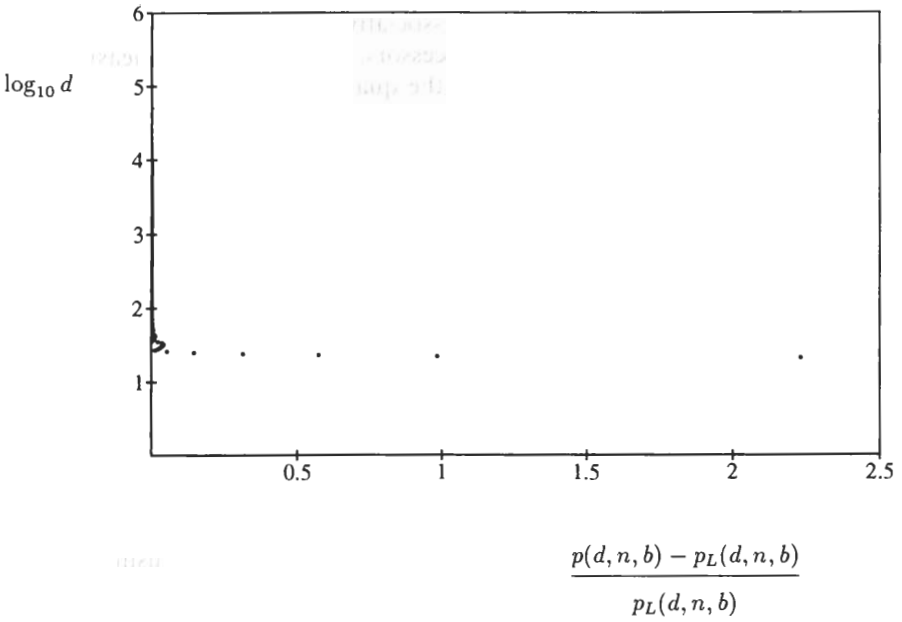

FIGURE 4

The relative difference between the actual size of networks and the lower bound p_L for a workload of 10^6 data arriving every 10 cycles.

the minimum. Since the architectures are trees, there is a single communication link per processor, and this economy of links is an advantage over the efficient networks described above.

3 Returns to Scale in Information and Information Processing

3.1. Meanings of "Returns to Scale"

It is not obvious what is a natural definition of returns to scale in information processing. Roughly speaking, we would like to know whether, if we multiply the "size" of the information processing workload by some factor, say k, we can obtain the result with more or less than k times the resources. From this perspective, the speed in obtaining the result is a *quality* of the output, since the greater the delay the less valuable will the output be, in general. In the information-processing described in

the previous section, performing an associative operation on n data items per cohort every b cycles with p processors, it is natural to measure the size of the workload by n and b, and the quantity of resources by p. The delay, d, is an (inverse) measure of quality. As we noted in Section 2, a lower bound on the efficiency frontier is given by

$$p \geq \frac{n+r-1}{b}, \tag{5}$$

$$d \geq \frac{n}{r} + \log_2 (r + n \bmod r), \tag{6}$$

where r varies from 1 to $n/2$.

From (6) we see that

$$d \geq 1 + \log_2 n, \tag{7}$$

with the lower bound being attained when $r = n/2$ (and r is an integer). No further decrease in the delay can be attained by using more processors. Thus we have the striking proposition: *It is not possible to maintain a constant quality* $(1/d)$ *while indefinitely increasing the size* (n) *of the information processing task, even if the quantity* (p) *of resources is unlimited.* In this strict sense, information processing exhibits strongly **decreasing returns to scale** in this particular model.

In this section we investigate whether this conclusion is modified if we embed the information processing task in a statistical decision problem. We do this in the context of an extended example suggested by the problem of predicting demand for a product in a large firm. Not surprisingly, we shall see that the nature of returns to scale depends on the statistical properties of the stochastic process of demands, and on the form of the function that describes the loss due to errors in prediction.

3.2. Predicting Demand

Suppose that a **firm wants to predict the total demand from a number** of sources. The data consist of the observations of past demands at all the sources. The optimal prediction is a function of these data, and this function depends on the probability law governing the stochastic process of demands, and on how far ahead one is predicting. On the other hand, the computation of this function requires that the original data items be processed, and this processing takes time, which in turn determines how far ahead one is predicting. The prediction error depends on the processing delay, and how much of the data is actually processed in calculating the prediction. As we saw in the previous section, the delay can be reduced and/or the amount of data processed can be increased by using more processors, but **only up to a point. In our example**, we shall specify a loss function and a stochastic process of demands such that the optimal prediction is a linear function of the data; in fact the processing will primarily require only **addition. Hence we can apply the results** of the previous section.

Suppose that there are N sources of demand at each date $t(-\infty < t < +\infty)$, let Q_{it} denote the demand from source i at date t, let

$$Q_t \equiv \sum_{i=1}^{N} Q_{it} \tag{8}$$

denote the total demand at date t, and let A_t denote the prediction of Q_t. Assume that the firm's profit at t is

$$U_t = \pi Q_t - \Psi(A_t, Q_t), \tag{9}$$

where π would be the profit per unit demanded if the prediction were perfectly accurate ($A_t = Q_t$), and $\Psi(A_t, Q_t)$ is the penalty or loss due to a prediction error. It is natural to assume that this loss is zero when $A_t = Q_t$, and increases as A_t departs from Q_t in either direction (not necessarily symmetrically). Notice that, if we measure "size" by total demand, then we have assumed that, *without any costs or delays due to information processing*, the firm would have constant returns to scale. The goal of our analysis is to see how the presence of those costs and delays changes how the profit depends on size.

