DISCUSSION PAPER SERIES

No. 4397

SPEED AND QUALITY OF COLLECTIVE DECISION-MAKING II: INCENTIVES FOR INFORMATION PROVISION

Hans Peter Grüner and Elisabeth Schulte

INDUSTRIAL ORGANIZATION and PUBLIC POLICY

Centre for Economic Policy Research

www.cepr.org

ISSN 0265-8003

SPEED AND QUALITY OF COLLECTIVE DECISION-MAKING II: INCENTIVES FOR INFORMATION PROVISION

Hans Peter Grüner, Universität Mannheim and CEPR
Elisabeth Schulte, Universität Mannheim

Discussion Paper No. 4397
April 2005

Centre for Economic Policy Research
90–98 Goswell Rd, London EC1V 7RR, UK
Tel: (44 20) 7878 2900, Fax: (44 20) 7878 2999
Email: cepr@cepr.org, Website: www.cepr.org

CEPR Discussion Paper No. 4397

April 2005

ABSTRACT

Speed and Quality of Collective Decision-Making II: Incentives for Information Provision*

This Paper provides a game theoretic extension of Radner's (1993) model of hierarchical information aggregation. It studies the role of the hierarchy design for the speed and quality of a collective decision process. The hierarchy is described as a programmed network of agents. The programme describes how information is processed within the network. The network of P identical managers has to aggregate information in the form of a set of n data items in order to make an informed decision. Each manager benefits from reaching an accurate decision but suffers from an individual cost of effort, which has to be provided in order to understand the information contained in a data item properly. We find that decentralized information processing increases incentives for information provision. There may be boundaries on the appropriate extend of decentralization, however. We also compare three different hierarchy designs: two balanced hierarchies and the fastest (skip-level) hierarchy, proposed by Radner. Skip-level reporting outperforms balanced hierarchies in terms of decision speed and in terms of decision quality.

JEL Classification: D23, D70, D83, L22 and P51
Keywords: hierarchies, incentives for information provision and information processing

Hans Peter Grüner
Lehrstuhl für Wirtschaftspolitik
Universität Mannheim
68131 Mannheim
GERMANY
Tel: 181 1886
Fax: 181 1884
Email: hgruener@rumms.uni-mannheim.de

For further Discussion Papers by this author see:
www.cepr.org/pubs/new-dps/dplist.asp?authorid=144072

Elisabeth Schulte
Department of Economics
Universität Mannheim
L7, 3-5
D-68131 Mannheim
GERMANY
Email: elisabeth.schulte@vwl.uni-mannheim.de

For further Discussion Papers by this author see:
www.cepr.org/pubs/new-dps/dplist.asp?authorid=160022

*We thank Enno Mammen for helpful remarks on Proposition 1 and participants of the workshop on federalism at ZEI Bonn, the workshop on international negotiation systems at Mannheim, and the 2004 ENTER Jamboree in Barcelona for helpful comments and suggestions. Elisabeth Schulte acknowledges financial support through the DFG research group on the institutionalization of international negotiation systems (IINS).

Submitted 16 February 2004

1 Introduction

Most collective decision processes are evaluated both in terms of the quality of decisions as well as in terms of the speed at which the decisions are reached. In general the design of an institution that governs such a decision will affect both dimensions of the outcome. This paper provides a game theoretic extension of Radner's (1993) model of decentralized information aggregation and studies the role of the hierarchy design for the speed and quality of collective decision processes in this extended framework.

In our paper, a hierarchy is modeled as a programmed network of agents. The programme describes how information is processed within the network. It specifies the assignment of information to the various decision makers and the sequence of information processing and message stages. Moreover, it underlies certain restrictions related to the amount of time that some operations consume.

We assume that individuals working in the hierarchy can only process information properly if they provide the necessary amount of effort. This individual effort is unobservable for other players. A programmed network therefore induces a game among its members. We characterize the equilibria of this game in terms of (i) the speed at which the decision is reached and (ii) the cost and quality of the decision.

We represent the information processing task by the problem of adding up the realization of a given number of i.i.d. random variables. In performing their tasks individuals may decide whether to provide effort or not. If an agent does not provide effort then he does not obtain any information about the realization of the random variable that he is supposed to handle. If he provides effort when reading a number he has full information about the respective number. All individuals in the hierarchy are interested in the quality of the overall result but they also suffer from an individual cost of providing effort.[1]

[1]We exclude any transfers to the agents that are related to their reports or to the quality of the

In a first step we study the case of a single decision maker whose task is to add n numbers. It turns out that - for some particular distributions of the random variables, the marginal gain from adding additional numbers increases when the set of remaining numbers is smaller. This yields a new rationale for decentralized information processing in hierarchies: an equilibrium where all individuals always provide effort exists if the number of objects assigned to each member is sufficiently small. This is why parallel information processing is not only faster than full centralization - it also provides better incentives for the participating managers. The reason for this lies in the complementarity of effort of different managers which arises endogenously in our model.

In a second step we turn to the comparison of different multi-layer hierarchies. In his seminal paper Radner (1993) studies the design of hierarchical structures when information processing takes time. This leads him to propose a hierarchical structure - which he calls a reduced tree - within which information is processed at maximum speed. The virtue of the reduced tree is that processors on all levels simultaneously process information right from the beginning. This guarantees maximal parallel processing and reduces the delay of the entire information processing procedure. We study the properties of the reduced tree in our modified setup. It turns out that it has virtues that lie beyond the efficient treatment of information characterized in Radner's paper.

The evaluation of a hierarchy focuses on three dimensions: (i) the speed as well as (ii) the cost of information processing (i.e. the number of operations to be performed), and (iii) the quality of the decision. We compare the performance of the reduced tree to two alternative balanced hierarchy designs. One is the steepest possible balanced

---

entire decision. Hence, our model relates to situations where the quality is not ex-post verifiable and where (i) the reports costitute soft information or (ii) transfers generate additional costs.

hierarchy which we call the "2^T-Tree" and the other one is the flattest possible hierarchy: the "centralized tree". The 2^T-Tree is the balanced hierarchy with the maximum number of hierarchy levels: every manager has at most two immediate subordinates. In the centralized tree there is one top manager and all the other agents are his immediate subordinates. Our second main result is that speed and quality may not be conflicting objectives when it comes to the evaluation of hierarchies. Reduced trees à la Radner outperform other arrangements in terms of both dimensions.

Our main result is that any programmed network $\tilde{N}$ in which all agents provide effort can - under certain restrictions - be replaced by a reduced tree $R(\tilde{N})$ with the same number of processors that also has a full effort equilibrium. Therefore $R(\tilde{N})$ generates the same classical surplus and a faster decision.

This paper is related to a recent literature on organization design that draws on insights from computer science, starting with Radner (1992, 1993). The reduced tree is designed for one-shot problems (to which we restrict attention), i.e. there is only one set of data to be processed, or the processing of the data is finished before another calculation task occurs. Van Zandt (1997, 1998) and Meagher, Orbay and Van Zandt (2001) study the case when new data comes in before the processing of the old set is finished. Orbay (2002) adds the frequency with which new data arrives as a new dimension to the analysis of efficient hierarchies. Prat (1997) studies hierarchies in which a manager's ability is heterogenous, i.e. some managers are able to work faster than others, and the wage a manager is paid is a function of his ability. It turns out that with these modifications - except for the one made by Prat (1997) - the reduced tree is still (close to) efficient.

In Radner's model - and in most of the information processing literature following - individuals are thought of as machines, doing what they are programmed to do. There are at least two important features of information processing by human agents which may require modifications of this basic model. Firstly, an individual's calculation ability might not be perfect, i.e. occasionally individuals may make mistakes. Secondly, when delegating tasks, one has to make sure that the agent has an incentive to perform them. Thus, there emerges another dimension for the evaluation of hierarchies: the quality of the final result.

The joint analysis of speed and quality of hierarchical decision processes has previously been studied in Jehiel (1999) and also in Schulte and Grüner (2004). Jehiel considers the case where some signals get lost in the hierarchy with an exogenous probability, depending on the size of the groups of which the hierarchy consists. Schulte and Grüner study the role of the hierarchy design when individuals make mistakes with an exogenously given probability. In the present paper the quality of information collected is endogenous. The result that the reduced tree provides a better decision quality than a balanced hierarchy is the same.

Our paper is also related to a huge game theoretic literature that studies incentives in hierarchies, in particular Aghion and Tirole (1997), Mookherjee and Reichelstein (1997), and Melhumad, Mookherjee and Reichelstein (1995). These papers consider problems in which certain tasks as well as authority have to be delegated (and subdelegated) to (and by) agents whose interests diverge from that of the principal. Here, delegation involves a loss of control for the principal, but strengthens the incentives for the agent. In our model, all agents have the same objective: The available information shall be processed as accurately as possible in order to make an informed decision.

The paper is organized as follows. In section 2, we introduce the structure of the hierarchies and the procedure of information processing. The model set up is presented in section 3. In section 4 we solve the problem of a sole decision maker and turn to the equilibria of the game played by the members of the hierarchies in section 5. A comparison of the hierarchies is carried out in section 6. Section 7 concludes. All omitted proofs can be found in Appendix A. More information on useful rules for binomials can be found in Appendix B.

2 Hierarchy designs

In this section, we will present three possibilities to design a network hierarchically: the reduced tree, the 2^T-Tree and the centralized tree, which are interesting for the following reasons.

The reduced tree is the one that works fastest and therefore deserves special attention in a framework analyzing speed and quality of collective decision making. Balanced trees on the other hand are the most natural form to think of delegation: If there is need for delegation, one could expect that the delegation occurs in form of a division of the task in equal pieces. The same intuition should hold for further delegation. We restrict attention to two polar forms of balanced trees: the flattest one (centralized tree) and the steepest one where each agent has at least two subordinates (2^T-Tree).

There are n data items that have to be added up by P agents called managers. Managers are endowed with an inbox, a processing unit and a memory. Information processing can be done decentralized in a programmed network.[2] As far as the time structure is concerned, we assume in this description that the agents provide effort. In one unit of time, a manager reads a number from his inbox into his processing unit and adds it to his memory. At the beginning of the procedure, the memory is set to zero. When finished with the items in his inbox (which may contain raw data as well as partial results provided by subordinates), a manager sends his partial result to his direct superior. A subordinate's partial result is processed in the same way as raw data. The top manager's output is the final result.

