Discussion Paper #1121
R. September, 1996

"Coordination and the Structure of Firms"

Stanley Reiter
Northwestern University

www.kellogg.nwu.edu/research/math

CMS-EMS
The Center for
Mathematical Studies
in Economics &
Management Sciences

Northwestern University
2001 Sheridan Road 580 Leverone Hall Evanston, IL 60208-2014 USA

Discussion Paper No. 1121

Coordination and the Structure of Firms

by

Stanley Reiter[*]

May 1995, Revised September 1996

[*] Department of Economics, Managerial Economics and Decision Sciences, and Center for Mathematical Studies in Economics and Management Science, Northwestern University, Evanston, Illinois, 60208-2014.

Telephone: 847-491-3527; Fax: 847-491-2530, email: s-reiter@nwu.edu. This research was supported by NSF Grant No. IRI-902070.

Running Title: Coordination & Structure of Firms

Professor Stanley Reiter
Northwestern University
Center for Mathematical Studies
in Economics and Management Science
2001 Sheridan Road, Rm. 371
Leverone Hall
Evanston, IL 60208-2014

Abstract

This paper presents a model in which organizational structure emerges as the solution of an optimization problem. The objective is to compute the desired decisions in a given class of economic environments, and the constraints express limitation on the abilities of agents to compute and communicate. The "variable" to be solved for, whose values are interpreted as different organizational structures, is a type of directed graph (an EADAG). Depending on its structure, a solution graph may represent organization into one or more informationally independent units. The structure of solution graphs is derived from the nature of the coordination problem as expressed by the desired decision function.

Key words: Bounded rationality, Coordination, Computational complexity, Organizational structure, Theory of the firm.

Introduction

Section 1. Theory of the firm

All economic activity is ultimately economic activity of individuals; individuals are the atoms of economic theory. However, in a developed economy, a large part of that activity takes place in multiperson organizations. Production of goods and services is usually carried out by firms, which have been created to act as economic agents, and which even have in some respects the legal status of persons. Holmstrom and Tirole observe that "the volume of trade within firms is probably of the same order as market trade. Large firms are substantial subeconomies of their own with thousands of participants." (Holmstrom and Tirole [5], p. 63). Nevertheless, in economic theory a firm is typically treated as an individual decision maker. If, as Frank Knight thought, the focus of economic science is to understand how a society organizes its economic activity, a theory of the firm should address the firm's internal organization. Holmstrom and Tirole begin their chapter with the comment, "The theory of the firm has long posed a problem for economists. While substantial progress has been made on the description and analysis of market performance, firm behavior and organization have remained poorly understood." (Holmstrom and Tirole [5], p. 63).

A theory of the firm should determine the set of firms that exist, and their organizations and behaviors. One approach to this task derives from Coase's idea that economic institutions, including firms, exist to facilitate exchange and can be understood as optimal adaptations to contractual constraints. The transaction costs approach to the theory of the firm follows Coase. (Williamson, O. [25], [26]). According to that approach, a firm is a contract between parties (individuals) designed to minimize transaction costs between specialized factors of production.

> "A prime source of transaction costs is information. For technological reasons it pays to have people become specialized as specialization vastly expands the production potential. But along with specialization comes the problem of coordinating the actions of a differentially informed set of experts. This is costly for two reasons. Processing information takes time and effort even when parties share organizational goals. More typically, individuals have differing objectives and informational expertise may permit them to pursue their own objectives to the detriment of the organization as a whole." (Holmstrom and Tirole [5], p. 64).

The foregoing observations suggest what a formal theory of the firm should (perhaps ideally) look like. The formal model should contain:

I. a set of environments incorporating the relevant constraints on the activity to be coordinated, such as the set of individuals, preferences,technology and resource endowments;

II. a set of entities that can be interpreted as different organizational arrangements and behaviors, i.e., possible firms;

III. a collection of constraints that apply to individuals or the entities in II; the constraints should include those related to information processing, and those related to incentive compatibility, as well as technological and resource constraints;.

IV. a criterion in terms of which alternative organizational arrangements and behaviors can be compared, and which incorporates the effects of technological and resource constraints given from the environment, as well as the objectives of economic activity.

In other words, there should be a variable whose values correspond to different kinds of firms, a set of constraints on this variable expressing technological and resource constraints, the constraints on information processing, and incentive constraints on individuals, and groups of them, and a criterion of performance that expresses the goals of action. A model of this kind defines a set of possible firms. To solve the model means to find a subset of the set of possible firms that maximizes the criterion over the set of environments specified, subject to the constraints.

The distinction between positive and normative theory becomes subtle when theory is built on the idea that a firm is to be understood in terms of optimal adaptation to constraints. We may therefore take the viewpoint of a designer of firms, without commitment as to whether the theory is to be interpreted as positive or normative.

The aim of this paper is to take a step toward the formulation of such a theory. In this step the constraints arising from incentives are ignored. The focus is on the effects of the "prime source of transaction costs," namely the fact that information processing takes time and effort. This fact arises from limitations on the capacities of individuals and equipment to process information. This leaves us with the problem of solving for firms as optimal adaptations to the constraints on information processing capacities of individuals, i.e., adaptations to the bounds on rationality. Of course, a model with this limitation will not yield a complete theory of firms, but at best a theory of organizational units that are in some sense informationally distinguished from one another. Such a unit might be called an informationally distinct organization, or briefly, a division. As the statement quoted from Holmstrom and Tirole suggests, the problem in which only informational aspects are treated is analytically separable from that in which incentive constraints are considered. The Benchmark Case discussed later in this Introduction supports the view that confining attention to informational constraints yields a substantial area of problems with significant practical implications. Radner has noted that " ... a reasonable estimate is that more than one-half of U.S. workers (including managers) do information processing as their primary activity. If we add to managers (managers are those who figure out 'what to do', while workers are those who 'do') those who support managerial functions, we probably come out with roughly one-third of the U.S. workforce." (Radner, R. [18]). (Radner, R. [19], p. 1109).

We turn now to a brief, informal summary of the model, followed by a summary of the other contents of the paper.