Let H_τ denote the history of the process $(Q_{i\tau})$ up through date τ; this constitutes the "raw data" available to the firm at that date. In addition, there may be some previously processed data available, *i.e.*, the results of previous calculations; denote these by S_τ. At date τ the firm begins the process of calculating a prediction, the result of which appears at date $t = \tau + D$; thus it takes D units of time to process the data. We shall suppose that this result is A_t, the prediction of Q_t.

(More generally, one might consider the case in which the result is the prediction of total demand at some date subsequent to t, say $t + D'$; *e.g.*, D' might be the time required for production once it had been decided how much to produce. However, to simplify the exposition, we assume here that $D' = 0$.)

To summarize the above discussion, the prediction A_t is some function of H_{t-D} and S_{t-D}, where D is the delay due to information processing; call this function α_t. In fact, we should allow for the possibility that the delays may be different at different dates (*see* below); thus

$$A_t = \alpha_t(H_{t-D_t}, S_{t-D_t}). \tag{10}$$

A *policy* is a sequence of functions α_t. In the examples we analyze, the stochastic process (Q_{it}) is stationary. However, we shall not assume that the *policy* (α_t) is stationary in the sense of dynamic programming, *i.e.*, that the functions α_t are the same for all t. Indeed, we shall see that in some cases optimal policies may in a sense be cyclic. In any case, the information-processing workload will be periodic, as described in Section 2.

We shall, however, consider only policies for which the long-run average

$$\bar{u} \equiv \lim_{T \to \infty} \left(\frac{1}{T}\right) \sum_{t=0}^{T-1} EU_t \tag{11}$$

exists, where "E" denotes "expectation". Furthermore, the functions α_t will all be calculated using the same parallel processing network P (*see* Sec. 2), with p processors. Accordingly, the *net profit* is defined to be

$$\bar{v} \equiv \bar{u} - \varphi p, \tag{12}$$

where φ is the unit cost of processors. The delay, D_t, depends on the function α_t to be calculated, and on the network P. Thus we want to choose a policy (α_t) and a network P to maximize the net profit $\bar{v}$.

Some care is needed in the interpretation of the units of time. Suppose for the moment that time is a continuous variable, rather than discrete, and that the process (Q_{it}) is continuous in t (almost surely). If the total demand, Q_t, can be computed with a very small delay, then the prediction, A_t, will be very close to Q_t, and the gross profit, U_t, will be very close to its maximum, πQ_t. On the other hand, if the delay, D_t, is substantial, and Q_{it} and H_{t-D_t} are almost independently distributed, then the information will be almost worthless.

If – as in the present analysis – we take time to be discrete (integer-valued), then it is convenient in this section to take as the unit of time the interval between the arrivals of new data Q_{it}. In these units, let δ denote the time it takes to perform one "cycle" in the processing network. Thus $D_t = \delta d_t$, where d_t is the processing delay measured in processing cycles, as in Sec. 2. If D_t were sufficiently *small*, then for practical purposes it might be adequate to model the situation in discrete time as if D_t were zero, so that predictions would have zero error. However, in the present analysis we shall be interested in cases in which D_t is *large*, which leads us to assume that D_t is a positive integer.

In what follows, we shall be considering two alternative specific examples of the loss function Ψ:

• **Assumption QL (Quadratic Loss)**: The loss is proportional to the square of the difference between A and Q, *i.e.*,

$$\Psi(A, Q) = \psi(A - Q)^2, \tag{13}$$

where $\psi > 0$ is a fixed parameter.

• **Assumption LL (Linear Loss):** The loss is a piecewise linear function of $(A - Q)$, *i.e.*,

$$\Psi(A, Q) = \begin{cases} \psi_0(Q - A), & A \leqslant Q, \\ \psi_1(A - Q), & A \geqslant Q, \end{cases} \tag{14}$$

where ψ_0 and ψ_1 (>0) are fixed parameters.

Note that under Assumption QL the loss function is symmetric in the prediction error, whereas under Assumption LL it need not be.

We also make two alternative assumptions about the stochastic process of demands, Q_{it}. These are special cases of the following more general

structure:

$$(15)\qquad \begin{cases} Q_{it} = \mu + X_{it}, \\ X_{it} = Y_t + Z_{it}, \\ Y_t = \beta Y_{t-1} + V_t, \\ Z_{it} = \gamma Z_{i,t-1} + W_{it}, \end{cases}$$

where

1. t is integer-valued, $-\infty < t < +\infty$;
2. (V_t) and (W_{it}) are all Gaussian and independent, with mean 0;
3. $EV_t^2 = v^2$, $EW_{it}^2 = \omega^2$, $v \to \varepsilon$;
4. $\mu > 0$, $|\beta| < 1$, $|\gamma| < 1$;
5. (Y_t) and (Z_{it}) are stationary.

Note that X_{it} is the difference between Q_{it} and its mean, *i.e.*, is a "noise" variable. This noise itself is the sum of two components, a "common noise", Y_t, and an "idiosyncratic noise", Z_{it}, each of which is a linear first-order autoregressive process. The two special cases that we concentrate on are:

- **Assumption IN (Idiosyncratic Noise):**

$$(16)\qquad Y_t \equiv 0 \quad (\beta = v^2 = 0).$$

- **Assumption CN (Common Noise):**

$$(17)\qquad \gamma = 0.$$

Notice that the idiosyncratic noise is present in both cases, but under Assumption CN the idiosyncratic noise variables Z_{it} are independent and identically distributed.

3.3. Example 1

Our first specific example combines Assumptions QL (Quadratic Loss) and IN (Idiosyncratic Noise). We start with some general observations about this case. For a given date t, let $\tilde{H}_t$ denote the information on which the prediction A_t is actually conditioned. [4] Assumption QL implies that, for an optimal policy,

$$(18)\qquad \begin{aligned} A_t &= E(Q_t \mid \tilde{H}_t) \\ &= N\mu + \sum_{i=1}^{N} E(X_{it} \mid \tilde{H}_t). \end{aligned}$$

4. Formally, $\tilde{H}_t$ may be defined as the coarsest sigma-field in the underlying probability space with respect to which A_t is measurable.