[2]See Radner (1993) for a more detailed description of a programmed network.

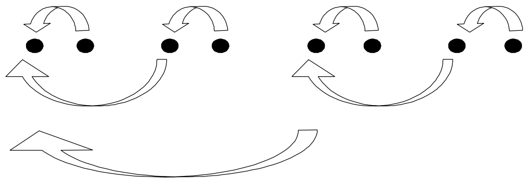

Figure 1: Construction of a reduced tree: Each even i is direct subordinate of $i-1$.

2.1 The reduced tree

For processing information in minimum time, Radner (1993) proposes the following hierarchy design: Number the managers subsequently from 1 to P and assign $\frac{n}{P}$ objects to each manager. (If $\frac{n}{P}$ is no integer, assign the largest integer smaller than $\frac{n}{P}$ to each manager and another one to the first $n \bmod P$ ones.)

Let $\lceil x \rceil$ denote the smallest integer $\geq x$. After $\lceil \frac{n}{P} \rceil$ units of time, every manager has reduced the information in his inbox to a partial result. Assign manager i's partial result to $(i-1)$'s inbox for each even i. Manager $(i-1)$ therewith becomes i's immediate superior and the number of managers still working is reduced by half. Renumber these managers in an appropriate manner and repeat the procedure until a single manager remains (the top manager). This procedure of constructing a reduced tree is illustrated in Figure 1, and Figure 2 depicts a reduced tree. The top manager has $\lceil \log_2(P) \rceil$ immediate subordinates. The n objects are reduced to one in $\lceil \frac{n}{P} \rceil + \lceil \log_2(P) \rceil$ units of time.

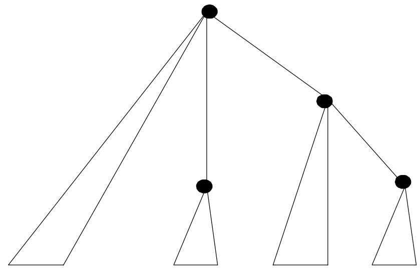

Figure 2: Reduced tree (A triangle represents a group of $\frac{n}{P}$ objects.)

2.2 Balanced trees

We now turn to the balanced trees which are characterized by two properties: (i) each manager's immediate subordinates are at the next lower level and (ii) on a given level, each manager has the same number of subordinates.[3]

We characterize two balanced tree designs: the steepest hierarchy one can build with P managers which we call "2^T-Tree" and the flattest possible balanced hierarchy which we call the "centralized tree".

2.2.1 The 2^T-tree

The 2^T-Tree is the largest balanced hierarchy one can construct with P managers. Each manager has at most two immediate subordinates.[4] Thus, on each level, twice as

[3]As far as this is possible. Call the number of managers working on the lowest level P_1 and the number of managers working on the next higher level P_2. Assign $P_1 \bmod P_2$ remaining lowest level managers as equally as possible to the next higher level managers. Apply this procedure analogously to the assignment of data on the lowest level.

[4]Again, as far as this is possible. If not, apply the same procedure as described in footnote 3.

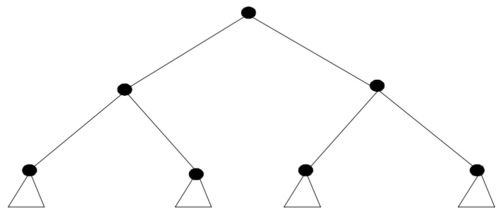

Figure 3: 2^T-Tree

much agents are working as in the next higher level. This implies that in a perfectly symmetric (strictly balanced) 2^T-Tree with P managers, there are $\frac{P+1}{2}$ managers working on level one. The number of levels in a 2^T-Tree is T. Note that in a strictly balanced 2^T-Tree there are exactly $2^T - 1$ managers.

2.2.2 The centralized tree

In the centralized tree, the $P-1$ level one-managers process $\frac{n}{P-1}$ items each (on average) and send their partial results to the top managers' inbox. This procedure takes $\lceil \frac{n}{P-1} \rceil$ units of time. Thereafter, the top manager needs again $P-1$ units of time to come to the final decision.

3 The model

3.1 The task of the hierarchy

Consider a group of $j = 1..P$ individuals whose task is to add up n numbers x_i, $i = 1..n$, where $x_i \in \{0, 1\}$. The agents are endowed with an inbox, in which raw data

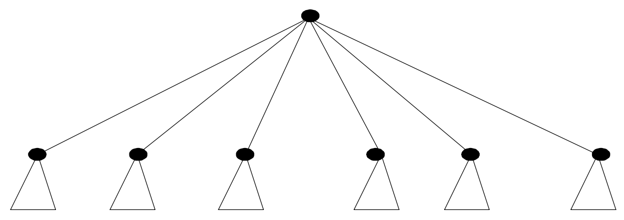

Figure 4: Centralized tree

and partial results provided by subordinates can be stored, a processing unit and a memory. The numbers x_i are independent random variables, each outcome is equally likely. Each individual may choose whether to provide effort when he reads a number from his inbox in order to obtain information about its value. Alternatively, he can make a guess. Effort comes at a fixed cost c. If the individual provides effort then he becomes aware of the true value of the number. If he does not provide effort then he gets no signal at all. Note, however, that due to the structure of the programmed network, the individual is forced to wait one unit of time even if he does not provide effort.

Individual preferences are identical for all agents. Individuals dislike deviations of the decision x from the true aggregate state of the world and suffer from providing effort. We represent these preferences by the following von Neumann-Morgenstern utility function:

$$u\left(x, \mathbf{x}, d_j\right) = -\left| x - \sum_{i=1}^{n} x_i \right| - d_j \cdot c, \tag{1}$$

where $\mathbf{x}$ denotes the vector of information and d_j measures how many times individual j provides effort. Individual effort is not observable.

This - and the underlying programmed network - fully describes a game. In this game the strategy is a plan that fixes (i) when to provide effort (possibly history dependent) and (ii) what to report (again possibly history dependent). Society has s members. Among them $s - P$ individuals are not part of the hierarchy. Their preferences are characterized by the objective function $-\left|x - \sum_{i=1}^{n} x_i\right|$.

3.2 Classical surplus and efficiency

Call the number of objects that are processed properly $m \in \{0, 1, ..n\}$. The classical surplus is defined as

$$v := -s \cdot k\left(n - m\right) - \sum_{j=1}^{P} d_j, \tag{2}$$

where s denotes population size and $k\left(l\right) = E\left(\left|\frac{l}{2} - \sum_{i=1}^{l} x_i\right|\right)$ denotes the expected cost of not properly processing l items.[5] In what follows, we assume that processing the whole set of data maximizes the classical surplus of the economy. Note that generally it is socially wasteful to add additional members to the hierarchy because this generates additional costs. Hence there are two possible reasons to decentralize information processing: (i) incentives or (ii) the speed of decision making. A hierarchy design is efficient if it is not dominated on one of the following dimensions by another one that performs at least equally well on the other dimension: (i) speed (ii) classical surplus.

[5]Similarly to Meagher and Van Zandt (1998) we consider the number of (effective) operations and not the number of managers employed as a measure of cost.

4 The role of hierarchies

4.1 A single decision maker

It is useful to first consider the case of a single individual that has to add up n numbers. At what cost $c(n)$ is he willing to provide effort when he performs this task? We start with the case of a single object, $n = 1$. The agent has the option to either (i) read the number at cost c or make a guess. The best guess is $1/2$.[6] The expected cost of guessing is $1/2$, hence $c(1) = 1/2$. Next consider an individual confronted with two objects. His best guess is $x = 1$. The cost of guessing is again $1/2$. Hence it pays to provide effort twice if $c \leq c(2) = 1/4$. Note that by providing effort once, the expected gain is zero in this case. In general we have:

Lemma 1 *Consider a single decision maker who has to add up n objects. The cost of guessing the sum is*

$$k(n) = E\left(\left|\frac{n}{2} - \sum_{i=1}^{n} x_i\right|\right) = \frac{1}{2^{n-1}} \cdot \sum_{i=0}^{\left\lfloor \frac{n}{2} \right\rfloor} \binom{n}{i} \cdot \left(\frac{n}{2} - i\right).^{7}$$

PROOF The best guess is $\frac{n}{2}$. Hence:

$$k(n) = E\left(\left|\frac{n}{2} - \sum_{i=1}^{n} x_i\right|\right). \tag{3}$$

Next note that the true result takes the value $i < \frac{n}{2}$ with probability $p^i = \frac{1}{2^n} \cdot \binom{n}{i}$. The cost of guessing in this case is $\frac{n}{2} - i$. Note that deviations are symmetric. Hence:

$$k(n) = 2 \cdot \sum_{i=0}^{\left\lfloor \frac{n}{2} \right\rfloor} \frac{1}{2^n} \cdot \binom{n}{i} \cdot \left(\frac{n}{2} - i\right) \tag{4}$$

[6] Given the structure of preferences, the expected utility maximizing guess of the sum of n i.i.d. random variables is the median realization of the sum, which coincides with its expected value $\frac{n}{2}$ in our model.

[7] $\lfloor x \rfloor$ denotes the largest integer $\leq x$.

$$= \frac{1}{2^{n-1}} \cdot \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{i} \cdot \left(\frac{n}{2} - i\right). \tag{5}$$

Q.E.D.

Proposition 1 *The function $k(n)$ is a weakly increasing step-function with diminishing increments. It can be defined*

(i) recursively

$$k(n+1) = \begin{cases} k(n) + \frac{p_{\frac{n}{2}}}{2}, & n \text{ even} \\ k(n), & n \text{ odd,} \end{cases}$$

with $p_x := \frac{1}{2^{2x}} \binom{2x}{x}$.

(ii)

$$k(n) = \frac{1}{2} + \sum_{i=1}^{\lceil \frac{n}{2} \rceil - 1} \frac{p_i}{2}.$$

(iii)

$$k(n) = \frac{1}{2} \left(1 + \frac{1}{2} \sum_{i=1}^{\lceil \frac{n}{2} \rceil - 1} \left(\prod_{j=0}^{i-2} \frac{2(1+j)+1}{2(1+j)+2}\right)\right).$$

The functional form of $k(n)$ is outlined in Figure 5.