We begin by considering how economic activity in a given class of economic environments might be organized. It is useful to think of organizing production in a class of environments in which production possibilities (technology of production), and constraints on the initial distribution of information about them, ("differentially informed experts") are given. In how many separate units ("divisions") should production be organized, and what should those divisions be doing (their structures)?

A class of environments, (represented parametrically in what follows) together with a performance criterion, or goal function, determine the optimal or desired action(s) to be taken in each possible environment; call this a desired decision function. For example, in a market economy a given production possibility set, and the goal of maximizing profit, determine the (decision) function that associates with each environment (technology, resources, etc.) the profit maximizing action(s) for that environment. (In this informal discussion functions also stand for correspondences.) A decision function (including its domain, the set of environments) and the constraints on the initial distribution of information define a coordination problem.

Managing production in a firm has many aspects. In this context, we ignore many of them and take the task of managing to be to figure out what to do; it is the task of computing the value of the desired decision function from the given information about the prevailing environment. This computation is constrained by the existing technology of information processing (distinct from the technology of production), and by the resources available for information processing tasks. To analyze this task, we need a model of information processing. The model of information processing technology used here is based on the Modular Network model (Mount and Reiter [15], [16], [17]). As extended here, the model allows us to define a "variable" whose "values" represent the available ways of solving the coordination problem. The variable is an Assigned Directed Acyclic Graph (ADAG).

An ADAG has several (four) components of cost associated with it. These include: costs of communication, a cost associated with the length of the computation, called "delay," and a cost associated with the number of agents used to carry out the computation. Two kinds of communication are distinguished; one is internal the other external. This distinction is important in the analysis. The motivation for it is discussed in the Rowing Example presented below.

It is assumed, mainly for simplicity, that the cost function is linear in these components. It follows that if the cost coefficients are positive, the cost minimizing solutions are found among those ADAGs that are efficient in the space of the cost components. These are Efficient ADAGs, or EADAGs. Each EADAG represents a solution to the coordination problem, i.e., to the managerial task of figuring out what to do given the decision function, the constraints on initial information, and the constraints on information processing. Each EADAG represents an efficient organization of that task.

The organization represented by an EADAG can be interpreted as consisting of a number (one or more) of informationally distinct organizations, depending on its structure. An

EADAG can be cut arbitraily into components, connected by external communication channels, called "crosslinks". The components can be regarded as separate organizational units, i.e., divisions, or as parts of a single division. The criterion is the relation between the number of crosslinks between components and the number of parameters characterizing environments. Roughly speaking, if the number of links is independent of the number of environmental parameters, then each component of the decomposed EADAG represents a separate division. If there is no such decomposition of the EADAG, then it represents one division. (Theorems that characterize EADAGs, and methods for constructing them are presented in Appendix 1 and Appendix 2.) The intuition behind this definition is discussed in Section 4.

Given the decision function, and the value function for which it is optimal, the"size" of the organization may or may not be bounded above. This depends on the net value function. The net value at a particular environment is the value corresponding to the actions determined by the decision function, less the costs of computing those decisions. Because environments are represented by parameters, the "size" of a coordination problem in a particular environment can be taken to be the number of parameters characterizing that environment. It is convenient to think of a class of environments such that the number of parameters is unbounded, called a large class of environments . If there is an environment with q parameters such that for all environments whose number of parameters is less than q, the net value function is positive, while for all environments whose number of parameters is at least q, the net value is not positive, then the size of the unit is bounded on that class. The bound on the size of a unit, and the size itself, which might be less than the bound if additional factors are involved, depends on all the components of cost, while the structure of units depends mainly on communication costs.

The organization of the rest of this paper is as follows. Because the approach taken in this paper is likely to be unfamiliar to most economists. it seems desirable to devote more effort to clarifying the intuitive underpinnings of the approach than might be needed for a more familiar model. We therefore begin with the case of a real manufacturing firm (a gear factory), very briefly described, that will illustrate some of the issues and ideas used in the formal model, and which is a real example to which the model is intended to apply.

In Section 2 the idea of coordination, which plays an important role in the case of the gear factory, is addressed directly, and its informational consequences are explored with the help of a thought experiment, the Rowing Example. In this example coordination requires that certain things become common knowledge among the participants. Whether common knowledge can be achieved depends on the nature of communication between the participants, which in turn depends on the organization in which the participants live. The consequences of different ways of organizing the activity to be coordinated motivate building the distinction between two types of communication, internal and external, into our formal model.

Following this, the informational task of coordinating is specified in terms of decision functions, and the Modular Network model. The salient features of this model are presented, following the formulation of Mount and Reiter [15] [16] [17] . Then the execution of an algorithm by assigning the component tasks to agents is modeled.

In Section 3 the costs of information processing are modeled, the concept of efficient assignments of tasks to agents is introduced, and the relationship between efficient assignments and cost minimization analyzed.

The interpretation of an assigned graph as a representation of organizational structure is discussed in Section 4.

In Section 5 the model is applied to three prototypical examples. In Example 1, which is intended to capture elements of the coordination problem in the gear factory, the efficient organization of production is in one unit; in Example 2, which is superficially similar to Example 1, but differs from it in an essential way, the efficient organization consists of two units. Example 3 is an Edgeworth Box economy. Two versions are considered, a static equilibrium version, and an iterative dynamic one. The efficient graphs yield decentralized organization into two units in the static case, and three in the iterative dynamic version.

It is also shown for Example 1 that the size of the one division which is the solution is bounded. This question is not addressed for the other examples.

Section 6 has two parts. The first presents some general results useful for analyzing the amount of external communication implied by a given coordination problem. This problem is closely related to a problem that has been addressed in two different contexts. One is the analysis of the communication requirements imposed by a distributed computation of a function (Abelson [1], in the smooth case); the second is the analysis of the communication requirements of realizing a given goal function by an (equilibrium) decentralized mechanism, which we may refer to briefly as the "message space literature." (A review of that literature can be found in Hurwicz IDE). Section 6 includes a brief summary of results based on Abelson [1],Hurwicz [6], Chen [2], and Hurwicz and Reiter [8].

The second part of this section summarizes William's genericity theorem (Williams [23], which shows that the coordination problem in Example 1 is prototypical in the sense that the function to be computed in that example is generic in the space of smooth functions with the Whitney topology. This suggests that we might expect that a large fraction of economic units would be multiperson organizations coordinated by internal information processing, i.e., that we should expect to see a lot of economic activity organized in informationally distinct organizations each consisting of more than one agent.