From (3.11) and Assumption IN,

$$X_{it} = \gamma X_{i,t-1} + W_{it}, \tag{19}$$

where (W_{it}) are independent and identically distributed Gaussian variables with mean zero and variance ω^2. In addition, since (X_{it}) is stationary,

$$\xi^2 \equiv EX_{it}^2 = \frac{\omega^2}{1-\gamma^2}; \tag{20}$$

recall that $|\gamma| < 1$. Note that, for each i, the process (X_{it}) is Markovian, and that the N processes (X_{1t}), (X_{2t}), ..., (X_{Nt}) are independent and identically distributed.

Since we do not require that the policy (α_t) be stationary, we cannot assume that $\tilde{H}_t$ is simply a shift of $\tilde{H}_{t-1}$. However, suppose that, for some i, s, and t, $\tilde{H}_t$ determines the history of the process $(X_{i\tau})$ up through date $(t-s)$; then, as is well known,

$$\tilde{X}_{it} \equiv E(X_{it} \mid \tilde{H}_t) = \gamma^s X_{i,t-s}, \tag{21}$$

$$E[(\tilde{X}_{it} - X_{it})^2 \mid \tilde{H}_t] = (1-\gamma^{2s})\xi^2. \tag{22}$$

(We shall use these last results below.)

To get a feeling for the problem of determining an optimal policy, consider for the moment a (not necessarily optimal) policy in which at every date τ one uses the entire history H_τ to predict $Q_{\tau+D}$. In this case, $\tilde{H}_t = H_{t-D}$. If we let

$$X_t \equiv \sum_{i=1}^{N} X_{it}, \tag{23}$$

and take $s = D$, it follows from (18)-(22) that the optimal prediction of Q_t is

$$A_t = N\mu + \gamma^D X_{t-D}, \tag{24}$$

and the corresponding (gross) profit is

$$\bar{u} = N\pi\mu - N(1-\gamma^{2D})\xi^2. \tag{25}$$

Observe also that if the prediction were based on no observations at all (no information processing), then the optimal prediction would be $A_t = N\mu$, and the corresponding gross profit would be

$$\bar{u}_0 = N\pi\mu - N\xi^2 \tag{26}$$

(which could be negative). Hence the gross *gain* from the information processing, *i.e.*, the value of the processed information is

$$G \equiv \bar{u} - u_0 = N\gamma^{2D}\xi^2. \tag{27}$$

Recall from Section 2 that the speed-up from parallelization is bounded. Specifically, from (7) we have

$$D \geqslant \delta(1 + \log_2 N), \tag{28}$$

D is the lag in the information upon which the prediction is based. Since this lag goes to infinity with N, the net gain per source from processing the information goes to zero. *I. e.*, since $\gamma^2 < 1$, we see from (27) and (28) that

$$\lim_{N \to \infty} \frac{G}{N} = 0.$$

This implies that asymptotically $\bar{u}$ and $\bar{u}_0$ grow at the same rate:

$$\lim_{N \to \infty} \frac{\bar{u}}{N} = \frac{\bar{u}_0}{N} = \pi\mu - \xi^2. \tag{29}$$

$\bar{u}_0$, the gross profit when no information is processed, is also the net profit when no information is processed. The policy defined by (24) does have processing costs, and the lower bound on the number of processors needed to sum the N demands each period is $N\delta$. Thus, the processing cost per cohort is at least $\varphi\delta$, and it follows from (29) that asymptotically this policy yields a lower net profit than the policy that conditions on no information at all.

Let's consider a more general class of policies. As noted above, at each date t one has potentially available for processing any coarsening $\tilde{H}_t$ of (H_t, S_t). This includes, as an extreme case, the possibility that $\tilde{H}_t = \tilde{H}_{t-1}$, *i. e.*, no new information is processed. We shall not address the problem in such generality, but confine our analysis to the following class of policies: every B periods one selects M new data items, and processes these, together with what results were previously stored, to obtain a new prediction; in between these "new data" periods one updates the previous prediction without using new data items.

Again, it is convenient to look at the gross gain from information processing, $G = \bar{u} - \bar{u}_0$. From (22) we see that, for any choice of M and B, the gain is maximized by sampling those M sources for which the previously processed data are the oldest. Thus (supposing that N/M is an integer), we divide the N sources into $R \equiv (N/M)$ "regions", and every B periods we process new data from that region with the oldest previously processed data. Every region is thereby sampled every (BR) periods. From (21), we see that every time a region is processed we need to use only the current observation for that region. Because the sources are independent, the total gain G is the sum of the corresponding gains for the individual sources.

We now derive the optimal prediction formula for this class of policies, and show that it is essentially of size M. Since μ is known, it is convenient to change the notation slightly so that A_t denotes the prediction of X_t, not Q_t. With this new notation, the optimal prediction is

$$A_t = E(X_t \mid \tilde{H}_t);$$

cf. (18). Let s_{rt} denote the age of the data items for region r used in calculating the prediction A_t; then [5] from (21),

5. This use of the symbol "r" in the remainder of this subsection should not be confused with its use in Sec. 2 and Sec. 3.1.

$$(30)\qquad A_t = \sum_r A_{rt},$$
$$A_{rt} = \gamma^{s_{rt}} \sum_{i \in r} X_{i,\, t - s_{rt}}.$$

There are two cases, according as new data were or were not used when the computation of A_t was begun. If new data for region r were used, then $s_{rt} = D$; otherwise $s_{rt} = 1 + s_{r,\, t-1}$. Hence, by (30)

$$(31)\qquad \begin{cases} \gamma A_{t-1}, & \text{if no new data,} \\ A_t = \gamma (A_{t-1} - A_{r,\, t-1}) + \gamma^D \sum_{i \in r} X_{i,\, t-D}, & \\ & \text{if new data were used for region } r. \end{cases}$$

Since each region has M sources, we see from (31) that the computation of A_t requires the addition of M data items every B periods, plus a few "overhead" operations every period, this latter number being bounded by a number independent of B, M, and N.