Lemma 2 *Consider a single decision maker who has to add up n objects. It pays to provide effort n times only if*

$$c \leq c(n) = \frac{k(n)}{n} = \frac{1}{2^{n-1} n} \sum_{i=0}^{\frac{n}{2}} \binom{n}{i} \left(\frac{n}{2} - i\right).$$

Proof The cost of providing effort n times is cn. The benefit of doing so is to avoid the cost of guessing $k(n)$. The functional form of $c(n)$ follows from applying Lemma 1. Q.E.D.

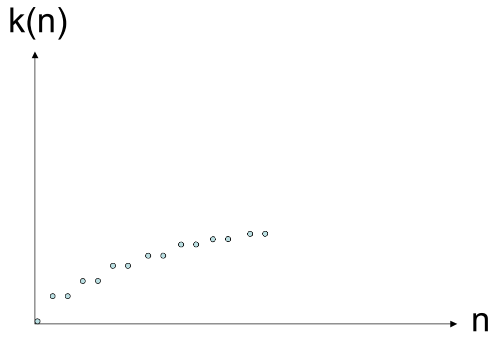

Figure 5: $k(n)$

Proposition 2 *The function $c\,(n)$ is a weakly decreasing step-function and can be defined recursively:*

$$c(n+1) = \begin{cases} c(n), & n \text{ even} \\ \frac{n}{n+1}c(n), & n \text{ odd.} \end{cases}$$

Figure 6 depicts the function $c(n)$. An immediate consequence of Proposition 2 is that:

Proposition 3 *A single decision maker who prefers providing effort n times to providing no effort will continue to provide effort, once he has provided effort the first time. The optimal strategy for the manager in the one-player game is to provide effort when reading all (none) of the n items if $c \leq (>)\, c\,(n)$.*

4.2 Division of tasks

The results from the previous section point out a major advantage of hierarchical structures. Whenever a task is divided into several parts that have to be performed

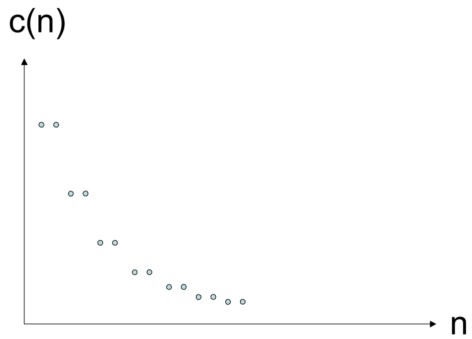

Figure 6: $c(n)$

by single individuals then the marginal gain from performing the own part must be sufficiently large in order to ensure the existence of a full effort equilibrium. Since the function $k(n)$ exhibits diminishing increments, the marginal gain becomes larger as the number of decision makers is increased, i.e. the number of items per manager is decreased. Therefore hierarchies may have the advantage of providing better incentives for information provision.

5 Equilibria in hierarchies

5.1 The equilibrium concept

We now turn to the analysis of the equilibrium in a hierarchy that consists of more than one single player. A strategy characterizes (i) whether or not a player wants to provide effort whenever the programme asks him to read up a number (depending on the history, i.e. the signals received before and the numbers read before) and (ii) the signal to be sent to any superior depending on the history. Given that nature first chooses the realizations of all random variables, the game does not have any subgames except the game itself. Therefore we concentrate on Nash equilibria. However, we require the strategies of all players to fulfill one additional requirement. Each player does not want to revise his equilibrium strategy while he plays (taking the strategies of all other players as given). This means that all strategies shall fulfill a basic requirement of time consistency.

Moreover, we restrict the analysis to equilibria where agents who are not perfectly informed about their items make the best guess when they report to their superior. The superior in turn uses the best guess in his further calculations.

5.2 Centralized tree equilibria

One obvious disadvantage of any centralized tree is that an equilibrium where all players choose not to provide effort exists independently of the parameter values. It remains to be studied whether there are additional equilibria where some or all players provide effort. In deed we find that several such equilibria may exist. Moreover, these equilibria may be ranked according to the Pareto-criterion. First it is useful to characterize these equilibria as follows:

Definition 1 *Consider a centralized tree. A strategy profile with the following properties is called an l-effort strategy profile:*
(i) l low level players provide effort, $P - 1 \geq l \geq 1$,
(ii) $P - 1 - l$ low level players do not provide effort,
(iii) the top player provides effort when reading the partial result from a low level player who belongs to the first group but not when he reads the partial result from a player who belongs to the second group.

Definition 2 *An l-effort strategy profile is an l-effort equilibrium if it is a Nash Equilibrium of the game played by the members of the centralized tree.*

We have:

Proposition 4 *(i) Every centralized tree has an equilibrium called no effort equilibrium where no player ever provides effort. In this equilibrium the top player guesses the sum. His guess is $\frac{n}{2}$.*

(ii) An l-effort equilibrium exists if and only if [8]

$$c \leq \gamma(l) = \min \left\{ \begin{array}{c} \frac{P-1}{n} \left(k \left(n - (l-1) \frac{n}{P-1} \right) - k \left(n - l \frac{n}{P-1} \right) \right), \\ \frac{1}{l} \left(k(n) - k \left(n - l \frac{n}{P-1} \right) \right) \end{array} \right\}.$$

Proof (i) Obvious.

(ii) A low level player has an incentive to provide effort $\frac{n}{P-1}$ times given the strategies of the other players if and only if $\frac{n}{P-1} c \leq \left(k \left(n - (l-1) \frac{n}{P-1} \right) - k \left(n - l \frac{n}{P-1} \right) \right)$.

(iii) It is obvious that it is a best response for the low level players not to provide effort if the top player does not provide effort when reading their partial results.

(iv) The top player has an incentive to provide effort l times reading the partial results of his subordinates who provided effort if and only if $lc \leq k(n) - k \left(n - l \frac{n}{P-1} \right)$. It is obvious that it is a best response not to provide effort when reading the partial results of those subordinates who do not provide effort. Q.E.D.

Note that due to the specific nature of the $c(n)$ function an l-effort equilibrium exists for a wider range of cost parameters if l - the number of low level players who provide effort - is larger. This is due to the fact that - with a smaller number of players providing effort - the marginal gain of providing effort for each of these l players decreases.

[8] For the ease of exposition, we drop integer restrictions at this stage of the analysis. Taking them into consideration would lead to case differentiations that do not add any insights, nor change the nature of the results.

Proposition 5 *If an l-effort equilibrium exists, then there exists an $l+1$-effort equilibrium.*

PROOF An l-effort equilibrium exists. Therefore we know that

(i) $c \leq \frac{P-1}{n}\left(k\left(n-(l-1)\frac{n}{P-1}\right)-k\left(n-l\frac{n}{P-1}\right)\right)$ and

(ii) $c \leq \frac{1}{l}\left(k(n)-k\left(n-l\frac{n}{P-1}\right)\right)$.

An $l+1$-effort equilibrium exists if

$c \leq \gamma(l+1) = \min\left\{\begin{array}{c} \frac{P-1}{n}\left(k\left(n-l\frac{n}{P-1}\right)-k\left(n-(l+1)\frac{n}{P-1}\right)\right) \\ , \frac{1}{l+1}\left(k(n)-k\left(n-(l+1)\frac{n}{P-1}\right)\right) \end{array}\right\}$ which is implied by (i) and (ii) because of the diminishing increments of $k(n)$.

Q.E.D.

Consequently we have:

Corollary 3 *(i) The $(P-1)$-effort equilibrium called full effort equilibrium always exists if an l-effort equilibrium exists.*
(ii) The full effort equilibrium exists if and only if

$$c \leq \gamma(P-1) := \min\left\{c\left(\left\lceil\frac{n}{P-1}\right\rceil\right), \frac{1}{P-1}k(n)\right\}^{9}.$$

If an l-effort equilibrium exists, there exist another $\binom{P-1}{l}-1$ equilibria in which the same number of low-level players provide effort but the identity of the effort players changes.

Proposition 6 *(i) The no effort equilibrium and the l-effort equilibria are the only equilibria of the game played by the members of a centralized tree.*
(ii) The no effort equilibrium is unique for high enough c.
(iii) If $\gamma(P-1)-\varepsilon \leq c \leq \gamma(P-1)$, ε small enough, only the full effort equilibrium

[9] $\lceil x\rceil$ denotes the smallest integer $\geq x$. Note that if $n \bmod (P-1) \neq 0$, then at least one low level player has to process $\left\lceil\frac{n}{P-1}\right\rceil$ items.

and the no effort equilibrium exist.

(iv) An l-effort equilibrium coexists with at least $\sum_{i=l}^{P-1} \binom{P-1}{i}$ other equilibria: the $\binom{P-1}{l} - 1$ other l-effort equilibria, the $(l+i)_{i \in \{1, \dots P-1-l\}}$-effort equilibria and the no effort equilibrium.

Proof (i) Follows from Propositions 3 and 4. Proposition 3 guarantees that there are no equilibria in which some managers process only a fraction of the set of data assigned to them. Proposition 4 eliminates putative equilibria in which the top manager provides effort when reading the file of a subordinate but the subordinate does not provide effort.

(ii) Obvious.

(iii) Follows from the diminishing increments of $k(n)$.

(iv) Follows directly from Propositions 4 and 5.

Q.E.D.

Moreover, we can rank some equilibria according to the Pareto-criterion:

Proposition 7 *Consider a centralized tree. The full effort equilibrium Pareto dominates all other equilibria whenever it exists.*

Proof Consider any agent i. First note that the full effort equilibrium payoff for agent i is larger than or equal to the payoff when agent i never provides effort while agents $-i$ always do. Next note that this payoff is more than player i's payoff in any equilibrium where less players provide effort. Q.E.D.

Hence, we expect the full effort equilibrium to be the most obvious way to play the game whenever this equilibrium exists. Otherwise we expect players to play the no effort equilibrium.