Finally, two appendices that present characterizations of, and methods for constructing, efficient assigned directed acyclic graphs from a given modular network with assigned inputs, may be found in Reiter [22].

A Benchmark Case

In building theory it is useful to have in mind a real instance of the phenomena the theory is designed to capture. The following brief description of the experience of one firm is presented for this purpose. The description is based on the author's experience designing and implementing a system for managing production in that firm. Only a sketchy description is given here. A more complete account can be found in Reiter [20]. The description given here is of the firm at the time when the system was designed and implemented.

The company manufactured high quality gears and gear assemblies, such as power transmission systems, for a variety of customers and applications, such as tractors, earthmoving equipment, among others. The gears ranged from small gears with diameters of about 3 inches to large gears about 3 feet in diameter. Some are on shafts, some on internal rings, external rings; virtually every type of gear and tooth geometry is made, and close specifications may apply to all physical and metallurgical properties.

These products were produced to order in a plant with about 1000 machines grouped in 250 work centers. The number of operations (on different machines) involved in producing a gear may vary from about 7 to about 50. In addition to its physical and metallurgical properties the promised time of delivery (due date) is important. Two aspects of delivery time are important:

(1) the length of time between the placing of an order and its due date, (called "lead time") ;

(2) the reliability of delivery at the requested or agreed time, called "meeting due date."

Several orders for the same physical gear placed at different times, or calling for different due dates even if placed at the same time are effectively different products. The specification of what is to be produced can be, and typically is, complex; in terms of the parameters needed for its description, it is of high dimension. Most orders are for between 50 and 300 pieces, but they can be for as few as 1 piece, or as many as 5000. The mix of orders shifts even in periods when the total volume of work remains relatively constant.

The performance on any one order depends on decisions made about how and when to produce the other orders in the shop, and on the state of machines in the shop during the relevant period. The task of managing production includes figuring out how to coordinate the execution of the thousands of operations needed to produce the mix of orders in the shop--figuring out what to do and when to do it. This task involves bringing to bear the values of many variables for each of hundreds of orders and thousands of operations. As the number of orders grows, the number of parameters that must be included in the calculations leading to the decisions required to manage production also grows, as does the complexity of the calculations.

Moreover, this task must be repeated periodically as old orders are completed, new orders enter and unforseeable events occur, such as machine breakdowns, or changes originating with customers.

The system in use at the time was the standard one for managing job shop production. Production managers decided when to release orders to the shop, using the due date, and allowing standard time estimates for completing the required work; the foreman of each work center scheduled the work in his work center using manual methods to calculate Gant charts; expediters from the production management department intervened in the shop to alter decisions in light of unforseen events and pressure from customers.

The new system that was designed and installed for this company combines computer programs for processing information and making decisions, human information processing, and human decision making.

The new system changed the information communicated within the firm and between the firm and its customers. It also changed the way management decisions are made about what to do and when to do it. The machines and workers in the shop were unchanged, at least until the firm decided to expand, and the incentive systems used were the same before and after the system was introduced.

The effects of the new system were manifold. We note here only two of them.

- When the installation of the system was completed there was a jump in productivity of about 33% The new level was maintained thereafter.
- After several years the company built a second plant, about the same size as the first one. Several years later it built a third plant. All plants were managed as one firm by the same system, with one computer.

The relevance of this case to the theoretical model presented below may be indicated by a few observations.

We are not here concerned with modeling production in the shop, but rather with modeling the management of that production. The decisions that must be made in order to produce a given assortment of orders with the given resources (machines, tools, labor and skills, etc) are the focus. Among these are the decisions as to what operations among those in the given assortment of orders will be performed on each of the 1000 machines at each moment of time, (the schedule).

It is conceivable that each machine or work center (a group of similar machines) could be managed by a separate business unit, a "mini-firm," even if physical considerations made it desirable that the work centers all be located in the same building.[1] Each mini-firm might, for instance, sell time on its machines, along with the other services needed to execute the specified operation on that machine. The buyers of time might be the customer submitting the orders, or for each order (job) a job-manager internal to the gear producing firm, who would know all the relevant information about a particular job. The idea behind this form of organization is that the prices of machine times would serve as the coordinating device. But, as the following simple example suggests, this hope is not likely to be fulfilled.

Consider a plant that consists of two work centers, A and B, each with just one machine, which we may also call A and B respectively. The customer' s job requires two operations, the first on machine A, the second on B. Each operation takes one hour per piece. The order is for 100 pieces. The due date for the job is D. We suppose that, because of the time and effort required to change the set-up of a machine, once processing starts on a machine it continues without interruption until all 100 pieces are done. Suppose to begin with that the customer has bought the interval [0,100] on machine A, and asks himself, What is the value to me of an interval [t, t'] on machine B? We consider a few specific intervals. For instance, the interval [100,200]. If the due date is later than 200, then this interval is worth more to the customer than it would be if it D were less than 200. If, for instance, D = 150, then the intervals [1, 101]. [2.102], ..., [50. 150] would each be worth more to the customer than would the intervals [t, t'] where t is greater than 50. Furthermore, given that the customer has already bought the time intervel [0,100] on machine A, he could meet any due date D later than 101, if he could buy the appropriate interval on machine B. Thus, the due date D = 101 could be met by buying the interval [1.101] on B and processing the first piece to finish on A, which it does at hour 1, immediately, so that it finishes on B at hour 2, the second so that it finishes at hour 3 and so forth.

However, if the due date were 150, value to the customer of an interval on machine A would be the same for all intervals [a,a'] where a is between 0 and 49, provided that the customer can buy the interval [50, 150] on machine 2. The customer cannot know the value to him of an interval of time on machine m, where m is either A or B, unless he also knows which interval of time on the other machine he will have to go with it. If the job has more operations in different work centers, in order for the customer to know the value to him of time in any one of them, he must know the intervals available to his job in all the work centers. Morever, since the intervals are determined by the processing time per piece in each work center and the number of pieces, when there are different jobs with different processing times per piece, and different numbers of pieces per job, the number of intervals that must be distinguished can be very large.