The data from any single source are used once every BR periods. From (31) and (22), the total gain for one source during such a "cycle" is at most (and approximately)

$$(\gamma^{2D} + \gamma^{2(D+1)} + \gamma^{2(D+2)} + \ldots + \gamma^{2(D+BR-1)})\xi^2 = \frac{\gamma^{2D}(1 - \gamma^{2BR})\xi^2}{1 - \gamma^2};$$

here D is the time required to add the M data items in a region; it is therefore a lower bound on the time to compute each prediction A_t, but also approximately equal to that time if M is large. Hence the average per period of the total gain from all sources is

$$(32)\qquad G \leqslant \left(\frac{M}{B}\right) \frac{\gamma^{2D}(1 - \gamma^{2BR})\xi^2}{(1 - \gamma^2)};$$

recall that $R = (N/M)$.

We now show that

$$(33)\qquad \lim_{N \to \infty} \frac{G}{N} = 0,$$

however D, M and B are chosen. The intuition is that as $N \to \infty$, either the average frequency with which each source is sampled also goes to infinity, or the average number of sources processed goes to infinity and thus so does the average processing delay. Either way, the predictions are conditioned on information whose age goes to infinity with N.

This basic intuition applies to any class of policies, but here is the formal argument for the class described above. Recall that

$$D \geqslant 1 + \log_2 M;$$

then

$$\frac{G}{N} \leqslant \left(\frac{M}{N}\right) f(N, M, B) \equiv g(N, M, B),$$

where

$$f(N, M, B) \equiv \frac{\gamma^{2\log_2 M}(1-\gamma^{2BR})\xi^2}{B(1-\gamma^2)}.$$

Note that f is bounded, and that

$$\lim_{M \to \infty} f(N, M, B) = 0,$$

uniformly in N and B (since $B \geqslant 1$). On the one hand

$$\lim_{\substack{N \to \infty \\ M \leqslant N^{1/2}}} g(N, M, B) = 0,$$

because $(M/N) \to 0$ and f is bounded. On the other hand

$$\lim_{\substack{N \to \infty \\ M > N^{1/2}}} g(N, M, B) = 0,$$

because $M \leqslant N$ and $f(N, M, B) \to 0$, which proves (33).

Hence, even with this general class of policies, it is still true that the gross profit with optimal information processing grows at the same rate (to $+\infty$ or $-\infty$) as does the gross profit without any information processing, as N increases without limit. Since for small N information processing may lead to strictly higher net profits, there are decreasing returns to scale in the approach to the limit (although possibly also increasing returns to scale in some range).

3.4. Example 2

Our second specific example combines assumptions LL (Linear Loss) and CN (Common Noise). We shall see that, in contrast with Example 1, in this case one has increasing, but asymptotically constant, returns to scale, and that the asymptotic per-unit gain from information processing is strictly positive.

We can summarize assumption CN by:

$$X_{it} = Y_t + Z_{it},$$
$$Y_t = \beta Y_{t-1} + V_t,$$

(V_t) and (Z_{it}) are *all* independent, Gaussian, with mean 0,

$$EV_t^2 = v^2, \quad EZ_{it}^2 = \zeta^2,$$

(Y_t) stationary, $|\beta| < 1$, $\eta^2 \equiv \mathrm{Var}(Y_t) = \dfrac{v^2}{1-\beta^2}$.

Let $Z_t \equiv \sum_i Z_{it}$ and, as in (3.15), $X_t \equiv \sum_i X_{it}$; then

$$X_t = NY_t + Z_t. \tag{34}$$

As before, let $\tilde{H}_t$ denote the information on which the prediction A_t of X_t is conditioned. We shall derive the optimal predictions below, but we first observe that, since the Z_t are independent and identically distributed with mean 0 and variance $N\zeta^2$, and the processes (Y_t) and (Z_t) are mutually independent,

$$\text{Var}(X_t \mid \tilde{H}_t) = N^2 \text{ Var}(Y_t \mid \tilde{H}_t) + N\zeta^2. \tag{35}$$

If we write

$$\begin{aligned} \tilde{\xi}_t^2 &\equiv \text{Var}(X_t \mid \tilde{H}_t), \\ \tilde{\eta}_t^2 &\equiv \text{Var}(Y_t \mid \tilde{H}_t), \end{aligned} \tag{36}$$

then (35) can be rewritten as

$$\tilde{\xi}_t^2 = N^2 \tilde{\eta}_t^2 + N\zeta^2. \tag{37}$$

As shown in the Appendix, the expected loss, $E\Psi(A_t, Q_t)$, in period t due to prediction error is

$$L(t) = \tilde{\xi}_t L_1, \tag{38}$$

where L_1 is the loss if $(A_t - Q_t)$ is Gaussian with mean zero and variance 1. Define the long-run-average loss by

$$\bar{L} \equiv \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} L(t) \tag{39}$$

(we consider only policies for which this limit exists). The total "cost", *i.e.*, the sum of the loss due to prediction error and the cost of the information-processing network P (with p processors), is

$$\bar{C} \equiv \bar{L} + \varphi p, \tag{40}$$

where φ is the cost per processor per period. For each N, one wants to choose a network and a prediction policy to minimize $\bar{C}$. We shall show that $(1/N) \min \bar{C}$ is decreasing in N, but that

$$\lim_{N \to \infty} \left(\frac{1}{N}\right) \min \bar{C} > 0; \tag{41}$$

hence, there are decreasing, but asymptotically constant, average costs per "unit size". Furthermore, we shall show that, if φ is not too large, the left-hand side of (41) is strictly less than the long-run-average loss from the prediction error one would have with no information (and no information processing).