5.3 Efficiency in a centralized tree

We have so far seen that a centralized tree may become more efficient if the number of subordinates increases because incentives are related to the number of objects processed. However, there is a countervailing effect due to the increase of operations that have to be performed at the upper level. We have

Proposition 8 *It does not pay to increase the number of information processing agents beyond $\tilde{P}(n)$ in a centralized tree, where $\tilde{P}(n)$ is implicitly defined by*

$$\tilde{P} = n \left(\prod_{k=\left\lfloor \frac{n}{2(\tilde{P}-1)}+1 \right\rfloor}^{\left\lfloor \frac{n}{2} \right\rfloor} \frac{2k-1}{2k} \right) + 1. \tag{6}$$

This would (i) increase the number of operations (ii) without reducing delay (iii) nor strengthening incentives.

From Corollary 3, we know that a full effort equilibrium exists if and only if $c \leq \gamma(P-1) = \min\left\{c\left(\left\lceil \frac{n}{P-1} \right\rceil\right), \frac{1}{P-1}k(n)\right\}$. The widest range of cost parameters is obtained with $\tilde{P}$ managers, when both values of the above expression are the same: $\gamma(\tilde{P}-1) = c\left(\left\lceil \frac{n}{\tilde{P}-1} \right\rceil\right) = \frac{1}{\tilde{P}-1}k(n)$. Note that as n grows by some factor α, $\tilde{P}$ increases by a factor $< \alpha$. Therefore, $\frac{n}{\tilde{P}-1}$ increases in n and $c\left(\frac{n}{\tilde{P}-1}\right) = \gamma(\tilde{P}-1)$ decreases in n. Thus, for any cost of effort c, there exists an upper bound on n beyond which it is not possible to process the data set with a two-level hierarchy. This observation suggests that hierarchies with more layers may be useful in some cases.

5.4 Reduced tree equilibria

We now turn to the Nash Equilibria of the reduced tree. We restrict the set of parameters for this class of hierarchies in the following way:

Assumption 1: $P = 2^a, a \in \mathbb{N}\backslash\{0\}$.

Assumption 2: $n = bP, b \in \mathbb{N}\backslash\{0,1\}$.

We impose these assumptions in order to avoid case distinguishes that blow up the proofs but do not add any insights to the analysis. Note that the parameter set is still unbounded. In contrast to the case of a centralized tree, the reduced tree does not always (i.e. for all cost parameters c) have a no effort equilibrium. The reason is that each manager picks up some objects himself. Therefore, for sufficiently low costs c, no effort cannot constitute an equilibrium.

Proposition 9 *Consider a reduced tree. There is a lower bound on c below which no equilibrium exists where no player provides effort.*

PROOF Suppose that all players do not provide effort. If $c < \frac{P}{n}\left(k(n) - k\left(n - \frac{n}{P}\right)\right)$ there exists a profitable deviation from the supposed equilibrium strategy for the top player. He will choose to provide effort reading the objects that are directly assigned to him. Q.E.D.

Call manager i's immediate superior $s(i)$ and let $s^n(i)$ be i's n^{th} superior. Call the top manager T and let $s(T) = T$. Let $S_i := \{i \cup k : k = s^n(i), n \leq T\}$ be the ordered set of mangers who are working on the calculation path starting with agent i.

Lemma 4 *(i) Manager i provides effort only if all managers $k \in S_i$ provide effort when reading the object containing i's information.*
(ii) A manager never provides effort when reading the information provided by a direct subordinate if this subordinate never provided effort.

PROOF (i) Providing effort if one of his superiors does not read the information does not change the result but involves costs for i.

(ii) Obvious. Q.E.D.

Proposition 4 describes a class of coordination failure equilibria similar to the l-effort equilibria in the centralized tree described in Proposition 4. In these equilibria, subordinates do not work because their superiors do not and superiors do not work because their subordinates do not.

Proposition 10 *(i) Consider a reduced tree with 2^a managers processing $b2^a$ data items, with $b \neq 3$. A full effort equilibrium exists if and only if*

$$c \leq c(b).$$

(ii) Consider a reduced tree. The full effort equilibrium Pareto dominates all other equilibria whenever it exists.

PROOF (i) See Appendix A.

(ii) The proof of Proposition 7 applies to the reduced tree as well. Q.E.D.

Thus, in order to induce all players in the reduced tree with 2^a managers to provide effort, one has to take care of the incentives of lowest level players only. An immediate consequence is that:

Corollary 5 *The parameter range for which full effort equilibria are attainable can be increased by increasing the number of managers processing information in the reduced tree, until $\frac{n}{P} = 2$.*

Hence, by decentralizing information processing tasks in the way proposed by Radner (1993), proper incentives can be provided by involving enough agents. Note that if it is not possible to provide proper incentives to the members of a reduced tree, then there exists no hierarchy design with which this is possible. This is due to the fact that in the reduced tree it suffices to provide incentives to the lowest level managers, which is a necessary condition for effort provision in other hierarchies as well. Since the number of lowest level managers is maximal in the reduced tree, it is not possible to provide them with better incentives.

Note that the proposition only holds if managers process a number of initial data items which is not equal to 3. For the rest of the paper we consider the case where $n/P \neq 3$.

5.5 2^T-tree equilibria

We restrict attention to the full effort equilibrium in the 2^T-Tree. From the reasoning above it should be clear, that the full effort equilibrium always exists if other equilibria than the no effort equilibrium exist. If it exists, it Pareto dominates all other equilibria.

Proposition 11 *Consider a 2^T-Tree with $2^a - 1$ managers, $a \in \mathbb{N}\backslash\{0,1\}$, processing a set of $b2^a, b \in \mathbb{N}\backslash\{0\}$ data items. The full effort equilibrium exists if and only if*

$$c \leq c(2b).$$

Proof There are 2^{a-1} lowest level managers who process $2b$ data items each. Given that all other agents provide effort, a lowest level manager provides effort if and only if $c \leq c(2b)$.

Consider the incentives of any other manager in the hierarchy: He removes more uncertainty than a lowest level manager and has to provide effort (weakly) less often. Thus, he has an incentive to provide effort if a lowest level manager has. Q.E.D.

6 Comparison of the systems

We find that the reduced tree ensures the existence of a full information equilibrium for a larger parameter range than the balanced hierarchies working with the same number of managers. This in turn means that for each balanced tree and for some cost parameters c there exists a reduced tree with an equal number of agents that aggregates information faster, better and at lower cost.

Proposition 12 *Consider a centralized tree and a reduced tree with* 2^a *managers processing* $n = b2^a$ *data items, with* $a \in \mathbb{N}\backslash\{0\}$, $b \in \mathbb{N}\backslash\{0,3\}$. *For*
(i) $c < \gamma(P-1)$ *both trees have a full effort equilibrium. The reduced tree works faster.*
(ii) $\gamma(P-1) < c < c\left(\frac{n}{P}\right)$ *the reduced tree has a full effort equilibrium while the centralized tree has no such equilibrium. The reduced tree works faster.*
(iii) $c > c\left(\frac{n}{P}\right)$ *the no effort equilibrium is unique in both, the centralized tree and the reduced tree. The reduced tree works faster.*
(iv) $c < \frac{P}{n}\left(k(n) - k\left(n - \frac{n}{P}\right)\right)$ *the no effort equilibrium fails to exist in the reduced tree while it exists for all parameter values in the centralized tree.*

Proof Follows from Corollary 3, Propositions 9 and 10. Remember that $\gamma(P-1) \leq c\left(\frac{n}{P-1}\right) < c\left(\frac{n}{P}\right)$. Q.E.D.

A similar result holds for the comparison of a reduced tree and the 2^T tree.

Proposition 13 *Consider a reduced tree with* 2^a *managers and* 2^T*-trees with* $2^a - 1$, *and* $2^{a+1} - 1$ *managers respectively, processing* $n = b2^a$ *data items. For*
(i) $c < c(2b)$ *all three trees have a full effort equilibrium. The reduced tree works fastest.*
(ii) $c(2b) < c < c(b)$ *in the* 2^T*-tree with* $2^a - 1$ *managers the no effort equilibrium is unique. The reduced tree and the* 2^T*-tree with* $2^{a+1} - 1$ *managers have full effort equilibria. The reduced tree works faster and uses less resources.*
(iii) $c < \frac{P}{n}\left(k(n) - k(n-b)\right)$ *the no effort equilibrium fails to exist in the reduced tree while it exists for all parameter values in the* 2^T*-trees.*

Proof Follows from Propositions 9, 10 and 11. Q.E.D.

7 Efficient hierarchy design

Our main result is that any programmed network $\tilde{N}$ that has a full effort equilibrium can be replaced by a reduced tree $R(\tilde{N})$ with the same number of processors that also has a full effort equilibrium. Therefore $R(\tilde{N})$ generates the same classical surplus and a faster decision. We are able to proof this result for a subclass of all programmed networks. These are hierarchies which can be transformed into a reduced tree which is complete in the following sense:

Definition 3 *A reduced tree is called complete if* $P = 2^a$*, and* $n = b \cdot P$*, where* $a, b \in \mathbb{N}\backslash\{0\}$*.*

Definition 4 *A network* $\tilde{N}$ *which processes* n *objects and has* P *processors with* $P = 2^a$*, and* $n = b \cdot P$*, where* $a, b \in \mathbb{N}\backslash\{0\}$ *is called completely transferable.*

Our result concerns hierarchies that are completely transferable. The proof requires the following Lemma:

Lemma 6 *Consider a single manager, who subsequently adds two objects,* 1*, and* 2*. Providing effort on each object costs* c*. Assume that object* 1 *contains less information than object* 2*. If it is optimal to provide full effort for this manager then it is optimal to provide full effort after the permutation of both tasks.*

PROOF This follows immediately from the shape of the cost function $c(n)$. The manager wants to provide effort on object 1 - which now comes second - because the marginal gain from doing so has increased. He wants to provide effort on object 2 because it costs as much as the processing of object 1 in the initial situation but contains more information. Q.E.D.

Proposition 14 *Consider any given programmed network $\tilde{N}$ that is completely transferable into a reduced tree $R(\tilde{N})$ with 2^a managers, $n = b2^a$ data items, with $a \in \mathbb{N}\backslash\{0\}$, $b \in \mathbb{N}\backslash\{0,3\}$. If $\tilde{N}$ has a full effort equilibrium then $R(\tilde{N})$ has a full effort equilibrium.*

PROOF Modify the initial network in the following way: let each superior wait until the information by all subordinates has been provided and let him process the partial results before he reads his initial information. Call this new programmed network $\tilde{N}'$. Applying the above Lemma repeatedly yields that if network $\tilde{N}$ has a full effort equilibrium then $\tilde{N}'$ has a full effort equilibrium as well.