If prices are to be used to equate supply and demand for machine time, then each interval of time in each work center will require a different price. Even if a set of equilibrium prices (making supply equal to demand) existed, it would be a very large problem to find them, a problem not different from that of constructing production schedules directly. But we cannot be assured that equilibrium prices do exist, because of the indivisibilities inherent in the problem, and because of externalities also inherent in the problem. The latter arise because intervals of time on given machines that are each independently feasible, may together not be feasible. For example, the interval [50, 150] is feasible on machine A, and so is the interval [1, 101] on machine B, but the pair is not feasible. In other language, the production set describing the two machine technology is not the cartesian product of its projections onto the axes containing the individual production sets of machines A and B respectively.

The preceeding discussion of the coordination task in this gear manufacturing plant is informal. However, without claiming to have proved anything, it strongly suggests that the coordination problem presented in this case calls for all the information to be processed within a single organizational unit, and that, because of informational complementarities, management of production could not be improved by creating informationally distinct divisions, and might well do worse.

Moreover, as the number of orders (per unit time) grows, the amount of information (the number of parameters) to be taken into account also grows. With that growth, the difficulty of figuring out good schedules grows rapidly. The method used by the company was the standard way of calculating Gant charts, a calculation carried out in part by production schedulers and mainly by work center foremen. This could be described as applying an algorithm to the data about jobs and machines, using the resources available for information processing. The algorithms known at that time and the resources available to execute them are inadequate to the task of computing shop schedules when the number of jobs and machines is moderately large. While in fact decisions made in the shop, however arrived at, in effect determine a schedule, that schedule is generally far from efficient. Symptoms of inefficiency included long lead times, chronic failure to meet due dates, very large in-process inventories and in general a chaotic atmosphere of recurrent emergencies.

The size of the firm was not limited by technological constraints, or by the extent of the market, either of which could be binding, but in this case were not. Instead, the firm was limited in size by the capacity of the management to run it efficiently, i.e., by the technology of information processing, and by the capacities of the resources available for information processing, that is, by the limitations of the algorithms for scheduling production known at the time, and by the capacities of the resources (people and machines) for executing the algorithms.

The new system changed the technology of information processing, and therefore the technology of managing. The new system introduced a new algorithm for calculating shop schedules, and new resources in the form of computers, for executing the algorithm. The result was a large increase in the ability of the management to make good decisions to control production. This improved productivity and also permitted a large growth in the size of the firm they were able to manage.

The role of incentives in bringing about the changes this firm experienced in its internal organization, productivity and size was limited. The incentive system used in the business, both in the shop and for the management, was the same before and after. Incentives to make higher profits no doubt encouraged the management to run the risk of investing in the design of a new and then unproven system, but, their perception that the existing system was overwhelmed played a more important role than any then necessarily vague anticipation of higher profits.

This example tends to support the notion that it may sometimes be productive to consider information processing and computational constraints separately from incentive constraints. It also represents a real situation of which Example 1 below is a more abstract prototype.

We return to this case at the end of Section 4 after the proposed model has been presented.

Section 2. Informational limitations and coordination

Multiperson organization is in a sense forced on economic activity by limitations on the capacities of economic agents to communicate and process information, even with the aid of computers and telecommunications equipment. These limitations define constraints on organization analogous to the constraints on production imposed by technological knowledge generally. A design of organization that violates these constraints is Utopian in the same sense as one that violates other laws of nature, physical or human. However, within the limits of feasibility imposed by informational constraints there appear to be many choices, just as there are options in production. It is of central interest to the design of organizations, and therefore to the theory of firms, to know the structural organizational consequences of different coordination requirements under given limitations on information processing capabilities of economic agents.

The common understanding of the term "coordination" is that it refers to a situation in which several actions "fit well together," or "match." A formal way of expressing this in general terms is that at least two variables specifying actions or decisions are involved, and that not every feasible combination of these variables leads to the same “utility” of outcomes. In a simple setting, there is a function whose domain is the cartesian product of the domains of the two (or more) variables. Fixing a value of this function defines a locus of points in the product domain, a level set. A vector of variable values is deemed coordinated if it lies in the specified locus. An alternative formulation is that the value of the function defines a "degree of coordination." The following example helps to explore and clarify ideas about coordination that motivate the model presented subsequently.

Rowing, a motivating example.

We consider two oarsman in a racing shell. The speed of the boat depends on a number of factors: the power of the oarsmen, the pace-number of strokes per minute- the skill of the oarsmen in executing the strokes, and the interaction between the oarsmen. For any given characteristics of the individual oarsmen the speed of the boat depends on the degree of synchronization of the strokes.The strokes of different oarsmen should be synchronized to obtain the highest speed given the other characteristics of the strokes. Any deviation from simultaneity reduces the speed of the boat. That is, the actions of the oarsmen should be coordinated. To focus on the issue of coordination it is helpful to consider a thought experiment.

Instead of the two oarsmen sittting in one boat, imagine that there is a "sculling simulator", similar in concept to a flight simulator, in which the scullers (rowers) each sit alone in a room equipped with a rowing seat and a pair of oars. The simulator calculates the movements of the boat, including its speed, resulting from the actions of the two oarsmen.

Suppose the event involved is a trial in which the rowers seek to attain the highest possible speed for a given number of strokes, once they have started rowing. Maximum speed is attained when the two rowers synchronize each of their strokes, other things equal. We may simplify the situation by assuming that the pattern of a stroke in time is the same for the two rowers. Hence coordination reduces to synchronizing the start of each stroke, in particular the first stroke. Thus, each rower must decide when to start his first stroke. We may suppose that no rower will start his stroke unless he is assured that the other will also start at the same time, because one rower starting alone will destroy the possibililty of maximizing the speed.

Suppose that the simulator has a communication channel through which the rowers may send signals or messages to one another, say a runner who can carry messages down the corridor between the rooms. The channel is not completely reliable; the time it takes for a message to be transmitted is variable and unknown, and includes the possibility that a message sent may never arrive. One rower, say, rower 1, could send a message to the other of the form, "I will start rowing x time units from now." Because transmission takes some variable unknown time, rower 2 cannot know exactly when the message was sent, and hence cannot know exactly when to start rowing himself. Furthermore, he might think that rower 1 does not know whether his message was received, or, if received, when it was received.