In fact, one can prove that $(1/N) \min \bar{C}$ is decreasing in N without knowing the optimal network and policy. From (35)-(38) we see that, for any fixed N and P

$$L(t) = N\left[\tilde{\eta}_t^2 + \frac{\zeta^2}{N}\right]^{1/2} L_1, \tag{42}$$

where $\tilde{\eta}_t^2$ is determined by the policy. The total cost in period t is therefore

$$C(t) = N\left[\tilde{\eta}_t^2 + \frac{\zeta^2}{N}\right]^{1/2} L_1 + \varphi p. \tag{43}$$

For any $N' > N$ one can achieve the same $\tilde{\eta}_t^2$, with the same p, simply by conditioning on the same information, *i.e.*, by ignoring the additional information made available by adding new sources. Hence, if we fix the network and policy in this sense and let N increase, $C(t)/N$ will decrease in N, and hence so will $\bar{C}/N$. We write

$$\frac{\bar{C}}{N} = c(N; \alpha, P) \tag{44}$$

in order to emphasize the dependence of $\bar{C}/N$ on the size N, the policy α, and the network P. If a pair (α, P) is not well defined for a given N, we assign $c(N; \alpha, P)$ the value $+\infty$. If (α, P) is well defined for N, then it is also well defined for all $N' > N$, in the sense described above. Thus, for each pair (α, P), the function $c(.; \alpha, P)$ is nonincreasing in N, and is strictly decreasing where finite. Define

$$\hat{c}(N) = \min_{(\alpha, P)} c(N; \alpha, P);$$

then $\hat{c}$ is strictly decreasing in N, which is what we wished to prove.

To obtain more detailed information about the function $\hat{c}$, we need information about the optimal pair (α, P) for given N. It is shown in the Appendix that the optimal prediction conditioned on the information $\tilde{H}_t$ is

$$A_t = NE(Y_t \mid \tilde{H}_t) + \tilde{\eta}_t a^1, \tag{45}$$

where $\tilde{\eta}_t$ is defined in (36) and a^1 is a constant. To determine a^1, for the moment let F denote the standard Gaussian cumulative distribution function (mean 0, variance 1); then a^1 is the solution of

$$\frac{F(a^1)}{1 - F(a^1)} = \frac{\psi_0}{\psi_1}, \tag{46}$$

where ψ_0 and ψ_1 are the coefficients in the piecewise-linear loss function (14). Recall that the corresponding expected loss is given by (36)-(38),

$$L(t) = \tilde{\xi}_t L_1$$
$$\tilde{\xi}_t^2 = N^2 \tilde{\eta}_t^2 + N\zeta^2$$
$$\tilde{\eta}_t^2 = \mathrm{Var}(Y_t \mid \tilde{H}_t).$$

Thus the information affects the expected loss only through the conditional variance of Y_t given $\tilde{H}_t$. In a sense, each data item X_{is} provides a "noisy" observation of Y_s, where Z_{is} is the "noise". Since the variables (Z_{is}) are independent and identically distributed, and are independent of the process (Y_s), and since the (Y_s) process is Markovian, all data items X_{is} from the same date s provide equally valuable information, and more recent data items are more valuable than less recent ones. These considerations lead

us to the following class of policies: every B periods one selects M sources and uses the most recent data items from these sources together with previously stored results to calculate a prediction; in between these periods one simply updates the prediction using previously stored results without incorporating new data items. Without loss of generality, the same M sources can be used every B periods.

To simplify the exposition, we give the details of the analysis only for the case $B=1$. (At the end of the section, we sketch the extension of the argument to the case in which B is chosen optimally.) Thus, at time t we wish to predict X_t from the history of M sources up through some date $t-D$. Because of the stationarity of the processes, it is sufficient to take $t-D=0$; denote the corresponding history by I_0, *i. e.*,

$$\tilde{H}_t = I_0 \equiv \{ X_{is},\ i=1,\ \ldots,\ M,\ s \leqslant 0 \}.$$

From (42) and (45), we are interested in calculating

$$\begin{cases} \tilde{Y}_t \equiv E(Y_t \mid I_0) \\ \tilde{\eta}_t^2 \equiv \mathrm{Var}(Y_t \mid I_0). \end{cases} \tag{47}$$

For the purpose of this calculation, define

$$\begin{cases} x_s \equiv \dfrac{1}{M} \sum_{i=1}^{M} X_{is}, \\ z_s \equiv \dfrac{1}{M} \sum_{i=1}^{M} Z_{is}; \end{cases} \tag{48}$$

then the stationary process (x_s, Y_s, z_s) is determined by:

$$\begin{cases} x_s = Y_s + z_s, \\ Y_s = \beta Y_{s-1} + V_s, \end{cases} \tag{49}$$

where (z_s) and (V_s) are independent and Gaussian, with mean 0 and with

$$\begin{cases} EV_s^2 = v^2, \\ E z_s^2 = \dfrac{\zeta^2}{M}. \end{cases} \tag{50}$$