Next, note that each manager is willing to provide effort on his initial objects given that all other objects have been processed. This follows from the structure of the network $\tilde{N}'$ and from the fact that $\tilde{N}'$ has a full effort equilibrium.

In network $\tilde{N}'$ there has to be one manager who processes b initial objects or more. Therefore processing all b objects pays for a manager who only processes his b initial objects in a reduced tree. This in turn is a sufficient condition for the existence of a full effort equilibrium in $R(\tilde{N})$. Q.E.D.

This result has two immediate consequences for the efficient design of programmed networks.

Corollary 7 *Consider any given programmed network that can be transferred completely into a reduced tree with 2^a managers, $n = b2^a$ data items, with $a \in \mathbb{N}\backslash\{0\}$, $b \in \mathbb{N}\backslash\{0,3\}$. If $\tilde{N}$ has a full effort equilibrium then the reduced tree $R(\tilde{N})$ generates the same classical surplus and is at least a fast.*

Note also that, as a consequence, all efficient programmed networks are either (i) reduced trees or (ii) not completely transferable or (iii) efficiency is not compatible with full effort provision.

8 Conclusion

Our analysis merges two recent strands of the literature on organization design: game theory and the theory of the design of programmed networks. We find that there is an elementary advantage of decentralized structures: they provide better incentives for self-interested managers. Given the underlying stochastic structure we find that the division of tasks favors equilibria where all players provide effort because the marginal return from smaller tasks is larger. The reason lies in the complementarity of effort of different managers which arises endogenously in our model.

Our second main result is that speed and quality may not be conflicting objectives when it comes to the evaluation of hierarchies. We find that reduced trees à la Radner (1993) outperform other arrangements in terms of both dimensions.

Some of our main results are robust for all underlying stochastic structures that guarantee the (i) monotonicity of $k(n)$ and $c(n)$ and (ii) the diminishing increments of $k(n)$ and (iii) the diminishing decrements of $c(n)$. This holds for the advantage of hierarchies vis-à-vis the single decision maker and for the advantage of some reduced trees vis-à-vis a balanced structure. One interesting extension would be to study the game induced by a programmed network when one of these conditions does not hold, another would be to analyze when one may expect such a structure.

There are several further useful extensions of the present work. One such extension is to study different forms of conflict of interest than the one described in the present paper. In the present paper the only conflict of interest among agents concerned the individual disutility from providing effort. In many interesting applications however, there is also some disagreement about the best decision even if there is full information. A second approach would be to change the verifiability structure. If some data is verifiable ex post then monetary incentives may be used and efficient hierarchies might look different. A third alternative approach would link the cost of information aggregation to the complexity of the task. All three extensions may prove useful in extending the range of applications of this framework.

9 Appendix

In Appendix A, we provide the proofs omitted in the text. Appendix B collects some rules concerning binomials used within the paper. We derive some rules explicitly that cannot be found in standard formularies.

9.1 A. Proofs

Proof of Proposition 1

(i) First consider the case where n is an even number. We have:

$$k(n) = 2\sum_{i=\frac{n}{2}}^{n} p^i \left(i - \frac{n}{2}\right), \tag{7}$$

The $(n+1)^{\text{th}}$ number can either be 0 or 1. The cost of guessing $(n+1)$ numbers therefore is:

$$\begin{aligned} k(n+1) &= \tfrac{1}{2}\left(2\textstyle\sum_{i=\frac{n}{2}+1}^{n} p^i \left(i - \frac{n+1}{2} + \frac{1}{2}\right) + \frac{p_{\frac{n}{2}}}{2}\right) \\ &\quad + \tfrac{1}{2}\left(2\textstyle\sum_{i=\frac{n}{2}+1}^{n} p^i \left(i - \frac{n-1}{2} - \frac{1}{2}\right) + \frac{p_{\frac{n}{2}}}{2}\right) \\ &= 2\textstyle\sum_{i=\frac{n}{2}}^{n} p^i \left(i - \frac{n}{2}\right) + \frac{p_{\frac{n}{2}}}{2}. \end{aligned}$$

So we have

$$k(n+1) - k(n) = \tfrac{p_{\frac{n}{2}}}{2} > 0, \tag{8}$$

for n even. Next consider the case where n is an odd number. We have:

$$k(n) = 2\sum_{i=\frac{n}{2}+\frac{1}{2}}^{n} p^i \left(i - \frac{n}{2}\right), \tag{9}$$

and

$$\begin{aligned}
k(n+1) &= \tfrac{1}{2}\left(2\sum_{i=\frac{n}{2}+\frac{1}{2}}^{n} p_i\left(i-\tfrac{n+1}{2}+\tfrac{1}{2}\right)\right) \\
&\quad +\tfrac{1}{2}\left(2\sum_{i=\frac{n}{2}+\frac{1}{2}}^{n} p_i\left(i-\tfrac{n-1}{2}-\tfrac{1}{2}\right)\right) \\
&= \sum_{i=\frac{n}{2}+\frac{1}{2}}^{n} p_i\left(i-\tfrac{n+1}{2}+\tfrac{1}{2}\right)+\sum_{i=\frac{n}{2}+\frac{1}{2}}^{n} p_i\left(i-\tfrac{n-1}{2}-\tfrac{1}{2}\right) \\
&= \sum_{i=\frac{n}{2}+\frac{1}{2}}^{n} p_i\left(i-\tfrac{n}{2}\right)+\sum_{i=\frac{n}{2}+\frac{1}{2}}^{n} p_i\left(i-\tfrac{n}{2}\right) \\
&= \sum_{i=\frac{n}{2}+\frac{1}{2}}^{n} p_i\left(i-\tfrac{n}{2}\right)+\sum_{i=\frac{n}{2}+\frac{1}{2}}^{n} p_i\left(i-\tfrac{n}{2}\right)=k(n).
\end{aligned}$$

(ii)

$$k(n+1)=\begin{cases} k(n)+\frac{p_{\frac{n}{2}}}{2},\ n \text{ even} \\ k(n),\ n \text{ odd}\end{cases} \tag{10}$$

$$\Rightarrow k(n+2)=\begin{cases} k(n)+\frac{p_{\frac{n}{2}}}{2},\ n \text{ even} \\ k(n)+\frac{p_{\lceil\frac{n}{2}\rceil}}{2},\ n \text{ odd}\end{cases} \tag{11}$$

$$\Rightarrow k(x)=k(1)+\sum_{i=1}^{\lceil\frac{x}{2}\rceil-1}\frac{p_i}{2}=\frac{1}{2}+\sum_{i=1}^{\lceil\frac{x}{2}\rceil-1}\frac{p_i}{2}. \tag{12}$$

(iii) To show:[10]

$$\begin{aligned}
\sum_{i=1}^{\lceil\frac{x}{2}\rceil-1}\frac{p_i}{2} &= \tfrac{1}{4}\sum_{i=1}^{\lceil\frac{n}{2}\rceil-1}\left(\prod_{j=0}^{i-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right) \\
\Leftrightarrow\ p_x &= \tfrac{1}{2}\left(\prod_{j=0}^{x-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right).
\end{aligned} \tag{13}$$

$$\begin{aligned}
p_x &= \tfrac{1}{2^{2x}}\binom{2x}{x} \\
p_{x+1} &= \tfrac{1}{2^{2(x+1)}}\binom{2(x+1)}{x+1} \\
&\overset{C}{=} \tfrac{1}{2^{2(x+1)}}2\left(\binom{2x}{x}+\sum_{j=x}^{2x-1}\binom{j}{x}\right) \\
p_{x+1} &= \tfrac{1}{2}\tfrac{1}{2^{2x}}\left(\binom{2x}{x}+\sum_{j=x}^{2x-1}\binom{j}{x}\right)
\end{aligned}$$

---

[10]The rules used for transformation can be found in Appendix B.

$$
\begin{aligned}
p_{x+1} &= y p_x \Leftrightarrow \\
y &= \tfrac{p_{x+1}}{p_x} \Leftrightarrow \\
&= \frac{\frac{1}{2}\frac{1}{2^{2x}}\left(\binom{2x}{x}+\sum_{j=x}^{2x-1}\binom{j}{x}\right)}{\frac{1}{2^{2x}}\binom{2x}{x}} \\
&= \frac{\frac{1}{2}\left(\binom{2x}{x}+\sum_{j=x}^{2x-1}\binom{j}{x}\right)}{\binom{2x}{x}} \\
&\overset{X}{=} \frac{\frac{1}{2}\left(\binom{2x}{x}+\frac{x}{1+x}\binom{2x}{x}\right)}{\binom{2x}{x}} \\
&= \frac{2x+1}{2x+2}
\end{aligned}
$$

Thus we have

$$p_{x+1} = \tfrac{2x+1}{2x+2} p_x$$

and

$$
\begin{aligned}
p_{x+y} &= \tfrac{2(x+y-1)+1}{2(x+y-1)+2} p_{x+y-1} \\
&= \tfrac{2(x+y-1)+1}{2(x+y-1)+2}\tfrac{2(x+y-2)+1}{2(x+y-2)+2} p_{x+y-2} \\
&= \tfrac{2(x+y-1)+1}{2(x+y-1)+2}\tfrac{2(x+y-2)+1}{2(x+y-2)+2}\cdots\tfrac{2x+1}{2x+2} p_x \\
&= \left(\prod_{j=0}^{y-1}\tfrac{2(x+j)+1}{2(x+j)+2}\right) p_x,
\end{aligned}
$$

which implies that

$$
\begin{aligned}
p_x &= \left(\prod_{j=0}^{x-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right) p_1 \qquad (14) \\
&= \tfrac{1}{2}\left(\prod_{j=0}^{x-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right).
\end{aligned}
$$

Q.E.D.