This is a version of a problem well-known as the Coordinated Attack problem. In that problem two commanders are physically separated and can communicate via an unreliable channel. They must agree to attack at dawn. Here, while the time of the attack is not a problem, because the dawn is observed by both and this fact is common knowledge, it is not common knowledge that they will both attack at dawn. To attain common knowledge that they will both attack at dawn is equivalent to an infinite sequence of statements of the form, "I, commander A will attack at dawn." "I commander B will attack at dawn." "I commander A know that commander B will attack at dawn." "I commander B know that commander A will attack at dawn." " I commander A know that commander B knows that I will attack at dawn," and so on ad infinitum. In the Coordinated Attack problem, neither commander attacks, because common knowledge that they will both attack cannot be attained in finite time. In the equivalent rowing experiment, neither rower will start rowing.

Of course in actual rowing races, the organization of rowing allows the rowers to use a different channel of communication from that in the rowing simulator. The thought experiments helps to illuminate the actual rowing situation, in which the rowers sit one behind the other in the same boat. That arrangement somehow allows them to attain common knowledge of their agreement to start the next stroke at a particular time. In other words, physical proximity makes available channels through which a very large amount of information can be communicated quickly. Experience as a rowing team allows them to learn to use that information to coordinate their actions.

Another configuration is a rowing crew consisting of two oarsmen and a coxswain. Both rowers face the coxswain, who, like the conductor of an orchestra, gives the beat. (He also steers the boat.) This configuration makes the time of the next beat common knowledge for all practical purposes. Each rower senses the beat given by the coxswain, by hearing, seeing, and feeling the vibration of the beat through the boat, and each knows that the other is exposed to the same signals. (The coxswain actually beats a rhythm which the rowers use to predict the start of the next beat. Therefore it takes a period of experience together for the crew of three to come to have confidence in one another's responses. In this situation the common knowledge is perhaps of the functions that describe their individual reactions. Of course, in practice they may achieve only a close approximation to common knowledge. In the case of an eight person crew, it is evidently too difficult to achieve a satisfactory degree of coordination without the use of a coxswain. A coxswain is part of the crew in all eight oared crew racing events.)

To sum up, the coordination task involved in rowing efficiently requires the communication of a very large amount of information among the crew. If the rowers are separated and therefore must use the technology of remote communication, they cannot coordinate their actions well, while if they are physically close, a different type of communication channel is available which does enable them to coordinate well.

To anticipate the modeling to come we introduce a little notation into the rowing example. For the present purpose we consider the actions of the rowers to be a sequence of strokes. We take explicit note of the time at which each stroke starts, and assume that all other relevant quantities are built into the function whose value measures the performance of the shell, e.g., its velocity, or the time it takes to cover a given distance. Thus, let x_j and y_j denote the times at which rower 1 and rower 2 start stroke j, respectively. Then the sequence (x_1,y_1), ... , (x_n,y_n) determines the performance of the shell over the course of n strokes. The function

$$F(x_1,y_1,\cdots,x_n,y_n)=\sum_{j=1}^{n}\left(k-h\left(x_j,y_j\right)\right)$$

gives the performance of the shell once rowing has started. (To avoid complexities which we will not in fact do anything with, we do not model here what happpens if neither oarsman starts rowing.) Here k is a constant (actually a parameter determined by all the relevant quantities other than the time at which the jth stroke begins, and which have been assumed constant) representing the best performance given all the relevant quantities, and h represents the deduction from optimal performance due to failure of coordination. A particular example of the function h is

$$h\left(x_j,y_j\right)=\bar{h}\left(\left(x_j-y_j\right)^2\right), \quad where\ \bar{h}(\bullet)\geq 0, and\ \bar{h}(0)=0\ .$$

In informational terms, the problem of achieving coordinated behavior is that of computing the minimum of the function h. In general, as remarked above, from an informational standpoint a coordination problem is a problem of evaluating a function of the variables whose values are to be coordinated.

To sum up, tasks differ as to the amount of information required to achieve coordination. Some can require transmission of very large amounts information among the participants. The channels available for communication seem to be of at least two types, those for remote communication, , and those that apply when human beings are in close proximity for long periods of time. The latter seem to be capable of carrying a great deal of information, often subtle, quickly. The former seem to be of relatively limited capacity, i.e., relatively slow, and unreliable. The distinction between two types of communication links is important in the model presented below.

Firms and Information Processing

With these examples in mind, we return to the discussion of firms and economic organizations. In economic theory, the firm is typically viewed as a single economic agent whose behavior is described by a decision rule. In carrying out an action, the firm may be viewed as computing the value of its decision rule--a function from inputs that specify its environment, to outputs that designate its actions. (Here we are focusing on the routine operations of the firm.) Because of the limitations on information processing capabilities, the task of deciding must be spread among a number of individuals or agents. The outcome of this process depends on how the information processing is organized , i.e., on how the component tasks are allocated.

While a firm typically computes its decisions repeatedly as circumstances change over time, we abstract from the full dynamic problem and instead consider that the firm faces a set of possible environments and that it must compute its decision rule not just for one particular environment, but for <u>any</u> environment in that set. In the case of the gear factory, the management must be able to compute its shop schedule for any set of orders it might receive. The set of environments and the decision rule that characterizes a firm are specified once and for all. Exploration of a dynamic model in which the firm is viewed as adapting to a changing environment is reserved for subsequent analysis, in which the analysis carried out here would be one step.

Accordingly, a mode of organization in its informational aspect consists of:

(i) an algorithm for computing the decision rule, and

(ii) an assignment to agents of the steps required to execute the algorithm.

In the case of the gear factory, an environment includes the set of orders that must be produced, including specification of due dates and technical dimensions of the product, as well as the state of the machines in the shop given from the past at the time the decisions must be made. The decisions to be made include how and when all the required work will be done. Thus, the decision function has as inputs all the data available about orders and machines, and as outputs, the shop schedule for some prescribed interval of time into the future. This is the function to be computed. Note that while the shop schedule itself involves time, and is dynamic, the algorithm is one that computes a single shop schedule, having a single date at which it takes effect. In the gear factory this algorithm is embedded in a dynamic system that computes a sequence of schedules. We abstract from the latter dynamics here.