Hence

$$\tilde{Y}_t = E(x_t \mid I_0) \equiv \tilde{x}_t. \tag{51}$$

It can be shown (*see* YAGLOM, **1962, pp.** 119-121) that $\tilde{x}_t$ has the form

$$\tilde{x}_t = \beta^{t-1}(\beta - \alpha) \sum_{s=0}^{\infty} \alpha^s x_{-s}, \tag{52}$$

where α is a constant depending on the parameters of the process (*see* below), and $|\alpha| < 1$. It follows that, if

$$\tilde{x}_{t+1} = E(x_{t+1} \mid I_1),$$

then

$$\tilde{x}_{t+1} = \beta^{t-1}(\beta-\alpha)\sum_{s=0}^{\infty}\alpha^s x_{1-s} \tag{53}$$
$$= \beta^{t-1}(\beta-\alpha)x_1 + \alpha\tilde{x}_t.$$

In other words, in order to calculate $\tilde{x}_{t+1}$, we need only store $\tilde{x}_t$, calculate the sum x_1, and then calculate the linear function (53). Thus, except for a few "overhead operations", the number of which is independent of M and N, what is required is to add up the M data items X_{i1} to form x_1.

We turn now to $\tilde{\eta}_t^2 \equiv \mathrm{Var}(Y_t \mid I_0)$. A straightforward calculation shows that

$$\tilde{\eta}_t^2 = \beta^{2(t-1)} f(M) + (1-\beta^{2(t-1)})\eta^2, \tag{54}$$

where

$$f(M) \equiv \mathrm{Var}(Y_1 \mid I_0) \tag{55}$$
$$= E\{(\tilde{Y}_1 - Y_1)^2 \mid I_0\},$$

and

$$\tilde{Y}_1 = E(Y_1 \mid I_0) = E(x_1 \mid I_0) \equiv \tilde{x}_1. \tag{56}$$

From (49)-(50),

$$\begin{aligned} K(M) &\equiv E\{(\tilde{x}_1 - x_1)^2 \mid I_0\} \\ &= E\{(\tilde{Y}_1 - x_1)^2 \mid I_0\} \\ &= E\{(\tilde{Y}_1 - Y_1 - z_1)^2 \mid I_0\} \\ &= E\{(\tilde{Y}_1 - Y_1)^2 - 2(\tilde{Y}_1 - Y_1)z_1 + z_1^2 \mid I_0\} \\ &= f(M) - 0 + \frac{\zeta^2}{M}. \end{aligned}$$

Hence

$$f(M) = K(M) - \frac{\zeta^2}{M}. \tag{57}$$

It can be shown (again, *see* YAGLOM, *op. cit.*) that

$$K(M) = \frac{\zeta^2\beta}{M\alpha}, \tag{58}$$

where α (the same constant as in (52)) is the root of

$$\alpha^2 - B(M)\alpha + 1 = 0 \tag{59}$$

that satisfies $|\alpha| < 1$, and

$$B(M) \equiv \frac{1}{\beta}\left[(1+\beta^2) + \frac{M v^2}{\zeta^2}\right]. \tag{60}$$

Alternatively, a little algebra applied to (58)-(60) shows that K(M) is a root of

$$K^2 - \bar{B}(M)K + \left(\frac{\zeta^2\beta}{M}\right)^2 = 0,$$

where

$$\bar{B}(M) \equiv \frac{\zeta^2(1+\beta^2)}{M} + v^2.$$

Hence

$$\lim_{M \to \infty} K(M) = \lim_{M \to \infty} \bar{B}(M) = v^2, \tag{61}$$

and so, from (57),

$$\lim_{M \to \infty} f(M) = v^2.$$

Since increasing M refines the information I_0, we know from general considerations that f is decreasing in M, and that, for $M \geq 1$, $f(M) < \eta^2$. We can summarize these last facts as follows:

$$v^2 < f(M) < \eta^2 = \frac{v^2}{1-\beta^2}, \tag{62}$$

$$f(M) \downarrow v^2 \quad \text{as} \quad M \uparrow \infty.$$

Recall that our prediction of X_t is conditioned on I_0, so that $t = D$. Combining (42), (54) and (55), we see that the average loss per period is

$$\bar{L} = N\left[\tilde{\eta}_D^2 + \frac{\zeta^2}{N}\right]^{1/2} L_1, \tag{63}$$

$$\tilde{\eta}_D^2 = \beta^{2(D-1)} f(M) + (1 - \beta^{2(D-1)})\eta^2.$$

The delay D, will in turn depend on M and the number of processors, p.

$$D = \tilde{D}(M, p). \tag{64}$$

Recall that given N, M and p are chosen to minimize

$$\bar{C} = \bar{L} + \varphi p,$$

or equivalently, to minimize

$$\frac{\bar{C}}{N} = \left[\tilde{\eta}_D^2 + \frac{\zeta^2}{N}\right]^{1/2} L_1 + \frac{\varphi p}{N}. \tag{65}$$

Note that $f(M)$, and hence $\tilde{\eta}_D^2$, are bounded between v^2 and η^2, and independent of N. Hence the optimal pair (M, p) will be approximately independent of N for large N; let $(\hat{M}, \hat{p})$ denote the limit of these optimal pairs as N increases without limit, and let $\hat{D}$ denote the corresponding

delay. In what follows, we assume that φ is small enough (processors are cheap enough) so that $\hat{M} \geq 1$. It follows from (65) that

$$\lim_{N \to \infty} \min\left(\frac{\bar{C}}{N}\right) = \tilde{\eta}_{\hat{D}} L_1. \tag{66}$$

On the other hand, the total expected loss per period without information (an without information processing) is

$$\bar{I} \equiv N\left[\eta^2 + \frac{\xi^2}{N}\right]^{1/2} L_1,$$

so that

$$\lim_{N \to \infty} \frac{\bar{I}}{N} = \eta L_1. \tag{67}$$

Since $\tilde{\eta}_{\hat{D}} < \eta$, we have the desired result, from (54), (66) and (67):

$$v L_1 < \lim_{N \to \infty} \min\left(\frac{\bar{C}}{N}\right) < \lim_{N \to \infty} \left(\frac{\bar{I}}{N}\right). \tag{68}$$

If we allow $B > 1$, then (as in Example 1), the delay from the start of the information processing to the actual use of the prediction will vary cyclically from D to $(D + B - 1)$. The average expected loss per period, $\bar{L}$, will then be an average of terms like (63), and the results will be qualitatively similar to those for the case $B = 1$. In particular, the conclusion (68) will still be valid.