PROOF OF PROPOSITION 2

Consider the case with n even.

$$
\begin{aligned}
c(n) &= c(n+1) \Leftrightarrow \\
\tfrac{k(n)}{n} &= \tfrac{k(n+1)}{n+1} \Leftrightarrow
\end{aligned}
$$

$$\begin{aligned}
(n+1)\,k(n) &= nk(n+1) \Leftrightarrow \\
(n+1)\,k(n) &= n\left(k\left(n\right)+\tfrac{p_{\frac{n}{2}}}{2}\right) \Leftrightarrow \\
k\left(n\right) &= \frac{np_{\frac{n}{2}}}{2}.
\end{aligned}$$

Suppose that the above equality holds for some even n and note that it holds for $n = 2$. Consider the next even number.

$$\begin{aligned}
k\left(n+2\right) &= \tfrac{(n+2)p_{\frac{n}{2}+1}}{2} \Leftrightarrow \\
k\left(n\right)+\frac{p_{\frac{n}{2}}}{2} &= \tfrac{(n+2)p_{\frac{n}{2}+1}}{2} \Leftrightarrow \\
\frac{p_{\frac{n}{2}}}{2} &= \tfrac{(n+2)p_{\frac{n}{2}+1}}{2} - n\tfrac{p_{\frac{n}{2}}}{2} \Leftrightarrow \\
1 &= \tfrac{(n+2)p_{\frac{n}{2}+1}}{p_{\frac{n}{2}}} - n \Leftrightarrow \\
1 &= \tfrac{2^n}{2^{n+2}}\tfrac{(n+2)\binom{n+2}{\frac{n}{2}+1}}{\binom{n}{\frac{n}{2}}} - n \Leftrightarrow \\
1 &= \tfrac{1}{4}\tfrac{(n+2)(n+2)!}{\left(\left(\frac{n}{2}+1\right)!\right)^2}\tfrac{\left(\frac{n}{2}!\right)^2}{(n!)} - n \Leftrightarrow \\
4\left(1+n\right) &= \tfrac{(n+1)(n+2)^2}{\left(\frac{n}{2}+1\right)^2} \Leftrightarrow \\
4\left(\tfrac{n}{2}+1\right)^2 &= \left(n+2\right)^2 \Leftrightarrow \\
2\left(\tfrac{n}{2}+1\right) &= \left(n+2\right).
\end{aligned}$$

Next, consider the case with n odd.

$$c(n+1) = \tfrac{k(n+1)}{n+1} = \tfrac{k(n)}{n+1} = \tfrac{nc(n)}{n+1}.$$

Q.E.D.

Proof of Proposition 8

(i) Obvious. (ii) The delay is minimized at $\hat{P} = \sqrt{n} + 1$, in which case every agent processes the same number of objects. But in this situation the top-manager has better incentives than his subordinates, because his objects contain more information. Thus it must be that $\tilde{P} > \hat{P}$.

(iii) The best incentives are provided if $\gamma(P-1)$ is maximized. By increasing the number of low level processors, incentives of low level players and the top player are driven in opposite directions. The value for P that maximizes $\gamma(P-1)$ given n is obtained by

$$\begin{aligned}
c\left(\left\lceil \tfrac{n}{P-1} \right\rceil\right) &= \tfrac{1}{P-1} k(n) \Leftrightarrow \\
c\left(\left\lceil \tfrac{n}{P-1} \right\rceil\right) &= \tfrac{n}{P-1} c(n) \Leftrightarrow \\
\frac{c\left(\left\lceil \frac{n}{P-1} \right\rceil\right)}{c(n)} &= \tfrac{n}{P-1} \Leftrightarrow \\
\prod_{k=\left\lfloor \frac{n}{2(P-1)}+1 \right\rfloor}^{\left\lfloor \frac{n}{2} \right\rfloor} \frac{2k}{2k-1} &= \tfrac{n}{P-1} \Leftrightarrow \\
P &= n \left(\prod_{k=\left\lfloor \frac{n}{2(P-1)}+1 \right\rfloor}^{\left\lfloor \frac{n}{2} \right\rfloor} \frac{2k-1}{2k} \right) + 1.
\end{aligned}$$

Q.E.D.

Proof of Proposition 10

The proof of Proposition 10 consists of two parts, the "IF"-part and the "ONLY IF"-part.

"IF"

Consider the top player. Assume all others provide effort. His best response will include to provide effort when reading the object provided by the last subordinate no matter what he did before if

$$c \leq k(n) - k\left(\tfrac{n}{2}\right) = k\left(2^a b\right) - k\left(2^{a-1} b\right). \tag{15}$$

The rest of this player's strategic situation is equivalent to the one of the top player's last subordinate. Therefore, his incentives to provide effort are the same and he will provide effort in equilibrium if his subordinate does and the above condition holds. Now, consider the top player's last subordinate. Assume again that all other players provide effort. The same reasoning as above applies: This player will provide effort when reading the information provided by his last subordinate no matter what he did before if

$$c \leq k\left(\tfrac{n}{2}\right) - k\left(\tfrac{n}{4}\right) = k\left(2^{a-1}b\right) - k\left(2^{a-2}b\right). \tag{16}$$

The remaining strategic situation is than again equivalent to the one of his last subordinate.

We conclude: All players provide full effort if

$$c \leq \min\left\{c(b), \min_j \left\{k(2^j b) - k(2^{j-1}b)\right\}_{j=1\ldots a}\right\}. \tag{17}$$

The remainder of the proof is to show (i) that the thresholds are nondecreasing in j, which implies that the sufficient condition for effort provision reduces to

$$c \leq \min\left\{c(b), k(2b) - k(b)\right\} \tag{18}$$

and then to show (ii) that

$$c(b) \leq k(2b) - k(b), \tag{19}$$

which reduces the sufficient condition to

$$c \leq c(b) \tag{20}$$

and completes the proof of the "if"-part.

(i) To show: $\forall j \in \mathbb{N}\backslash\{0\}, \forall b \in \mathbb{N}\backslash\{0,1\}$:

$$k(2^{j+1}b) - k(2^j b) > k(2^j b) - k(2^{j-1}b).$$

Remember from Proposition 1(iii):

$$k(n) = \tfrac{1}{2}\left(1 + \tfrac{1}{2}\sum_{i=1}^{\lceil \frac{n}{2} \rceil - 1}\left(\prod_{j=0}^{i-2} \tfrac{2(1+j)+1}{2(1+j)+2}\right)\right).$$

That is

$$k(2^{j+1}b) = \tfrac{1}{2}\left(1 + \tfrac{1}{2}\sum_{i=1}^{2^j b - 1}\left(\prod_{j=0}^{i-2} \tfrac{2(1+j)+1}{2(1+j)+2}\right)\right)$$

and

$$
\begin{aligned}
& k(2^{j+1}b) - k(2^j b) \\
= \; & \frac{1}{2}\left(1+\frac{1}{2}\sum_{i=1}^{2^j b-1}\left(\prod_{j=0}^{i-2}\frac{2(1+j)+1}{2(1+j)+2}\right)\right) - \frac{1}{2}\left(1+\frac{1}{2}\sum_{i=1}^{2^{j-1}b-1}\left(\prod_{j=0}^{i-2}\frac{2(1+j)+1}{2(1+j)+2}\right)\right) \\
= \; & \frac{1}{4}\sum_{i=1}^{2^j b-1}\left(\prod_{k=0}^{i-2}\frac{2(1+k)+1}{2(1+k)+2}\right) - \frac{1}{4}\sum_{i=1}^{2^{j-1}b-1}\left(\prod_{k=0}^{i-2}\frac{2(1+k)+1}{2(1+k)+2}\right) \\
= \; & \frac{1}{4}\sum_{i=2^{j-1}b}^{2^j b-1}\left(\prod_{k=0}^{i-2}\frac{2(1+k)+1}{2(1+k)+2}\right).
\end{aligned}
$$

$$k(2^{j+1}b) - k(2^j b) > k(2^j b) - k(2^{j-1}b)$$

$$
\begin{aligned}
\Leftrightarrow \; & \frac{1}{4}\sum_{i=2^{j-1}b}^{2^j b-1}\left(\prod_{k=0}^{i-2}\frac{2(1+k)+1}{2(1+k)+2}\right) > \frac{1}{4}\sum_{i=\lceil 2^{j-2}b\rceil}^{2^{j-1}b-1}\left(\prod_{k=0}^{i-2}\frac{2(1+k)+1}{2(1+k)+2}\right) \\
\Leftrightarrow \; & \frac{\sum_{i=2^{j-1}b}^{2^j b-1}\left(\prod_{k=0}^{i-2}\frac{2(1+k)+1}{2(1+k)+2}\right)}{\sum_{i=\lceil 2^{j-2}b\rceil}^{2^{j-1}b-1}\left(\prod_{k=0}^{i-2}\frac{2(1+k)+1}{2(1+k)+2}\right)} > 1 \\
\Leftrightarrow \; & \frac{\left(\prod_{k=0}^{\lceil 2^{j-2}b\rceil-2}\frac{2(1+k)+1}{2(1+k)+2}\right)\left(\prod_{k=\lceil 2^{j-2}b\rceil-1}^{2^{j-1}b-2}\frac{2(1+k)+1}{2(1+k)+2}\right)\sum_{i=2^{j-1}b}^{2^j b-1}\left(\prod_{k=2^{j-1}b-1}^{i-2}\frac{2(1+k)+1}{2(1+k)+2}\right)}{\left(\prod_{k=0}^{\lceil 2^{j-2}b\rceil-2}\frac{2(1+k)+1}{2(1+k)+2}\right)\sum_{i=\lceil 2^{j-2}b\rceil}^{2^{j-1}b-1}\left(\prod_{k=\lceil 2^{j-2}b\rceil-1}^{i-2}\frac{2(1+k)+1}{2(1+k)+2}\right)} > 1 \\
\Leftrightarrow \; & \frac{\sum_{i=2^{j-1}b}^{2^j b-1}\left(\prod_{k=2^{j-1}b-1}^{i-2}\frac{k+\frac{3}{2}}{k+2}\right)}{\left(\prod_{k=\lceil 2^{j-2}b\rceil-1}^{2^{j-1}b-2}\frac{k+2}{k+\frac{3}{2}}\right)\sum_{i=\lceil 2^{j-2}b\rceil}^{2^{j-1}b-1}\left(\prod_{k=\lceil 2^{j-2}b\rceil-1}^{i-2}\frac{k+\frac{3}{2}}{k+2}\right)} > 1
\end{aligned}
$$

Note that:

(i)

$$\sum_{i=2^{j-1}b}^{2^j b-1}\left(\prod_{k=2^{j-1}b-1}^{i-2}\frac{k+\frac{3}{2}}{k+2}\right) > 2^{j-1}b\left(\frac{2^{j-1}b+\frac{1}{2}}{2^{j-1}b+1}\right)^{2^{j-1}b-2} \tag{21}$$

$2^{j-1}b$ summands, smallest summand $\prod_{k=2^{j-1}b-1}^{2^j b-3}\frac{k+\frac{3}{2}}{k+2}$, $2^{j-1}b-2$ factors smallest of which $\frac{2^{j-1}b+\frac{1}{2}}{2^{j-1}b+1}$,

(ii)

$$\sum_{i=\lceil 2^{j-2}b\rceil}^{2^{j-1}b-1}\left(\left(\prod_{k=\lceil 2^{j-2}b\rceil-1}^{i-2}\frac{k+\frac{3}{2}}{k+2}\right)\left(\prod_{k=\lceil 2^{j-2}b\rceil-1}^{2^{j-1}b-2}\frac{k+2}{k+\frac{3}{2}}\right)\right) < \lfloor 2^{j-2}b\rfloor\left(\frac{2^{j-1}b}{2^{j-1}b-\frac{1}{2}}\right)^{\lceil 2^{j-2}b\rceil-1}$$

$\lfloor 2^{j-2}b\rfloor$ summands highest of which is smaller than $\left(\frac{2^{j-1}b}{2^{j-1}b-\frac{1}{2}}\right)^{\lceil 2^{j-2}b\rceil-1}$

Therefore:

$$\frac{\sum_{i=2^{j-1}b}^{2^j b-1}\left(\prod_{k=2^{j-2}b-1}^{i-2}\frac{k+\frac{3}{2}}{k+2}\right)}{\left(\prod_{k=\lceil 2^{j-2}b\rceil-1}^{2^{j-1}b-2}\frac{k+2}{k+\frac{3}{2}}\right)\sum_{i=\lceil 2^{j-2}b\rceil}^{2^{j-1}b-1}\left(\prod_{k=\lceil 2^{j-2}b\rceil-1}^{i-2}\frac{k+\frac{3}{2}}{k+2}\right)}$$

$$> \frac{2^{j-1}b\left(\frac{2^{j-1}b+\frac{1}{2}}{2^{j-1}b+1}\right)^{2^{j-1}b-2}}{\lfloor 2^{j-2}b\rfloor\left(\frac{2^{j-1}b}{2^{j-1}b-\frac{1}{2}}\right)^{\lceil 2^{j-2}b\rceil-1}}$$

$$= \frac{2^{j-1}b}{\lfloor 2^{j-2}b\rfloor}\frac{\left(\frac{2^{j-1}b+\frac{1}{2}}{2^{j-1}b+1}\right)^{\lceil 2^{j-2}b\rceil-1}\left(\frac{2^{j-1}b+\frac{1}{2}}{2^{j-1}b+1}\right)^{2^{j-1}b-2-\left(\lceil 2^{j-2}b\rceil-1\right)}}{\left(\frac{2^{j-1}b}{2^{j-1}b-\frac{1}{2}}\right)^{\lceil 2^{j-2}b\rceil-1}}$$

$$= \frac{2^{j-1}b}{\lfloor 2^{j-2}b\rfloor}\frac{\left(\frac{2^{j-1}b+\frac{1}{2}}{2^{j-1}b+1}\right)^{\lceil 2^{j-2}b\rceil-1}\left(\frac{2^{j-1}b+\frac{1}{2}}{2^{j-1}b+1}\right)^{\lfloor 2^{j-2}b\rfloor-1}}{\left(\frac{2^{j-1}b}{2^{j-1}b-\frac{1}{2}}\right)^{\lceil 2^{j-2}b\rceil-1}}$$

$$= \frac{2^{j-1}b}{\lfloor 2^{j-2}b\rfloor}\left(\frac{2^{j-1}b+\frac{1}{2}}{2^{j-1}b+1}\frac{2^{j-1}b-\frac{1}{2}}{2^{j-1}b}\right)^{\lceil 2^{j-2}b\rceil-1}\left(\frac{2^{j-1}b+\frac{1}{2}}{2^{j-1}b+1}\right)^{\lfloor 2^{j-2}b\rfloor-1}$$

Abbreviate $2^{j-2}b := x \geq 1$

We have to distinguish two cases:

A) $\lfloor x\rfloor = \lceil x\rceil = x$

B) $\lfloor x\rfloor = \lceil x\rceil - 1$

Case A)

$$\frac{2^{j-1}b}{\lfloor 2^{j-2}b\rfloor}\left(\frac{2^{j-1}b+\frac{1}{2}}{2^{j-1}b+1}\frac{2^{j-1}b-\frac{1}{2}}{2^{j-1}b}\right)^{\lceil 2^{j-2}b\rceil-1}\left(\frac{2^{j-1}b+\frac{1}{2}}{2^{j-1}b+1}\right)^{\lfloor 2^{j-2}b\rfloor-1} = \frac{2x}{x}\left(\frac{2x+\frac{1}{2}}{2x+1}\frac{2x-\frac{1}{2}}{2x}\right)^{x-1}\left(\frac{2x+\frac{1}{2}}{2x+1}\right)^{x-1}$$

Because the factors are increasing in x,

$$2\left(\frac{2x+\frac{1}{2}}{2x+1}\frac{2x-\frac{1}{2}}{2x}\right)^{x-1}\left(\frac{2x+\frac{1}{2}}{2x+1}\right)^{x-1} \geq 2\left(\frac{2+\frac{1}{2}}{2+1}\frac{2+\frac{1}{2}}{2+1}\frac{2-\frac{1}{2}}{2}\right)^{x-1} = 2\left(\frac{25}{48}\right)^{x-1}$$

Thus, for case A, a sufficient condition for the thresholds to be nondecreasing is:

$$\begin{aligned} 2\left(\frac{25}{48}\right)^{x-1} &> 1 \\ \ln 2 + \ln\left(\frac{25}{48}\right)(x-1) &> 0 \\ x &> 1 - \frac{\ln 2}{\ln\left(\frac{25}{48}\right)} \\ x &> 2 \end{aligned}$$

Check for $x = 1$ and $x = 2$ explicitly:

$$\begin{aligned} 2\left(\frac{2+\frac{1}{2}}{2+1}\frac{2x-\frac{1}{2}}{2x}\right)^0\left(\frac{2+\frac{1}{2}}{2+1}\right)^0 &= 2 > 1 \\ 2\left(\frac{4+\frac{1}{2}}{4+1}\frac{4-\frac{1}{2}}{4}\right)\left(\frac{4+\frac{1}{2}}{4+1}\right) &= 1.4175 \end{aligned}$$

The sufficient condition holds $\forall x$, i.e. $\forall j, b$ in Case A.

Case B)

$$\begin{aligned} & 2\left(\frac{2^{j-1}b+\frac{1}{2}}{2^{j-1}b+1}\frac{2^{j-1}b-\frac{1}{2}}{2^{j-1}b}\right)^{\lceil 2^{j-2}b\rceil -1}\left(\frac{2^{j-1}b+\frac{1}{2}}{2^{j-1}b+1}\right)^{\lfloor 2^{j-2}b\rfloor -1} \\ = & \frac{2x}{\lfloor x\rfloor}\left(\frac{2x+\frac{1}{2}}{2x+1}\frac{2x-\frac{1}{2}}{2x}\right)^{\lceil x\rceil -1}\left(\frac{2x+\frac{1}{2}}{2x+1}\right)^{\lfloor x\rfloor -1} \\ = & \frac{2x}{\lfloor x\rfloor}\left(\frac{2x+\frac{1}{2}}{2x+1}\frac{2x-\frac{1}{2}}{2x}\right)^{\lfloor x\rfloor}\left(\frac{2x+\frac{1}{2}}{2x+1}\right)^{\lfloor x\rfloor -1} \\ > & 2\left(\frac{2+\frac{1}{2}}{2+1}\frac{2-\frac{1}{2}}{2}\right)^{\lfloor x\rfloor}\left(\frac{2+\frac{1}{2}}{2+1}\right)^{\lfloor x\rfloor -1} \\ = & 2\left(\frac{2+\frac{1}{2}}{2+1}\frac{2-\frac{1}{2}}{2}\right)^{\lfloor x\rfloor}\left(\frac{2+\frac{1}{2}}{2+1}\right)^{\lfloor x\rfloor -1} \\ = & 2\left(\frac{5}{8}\right)^{\lfloor x\rfloor}\left(\frac{5}{6}\right)^{\lfloor x\rfloor -1} \end{aligned}$$

For Case B, a sufficient condition for the thresholds to be nondecreasing is:

$$\begin{aligned} 2\left(\tfrac{5}{8}\right)^{\lfloor x\rfloor}\left(\tfrac{5}{6}\right)^{\lfloor x\rfloor -1} &> 1 \\ &\Leftrightarrow \ln 2 + \lfloor x\rfloor \ln\left(\tfrac{5}{8}\right) + (\lfloor x\rfloor - 1)\ln\left(\tfrac{5}{6}\right) > 0 \\ &\Leftrightarrow \lfloor x\rfloor > \tfrac{-\ln 2 - \ln 6 + \ln 5}{(2\ln 5 - \ln 8 - \ln 6)} \\ &\Leftrightarrow \lfloor x\rfloor > 1. \end{aligned}$$

Check for $\lfloor x\rfloor = 1$ explicitly:

$$2\left(\frac{2+\frac{1}{2}}{2+1}\frac{2-\frac{1}{2}}{2}\right)^1\left(\frac{2+\frac{1}{2}}{2+1}\right)^0 = 1.25 > 1.$$

Thus, the sufficient condition holds in Case B, too.