An algorithm for computing a function will be modeled here by a modular network, following the formulation of Mount and Reiter, [15], [16], [17][2]. In that model, a modular network is specified by a set of modules (elementary operations or functions), and a directed graph showing constraints on the order (partial) in which elementary operations can be performed. In other terms, a module can be visualized as a black box with possibly many input lines and one output line, taking one unit of time to compute its output from its inputs. (Here a unit of time is the time it takes to execute one elementary computational step and is assumed to be the same for all modules. Reference to time could be eliminated altogether by counting elementary steps instead.) A modular network consists of modules wired together subject to the condition that each input wire of a module is connected to at most one output wire. A module is interpreted as representing an elementary computation. What is considered to be elementary may be relative to the available means for computing. For instance, in some circumstances the basic arithmetic and logical operations may be taken to be the only elementary operations, while in other circumstances, say, when the computing is to be done by a person equipped with a computer and a program for finding the roots of a polynomial of given degree p in n variables, then one might sometimes want to consider finding roots of such polynomials to be elementary. The functions allowed to be modules, i.e., the operations assumed to be elementary, are restricted to a specified class.

The class of elementary functions, a primitive of the model, provides a formal way of expressing limitations on computational powers. The set of functions allowed to be modules might include, for example, Boolean functions, or Heavyside functions, or smooth functions, or polynomials of no more than a specified degree, or real analytic functions. (For some purposes it is appropriate to regard even continuous functions as elementary.) The class of elementary operations can formalize other limitations on computational abilities. For example, if, as has been pointed out by psychologists, the amount of information that a person can absorb at one time (or the inputs that a machine can accept simulataneously) is limited, then the class of elementary operations may be required to satisfy the condition that a module is a function of at most r variables, where r is a given positive integer, and each variable may be a d-dimensional vector, d a positive integer. A modular network that satisfies that condition is called an (r,d)-network.

Making the class of elementary operations a primitive of the model gives control over the level of reduction in a particular application, because a computation need only be reduced to expression in terms of the operations specified as elementary. (The graph expressing the structure of the algorithm involved is also a primitive of the model. Restrictions on the class of directed graphs allowed can also express bounds on the rationality of the agents. It should also be noted that the possibility of d > 1 allows the elementary operations to include conditional switches even when modules are restricted to be continuous or even smooth functions of real variables.)

Execution of a computation--assignments

The process of arriving at a decision is modeled as one of computing the value of the decision rule from observation of the values of its arguments. The possibilities of observing the values of variables in the environment are restricted. These restrictions are here assumed to be given; this assumption expresses the idea of "differentially informed experts" quoted in the Introduction. The computations called for by the algorithm used to evaluate the decision function must be performed by individuals, perhaps with the aid of computers or other equipment. The computational capacities of an individual, or an individual-computer combination, are expressed by the set of elementary operations (modules) that individual can execute.

(2.1) The assignment of modules to individuals is made subject to three constraints:

1) Any module assigned to an individual must be one of her elementary operations;

2) A module representing the observation of an input variable must satisfy the restriction on who may observe what;

3) Parallel Constraint: Each individual is capable of carrying out at most one elementary operation in one unit of time.

Thus, even if the algorithm allows two particular operations to be carried out in parallel, and therefore both could in principle be executed in one unit of time, (or one computational step) that can be done only if those operations are assigned to different individuals.

Different assignments of elementary operations can induce different organizational performance. For instance, the time (number of sequential computational steps) needed to complete the evalution of the function, and the patterns of communication and memory involved in the process depend on the assignment of modules to individuals. These factors play a central role in our analysis. If the computations can be spread out among many people, the time required may be reduced by doing them in parallel. But distributing the computation may entail more communication among the individuals, which might offset or even reverse the advantage of parallel operation. Furthermore, using more individuals, who are typically employees of the firm, usually entails higher costs. Modeling the time and effort of information processing when individuals have limited information processing capacity is the focus of the next section.

Section 3. Costs of information processing and efficient assignments

Suppose that the function $P : X \rightarrow Y$, where X and Y are Euclidean spaces, is the function to be computed, i.e., the decision rule. We might take "optimal adaptation to informational constraints" to mean organizing the required computation so that the cost of carrying it out is minimum. To this end, we first seek efficient ways of carrying out the required computation. Minimum cost is achieved by choosing among the efficient organizations of computing given the cost weights, or prices, which allow the different dimensions of cost factors to be combined into a cost figure.[3]

Suppose that there is an (r,d)-network $\mathcal{N}$ with modules in the class $\mathcal{F}$ that computes P in time t*. If t* is minimal for (r,d)-networks with modules in $\mathcal{F}$, then t* is the computational complexity of P relative to $\mathcal{L}$. (Mount and Reiter, [16], [17]). We suppose that the network $\mathcal{N}$ is one that achieves the minimum delay t*.

The network $\mathcal{N}$ may have feedback loops. It is well understood that such networks can be delooped. It is shown in Mount and Reiter (Mount and Reiter, [16], [17])[4] that for each such network $\mathcal{N}$, and a function P that it computes in time t*, there is a loop-free network, a directed acyclic graph (DAG), G, with the following properties:

(i) G has the same modules as $\mathcal{N}$, with the possible addition of projections;

(ii) G computes P in time t*.

We may therefore without loss of generality suppose that G is a tree.
The length of the longest walk from a leaf to a root of G, is t*. Denote by M the number of vertices of G that have modules other than projections. Denote by C the number of arcs in G that connect a pair of modules that are not both projections.

Figure A and Table I show a modular network and the computations it carries out. Figure B shows the tree that computes the function that the network in Figure A computes in 4 units of time (four elementary sequential computational steps).