3.5. Further Remarks

We derived results for the pairs of assumptions IN/QL (Example 1) and CN/LL (Example 2). What about the combinations IN/LL and CN/QL?

The general model was set up so that if there were no prediction error, there would be constant returns to scale. To isolate the effect that information processing has on returns to scale, it is also helpful that when all the sources are processed with a fixed delay (or when no information is processed at all), the returns to scale are constant, or at least asymptotically constant.

With the Idiosyncratic Noise assumption, there is a law of large numbers effect that is exactly offset by the Quadratic Loss assumption, so that returns to scale are constant for a fixed delay in Example 1. If we move to a linear loss (IN/LL), the strong law of large numbers implies that the average loss converges to zero, even if no information is processed at all. Thus, it is difficult to isolate the returns to scale due to information processing. One can at least conclude, albeit trivially, that again the average gross gain from processing information converges to zero, and so asymptotically no information is processed at all.

With the Linear Loss and Common Noise assumptions, the idiosyncratic component of the noise also asymptotically has a negligible effect on the average loss because of the law of large numbers. Specifically, the term ζ^2/N in equation (42) goes to 0 as N rises. However, the loss caused by the common noise term is proportional to the size of the firm when no information is processed, so that the average loss decreases with N but converges to a strictly positive number. Furthermore, the increasing returns to scale found in Example 2 are due in part to increasing returns in diminishing the loss from the common noise term, and not simply to the inherent decreasing returns to scale in the example.

If we move to a quadratic loss (CN/QL), the loss due to the idiosyncratic noise is now proportional to N, while the loss due to the common noise is proportional to N^2, for a fixed delay. Thus, there are inherently decreasing returns to scale, and it is again difficult to isolate the returns to scale due to information processing. We can at least say that, given the lower bound on processing delay, the sign of the returns to scale cannot be reversed by information processing.

4 Bibliographic Notes

McGuire and Marschak [1971] were perhaps the first to propose the model of a finite automaton as a formalization of the notion of a boundedly rational decision-maker; the individual processors in our models are specialized automata. Bounded rationality in economic decision-making has of course been emphasized by Simon [1972]; see also his interesting discussion of the significance of hierarchy in the design and organization of tasks [1981, Ch. 5]. For a more general discussion of the economics of managing, see (Radner [1989]). The model of a decision-making organization as a "network" of processors was explored by Marschak and Radner [1972], but only a few simple examples were studied in any detail. In a similar spirit, Marschak and Reichelstein [1987] studied conditions under which a "hierarchical" structure of decision-making would be efficient in a broader set of structures. In their model, every processor is also responsible for the final decision about some action variable, and the only cost of processing is that of communication. Their analysis derived some conditions under which hierarchy would be preferred, but we shall not attempt to summarize their results here.

Using a network model that is similar in spirit to ours, Mount and Reiter [1982, 1990] have studied the computational complexity of certain resource allocation mechanisms. In their model, a typical processor computes—in one unit of time—the value of some specific function, say $y = f(x_1, \ldots, x_n)$; furthermore, each of the variables in question, y, x_1, x_2, $\ldots$, x_n, is a vector of some specified dimension, say d. The processors are connected by a network (directed graph) of one-way communication

links, and after each computation cycle (unit of time) each processor sends its output to every processor to which it directly connected, or to the "outside"; such communications are instantaneous. An (r, d, F)-network is a network of processors, as above, such that:

1. No processor has more than r inputs ($s \leqslant r$);
2. Each processor computes some function f in a prespecified set F.

Roughly speaking, a resource allocation mechanism is implemented (realized) by having the economic agents communicate certain information to the network, which then computes the "equilibrium" of the mechanism. For any specification of r, d, and F, and any mechanism M, there will be some minimum time, say $t(M; r, d, F)$, required to compute an equilibrium of M with some (r, d, F)-network. Mount and Reiter show that there can be a tradeoff between $t(M; r, d, F)$ and the "amount" of information communicated by the agents, holding r, d, and F fixed. (Note that the agents are *outside* the network.) However, in the work cited, Mount and Reiter do not explicitly impute any cost to the number of processors, nor do they consider in a systematic way how $t(M; r, d, F)$ depends on r, d, and F.

MARSCHAK [1972] and MARSCHAK and RADNER [1972, Ch. 7] studied the effect of delay on the value of decisions, in some particular examples. The present paper is, in part, an attempt to pursue the program of research implicit in that earlier work.

Our approach has also been strongly influenced by the computer-science literature on parallel processing. In particular, the characterization of efficient one-shot networks in Section 2 has appeared in one form or another (*see* GIBBONS and RYTTER [1988, p. 12]). On the other hand, our characterization of efficient *renewal* networks seems to be new.