Therefore, the thresholds are nondecreasing and the sufficient condition for a full effort equilibrium is:

$$c \leq \min\left\{c(b), k(2b) - k(b)\right\}. \tag{22}$$

(ii) To show:

$$c(b) < k(2b) - k(b) \Leftrightarrow$$

$$\begin{aligned}
\tfrac{k(b)}{b} &\leq k(2b) - k(b) \Leftrightarrow \\
k(b)\tfrac{b+1}{b} &\leq k(2b) \Leftrightarrow \\
\tfrac{1}{2}\left(1 + \tfrac{1}{2}\sum_{i=1}^{\lceil \frac{b}{2} \rceil - 1}\left(\prod_{j=0}^{i-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right)\right)\tfrac{b+1}{b} &\leq \tfrac{1}{2}\left(1 + \tfrac{1}{2}\sum_{i=1}^{b-1}\left(\prod_{j=0}^{i-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right)\right) \Leftrightarrow \\
\tfrac{2}{b} + \tfrac{b+1}{b}\sum_{i=1}^{\lceil \frac{b}{2} \rceil - 1}\left(\prod_{j=0}^{i-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right) &\leq \sum_{i=1}^{b-1}\left(\prod_{j=0}^{i-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right) \Leftrightarrow \\
\tfrac{2}{b} + \tfrac{b+1}{b}\sum_{i=1}^{\lceil \frac{b}{2} \rceil - 1}\left(\prod_{j=0}^{i-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right) &\leq \sum_{i=1}^{\lceil \frac{b}{2} \rceil - 1}\left(\prod_{j=0}^{i-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right) \\
&\quad + \sum_{i=\lceil \frac{b}{2} \rceil}^{b-1}\left(\prod_{j=0}^{i-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right) \Leftrightarrow \\
\tfrac{2}{b} + \tfrac{1}{b}\sum_{i=1}^{\lceil \frac{b}{2} \rceil - 1}\left(\prod_{j=0}^{i-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right) &\leq \sum_{i=\lceil \frac{b}{2} \rceil}^{b-1}\left(\prod_{j=0}^{i-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right) \Leftrightarrow \\
2 + \sum_{i=1}^{\lceil \frac{b}{2} \rceil - 1}\left(\prod_{j=0}^{i-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right) &\leq b\sum_{i=\lceil \frac{b}{2} \rceil}^{b-1}\left(\prod_{j=0}^{i-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right).
\end{aligned}$$

The sum on the LHS contains $\lceil \frac{b}{2} \rceil$ summands, each of which is < 1. The sum on the RHS is > 1 if it contains ≥ 2 summands. Thus:

$$\begin{aligned}
2 + \sum_{i=1}^{\lceil \frac{b}{2} \rceil - 1}\left(\prod_{j=0}^{i-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right) &\leq b\sum_{i=\lceil \frac{b}{2} \rceil}^{b-1}\left(\prod_{j=0}^{i-2}\tfrac{2(1+j)+1}{2(1+j)+2}\right) \Leftarrow \\
2 + \lceil \tfrac{b}{2} \rceil &\leq b \Leftrightarrow \\
b &\geq 4.
\end{aligned}$$

Check for $b \in \{2, 3\}$ explicitly:

Holds for $b = 2$:

$$\begin{aligned}
\tfrac{k(b)}{b} &\leq k(2b) - k(b) \Leftrightarrow \\
\tfrac{k(2)}{2} &\leq k(4) - k(2) \Leftrightarrow
\end{aligned}$$

$$\begin{aligned}\frac{k(2)}{2} &\leq k(2) + \frac{p_1}{2} - k(2) \Leftrightarrow \\ \frac{k(2)}{2} &= \frac{1}{4} \leq \frac{p_1}{2} = \frac{1}{2}\frac{1}{4}\binom{2}{1} = \frac{1}{4}.\end{aligned}$$

Does not hold for $b = 3$:

$$\begin{aligned}\frac{k(b)}{b} &\leq k(2b) - k(b) \Leftrightarrow \\ \frac{k(3)}{3} &\leq k(6) - k(3) \Leftrightarrow \\ \frac{k(2)}{2} &\leq k(5) - k(4) \Leftrightarrow \\ \frac{1}{4} &\leq k(4) + \frac{p_2}{2} - k(4) \Leftrightarrow \\ \frac{1}{4} &\leq \frac{p_2}{2} = \frac{1}{2}\frac{1}{16}\binom{4}{2} = \frac{3}{16}.\end{aligned}$$

Q.E.D.

9.2 B. Rules for binomials

We summarize some general rules to which we refer within the paper.

$$\begin{aligned}\binom{n+1}{k+1} &= \binom{n}{k} + \binom{n-1}{k} + ... + \binom{k}{k} \\ \binom{n+1}{k+1} &= \binom{n}{k} + \sum_{j=k}^{n-1} \binom{j}{k} \qquad \text{(Rule A)}\end{aligned}$$

$$\binom{n+1}{k} = \frac{n+1}{n-k+1}\binom{n}{k} \qquad \text{(Rule B)}$$

$$\binom{2x+2}{x+1} = 2\left(\binom{2x}{x} + \sum_{j=x}^{2x-1} \binom{j}{x}\right) \qquad \text{(Rule C)}$$

$$\sum_{j=1}^{k} \binom{2k-j}{k} = \frac{k}{k+1}\binom{2k}{k} \qquad \text{(Rule X)}$$

Derivation of Rule C:

Rule B implies:

$$\binom{n+2}{k+1} = \frac{n+2}{n-k+1}\binom{n+1}{k+1}$$

and because of Rule A:

$$\binom{n+2}{k+1} = \frac{n+2}{n-k+1}\left(\binom{n}{k} + \sum_{j=k}^{n-1} \binom{j}{k}\right)$$

Specifically:

$$\begin{aligned}\binom{2x+2}{x+1} &= \tfrac{2x+2}{2x-x+1}\left(\binom{2x}{x}+\textstyle\sum_{j=x}^{2x-1}\binom{j}{x}\right)\\ \binom{2x+2}{x+1} &= 2\left(\binom{2x}{x}+\textstyle\sum_{j=x}^{2x-1}\binom{j}{x}\right) \qquad \text{(Rule C)}\end{aligned}$$

Derivation of Rule X:

$$\binom{n+1}{k+1}=\binom{n}{k}+\binom{n-1}{k}+\binom{n-2}{k}+++\binom{k}{k} \qquad \text{(Rule A)}$$

$\Rightarrow$

$$\begin{aligned}\binom{n-1}{k} &= \binom{n+1}{k+1}-\binom{n}{k}-\binom{n-2}{k}---\binom{k}{k}\\ \binom{n-2}{k} &= \binom{n+1}{k+1}-\binom{n}{k}-\binom{n-1}{k}---\binom{k}{k}\\ &\dots\\ \binom{k}{k} &= \binom{n+1}{k+1}-\binom{n}{k}-\binom{n-1}{k}-\binom{n-2}{k}---\binom{k+1}{k}\end{aligned}$$

$\Rightarrow$ For $n=2k$

$$\begin{aligned}\textstyle\sum_{j=1}^{k}\binom{n-j}{k} &= k\left(\binom{n+1}{k+1}-\binom{n}{k}\right)-(k-1)\textstyle\sum_{j=1}^{k}\binom{n-j}{k}\\ &= k\textstyle\sum_{j=1}^{k}\binom{n-j}{k}-(k-1)\textstyle\sum_{j=1}^{k}\binom{n-j}{k}\\ &= \textstyle\sum_{j=1}^{k}\binom{n-j}{k}\end{aligned}$$

$$\begin{aligned}\textstyle\sum_{j=1}^{k}\binom{n-j}{k} &= k\left(\binom{n+1}{k+1}-\binom{n}{k}\right)-(k-1)\textstyle\sum_{j=1}^{k}\binom{n-j}{k}\\ k\textstyle\sum_{j=1}^{k}\binom{n-j}{k} &= k\binom{n+1}{k+1}-k\binom{n}{k}\\ \textstyle\sum_{j=1}^{k}\binom{n-j}{k} &= \binom{n+1}{k+1}-\binom{n}{k}\end{aligned}$$

$$\binom{n+1}{k+1}=\binom{n}{k+1}+\binom{n}{k} \qquad \text{(Rule D)}$$

$\Rightarrow$

$$\textstyle\sum_{j=1}^{k}\binom{n-j}{k}=\binom{n}{k+1}$$

$$\binom{n}{k+1} = \frac{n-k}{k+1}\binom{n}{k} \qquad \text{(Rule E)}$$

With $n = 2k$:

$$\binom{2k}{k+1} = \frac{k}{k+1}\binom{2k}{k}$$

$$\sum_{j=1}^{k}\binom{2k-j}{k} = \frac{k}{k+1}\binom{2k}{k} \qquad \text{(Rule X)}$$

References

[1] Aghion, Philippe and Jean Tirole (1997): Formal and Real Authority in Organizations, *Journal of Political Economy* 105(1):1-29.

[2] Jehiel, Philippe (1999) "Information Aggregation and Communication in Organizations", *Management Science*, 45:659-669.

[3] Meagher, Kieron, Hakan Orbay and Timothy Van Zandt (2001): "Hierarchy Size and Environmental Uncertainty", CEPR Discussion Paper No. 2839.

[4] Meagher, Kieron and Timothy Van Zandt (1998): "Managerial Costs for one-shot decentralized Information Processing", *Review of Economic Design* 3: 329-345.

[5] Melumad, Nahum D., Dilip Mookherjee and Stefan Reichelstein, (1995) "Hierarchical Decentralization of Incentive Contracts," RAND Journal of Economics, 26:654-672.

[6] Mookherjee, Dilip and Stefan Reischelstein (1997): Budgeting and Hierarchical Control, Journal of Accounting Research 35(2):129-155.

[7] Orbay, Hakan (2002): "Information Processing Hierarchies", *Journal of Economic Theory* 105: 370-407.

[8] Prat, Andrea (1997): "Hierarchies of Processors with Endogenous Capacity", *Journal of Economic Theory* 77:214-222.

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

[10] Radner, Roy (1993): "The Organization of Decentralized Information Processing", *Econometrica* 61(5): 1109-1146.

[11] Schulte, Elisabeth and Grüner, Hans Peter (2004): "Speed and Quality of Collective Decision Making I: Imperfect Information Processing", CEPR Discussion Paper No. 4179.

[12] Van Zandt, Timothy (1997): "The Scheduling and Organization of Periodic Associative Computation: Essential networks", *Review of Economic Design* 3:15-27.

[13] Van Zandt, Timothy (1998): "The Scheduling and Organization of Periodic Associative Computation: Efficient networks", *Review of Economic Design* 3:93-127.