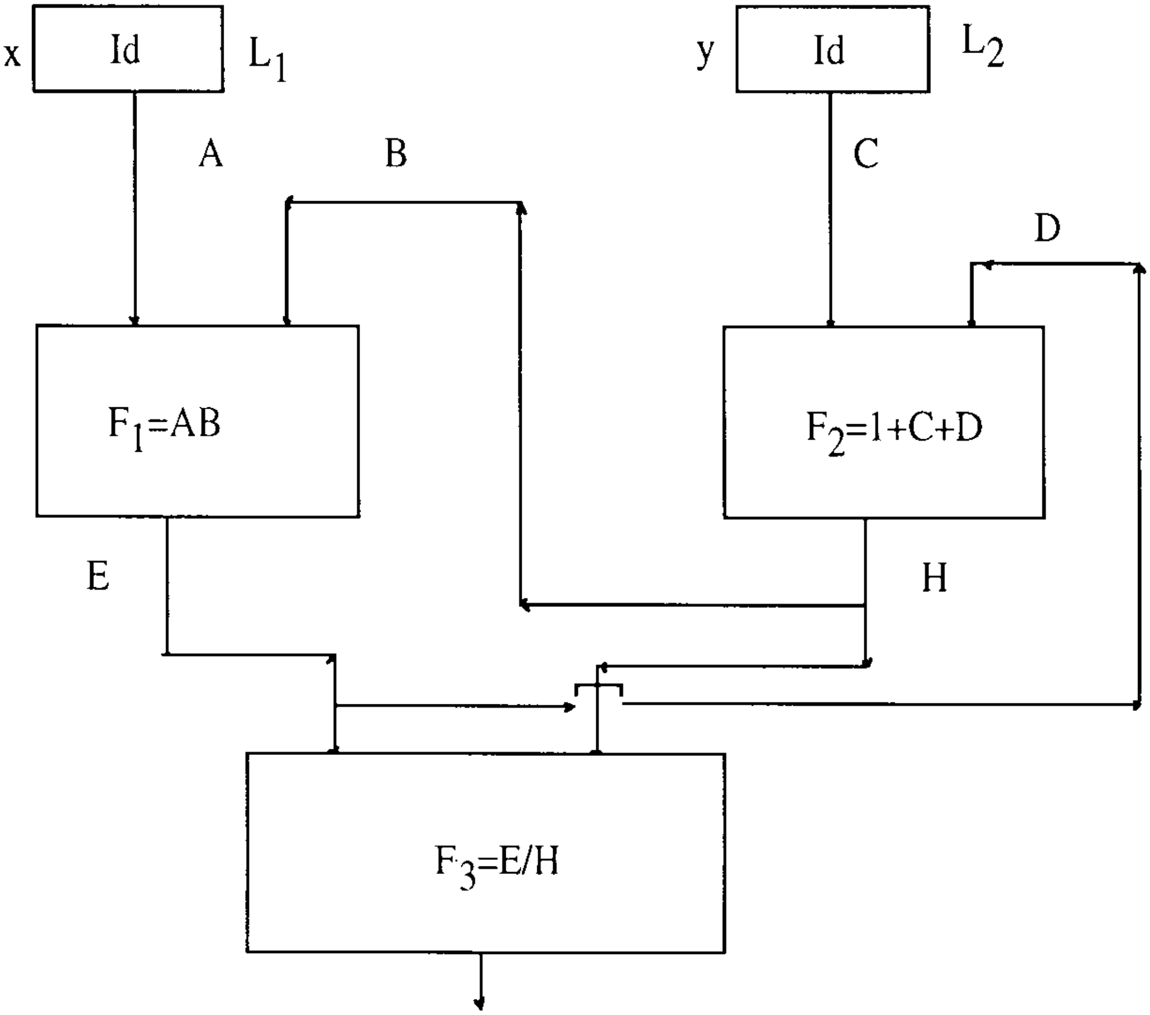
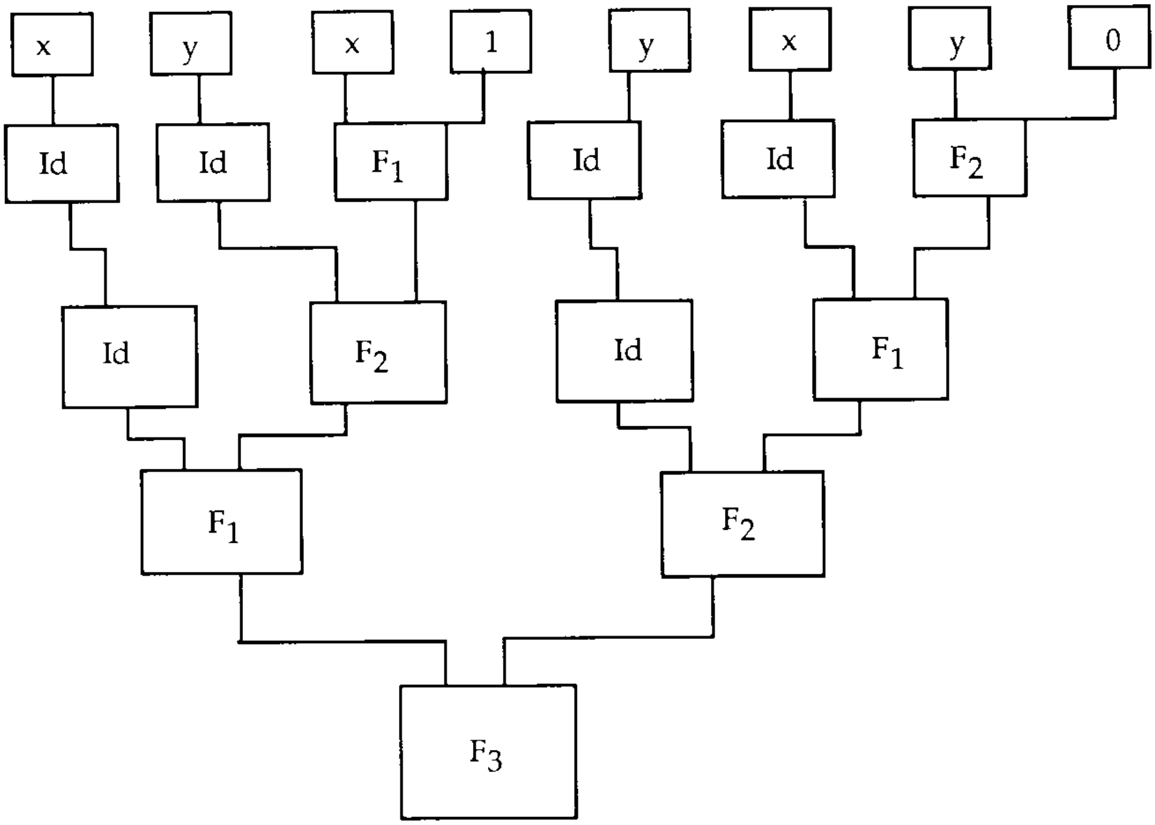