The Piece-wise Linear Loss Function

Consider the piece-wise linear loss function (14) used in Example 2 (assumption LL):

$$\Psi(A, Q)=\begin{cases}\psi_0(Q-A), & A\leqslant Q,\\ \psi_1(A-Q), & A\geqslant Q,\end{cases} \tag{69}$$

where ψ_0 and ψ_1 are strictly positive constants. Let F denote the cumulative distribution function (cdf) of the random variable Q; assume that F is absolutely continuous. For any fixed A, the expected loss is

$$\begin{aligned}L(A)&=E\Psi(A, Q)\\ &=\psi_1\int_{-\infty}^{A}(A-q)F'(q)\,dq+\psi_0\int_{A}^{\infty}(q-A)F'(q)\,dq.\end{aligned}$$

One easily verifies that the first and second derivatives of L are

$$\begin{aligned}L'(A)&=\psi_1 F(A)-\psi_0[1-F(A)],\\ L''(A)&=(\psi_1+\psi_0)F'(A).\end{aligned} \tag{70}$$

Hence L is convex and L′ is continuous; furthermore, $L'(A)<0$ for A sufficiently small, and $L'(A)>0$ for A sufficiently large. It follows that the optimal value of A is the solution of

$$\frac{F(A)}{1-F(A)}=\frac{\psi_0}{\psi_1}. \tag{71}$$

In particular, let a^1 denote the solution of (71) when F is the cdf of the standard Gaussian distribution (mean 0 and variance 1), and let L_1 be the corresponding minimum expected loss, $L(a^1)$.

It is straightforward to show that if Q* is a random variable, $Q\equiv hQ^*+k$, where $h>0$, A* is the optimal prediction of Q*, and A is the optimal prediction of Q, then

$$A=hA^*+k.$$

Furthermore, if L* and L are the minimum expected losses for A* and A, resp., then

$$L=hL^*.$$

In particular, if Q has the Gaussian distribution with mean μ and variance σ^2, then the optimal prediction of Q is $\sigma a^1+\mu$, and the corresponding minimum expected loss is σL_1.

• References

AOKI, M. (1986). – "Horizontal vs. Vertical Information Structure of the Firm", *American Economic Review*, 76, pp. 971-983.

CALVO, G. and WELLISZ S. (1978). – "Supervision, Loss of Control and the Optimal Size of the Firm", *Journal of Political Economy*, 86, pp. 943-952.

CREMER J. (1980). – "A Partial Theory of the Optimal Organization", *The Bell Journal of Economics*, 11, pp. 683-693.

GEANAKOPLOS, J. and MILGROM P. (1991). – "A Theory of Hierarchies Based on Limited Managerial Attention", *Journal of the Japanese and International Economies*, 5, pp. 205-225.

GIBBONS, A. and RYTTER, W. (1988). – *Efficient Parallel Algorithms*, Cambridge University Press, Cambridge.

KEREN, M. and LEVHARI, D. (1983). – "The Internal Organization of the Firm and the Shape of Average Costs", *The Bell Journal of Economics*, 14, pp. 474-486.

KEREN, M. and LEVHARI, D. (1979). – "The Optimum Span of Control in a Pure Hierarchy", *Management Science*, 11, pp. 1162-1172.

MARSCHAK, J. and RADNER, R. (1972). – *Economic Theory of Teams*, Yale University Press, New Haven.

MARSCHAK, T. A. (1972). – "Computation in Organizations", in McGuire and Radner, pp. 237-282.

MARSCHAK, T. A. and MCGUIRE, C. B. (1971). – "Lecture Notes on Economic Models for Organization Design", Graduate School of Business Administration, Univ. of Calif., Berkeley.

MARSCHAK, T. A. and REICHELSTEIN, S. (1987). – "Network Mechanisms, Informational Efficiency, and the Role of Hierarchies", Graduate School of Business Administration, Stanford University, Stanford, CA (unpublished).

MCAFEE, R. P. and MCMILLAN, J. (1990). – "Organization Diseconomies of Scale", University of California, San Diego (unpublished).

MCGUIRE, C. B. and RADNER, R. (1972). – *Decision and Organization*, North-Holland, Amsterdam; 2nd ed., University of Minnesota Press, Minneapolis, 1986.

MELAMUD, N., MOOKHERJEE, D. and REICHELSTEIN, S. (1988). – "Hierarchical Decentralization of Incentive Contracts", Stanford University (unpublished).

MINTZBERG, H. (1989). – *Mintzberg on Management*, The Free Press, New York.

MOUNT, K. R. and REITER, S. (1982). – "Computation, Communication, and Performance in Resource Allocation", paper presented at the CEME-NBER Decentralization Seminar, University of Minnesota, Minneapolis, May 21-23, 1982.

MOUNT, K. R. and REITER, S. (1990). – "A Model of Computing with Human Agents", Discussion Paper No. 890, Center for Mathematical Studies in Economics and Management Science, Northwestern University, Evanston, IL (unpublished).

RADNER, R. (1989). – "Hierarchy: The Economics of Managing", Marshall Lectures, Cambridge University, October (unpublished), to appear in *J. of Econ. Lit.*

RADNER, R. (1990). – "The Organization of Decentralized Information Processing", AT&T Bell Laboratories, February (unpublished); presented at the NBER-CEME Decentralization Seminar, Northwestern University, April 27-29, 1990.

SCHWARTZ, J. T. (1980). – "Ultracomputers", *ACM Transactions on Programming Languages and Systems*, 2, pp. 484-521.

SIMON, H. A. (1972). – "Theories of Bounded Rationality", *in* MCGUIRE and RADNER, pp. 161-176.

SIMON, H. A. (1981). – *The Sciences of the Artificial*, The MIT Press, Cambridge (2nd ed.).

VAN ZANDT, T. (1990). – "Efficient Parallel Addition", AT&T Bell Laboratories, Murray Hill, NJ (unpublished).

WALD, A. (1950). – *Statistical Decision Functions*, John Wiley, New York.

WILLIAMSON, O. (1967). – "Hierarchical Control and Optimum Firm Size", *Journal of Political Economy*, 75, pp. 123-138.

YAGLOM, A. M. (1962). – *An Introduction to the Theory of Stationary Random Functions*, Prentice-Hall, Englewood Cliffs, NJ.