Figure A Mount and Reiter (1980)

	L_1	L_2	F_1	F_2	F_3
σ	0	0	1	0	0
t					
0	x	y	0	1	0
1	x	y	x	$(1+y)$	0
2	x	y	$x(1+y)$	$(1+x+y)$	$x/(1+y)$
3	x	y	$x(1+x+y)$	$(1+x)(1+y)$	$\frac{x(1+y)}{1+x+y}$
4	x	y	$x(1+x)(1+y)$	$(1+y+x(1+x+y))$	$\frac{x(1+x+y)}{(1+x)(1+y)}$
5	x	y	$x(1+x+y+x^2+xy)$	$(1+y)(1+x+x^2)$	$\frac{x(1+x)(1+y)}{(1+y+x(1+x+y))}$

Table I

Figure B



The function computed in time 4 by the (2,1)-network in Figure B has the following expression as a superposition,

$$F_3(F_1(x, F_2(y, F_1(x,1)))), F_2(y, F_1(x, F_2(y,0))) =$$

$$\frac{F_1(x, F_2(y, F_1(x,1)))}{F_2(y, F_1(x, F_2(y,0)))} = \frac{x F_2(y, F_1(x,1))}{1 + y + F_1(x, F_2(y,0))} =$$

$$\frac{x(1 + y + F_1(x,1))}{1 + x + x F_2(y,0)} = \frac{x(1 + y + x)}{1 + y + x(1 + y)} = \frac{x(1 + x + y)}{(1 + x)(1 + y)} \; .$$

Assigning modules

We use the term agent to stand for an individual, or a computing devise, or an individual with a computing device capable of carrying out an elementary operation, i.e., capable of evaluating a module.

Let the set of agents be $\{1,\cdots,N\}$. Let a denote an assignment of the modules of G to agents, i.e., a is a function from the set of modules of G to $\{1,\cdots,N\}$. The restriction of a to the leaves of G must agree with the given restrictions on *who* may observe *what*. The assignment a determines several quantities of interest. First, the length of time (number of sequential steps) required to compute P under the given assignment, denoted $\tau(a)$. Second, the amount of communication that takes place within each agent.. Third, the amount of communication that takes place between different agents. Because the graph G is acyclic, (and so is the assigned graph) each arc carries one "message" in the course of a computation. Therefore the number of arcs is a measure of the amount of communication required to execute the algorithm represented by G. The distinction between different types of communication channels is modeled in the simplest way, namely, that there are two types of channels. Therefore, an assignment a determines the number of arcs of G that go between modules assigned to the same agent, (referred to as internal communication links, or briefly, selflinks); this number is denoted $c_1(a)$; the number of arcs of G that go between modules assigned to different agents, (referred to as external communication links, or briefly, crosslinks) , is denoted $c_2(a)$. The difference between the two types of channels is expressed as a cost difference.

Finally, an assignment a also determines the number of agents who are assigned modules of G, denoted n(a) ≤ N. These items determine costs associated with the assignment a. Let

$$\overline{\chi}(a) = \chi(c_1(a), c_2(a), \tau(a), n(a))$$

denote the cost function, a real valued function of the arguments shown.

We assume for simplicity that χ is linear and that the coefficients satisfy some inequalities.

Thus,

(3.1) $\chi(c_1(a),c_2(a),\tau(a),n(a)) = \alpha_0 + \alpha_1 c_1(a) + \alpha_2 c_2(a) + \alpha_3 \tau(a) + \alpha_4 n(a)$

and,

(3.2) $0 \le \alpha_1 \le \alpha_3 < \alpha_2, \quad 0 \le \alpha_4 \le \alpha_3 \quad 0 \le \alpha_0 .$

Notice that the assumptions (3.1) and (3.2) say that all selflinks have the same cost per link; crosslinks have a higher cost per link, but the same for all crosslinks. Since the network is acyclic, each link is used only once per computation, so the cost measured can also be interpreted as including the cost per message over the link. Use of an internal link may be interpreted as a retrieval from the "memory" of a single agent, since the link connects two modules that are executed by the same agent.

The interpretation of a crosslink and its cost coefficient is more complicated, because, as the rowing example suggests, the nature of a crosslink depends on whether the two agents involved are or are not in the same organizational unit. "Learning curves," "organizational learning," "organizational culture," "organizational memory" and "learning-by-doing," all suggest the existence of mechanisms of coordination internal to an organization, that do not operate across boundaries between organizations. They refer to situations in which close, stable and persistent interactions among agents allow them to observe one another's behavior closely and repeatedly in different situations, and so in these circumstances a transmission between agents in the same organization is more like a retrieval from memory than it is like a message exchange between strangers. In this way the cost of transmitting information between different agents in the same organization is less than it would be if those agents were not in the same organization. This can be expressed in the present model by making the cost coefficient of a crosslink, a link between two different agents, depend on whether the the agents are or are not part of the same organization. By choosing to put different agents in the same organization the cost per message of communication between them can be reduced. But we assume that this can be done only by incurring a fixed cost. The additional component of fixed cost represents the capital cost of creating and maintaining an organization as a going concern, and includes the cost of the internal infrastructure that facilitates internal communication.[5] Let α_2^* denote the cost of a crosslink between agents in the same organization or firm, and let α_2^{**} be the cost of a crosslink between agents in different organizations. We assume that

(3.3) $\alpha_1 \leq \alpha_2^* < \alpha_2^{**} = \alpha_2 ..$

The fixed cost increment α_0' is required in order to change crosslinks from external to internal, and hence the cost coefficient from $\alpha_2^{**} = \alpha_2$ to α_2^*.

Cost minimizing and efficient assignments

The problem of finding cost minimizing assignments is complicated by the fact that the minimization is subject to nonlinear integer constraints due to the complex interdependence of the cost factors, c_1, c_2, τ, n, given the directed graph that describes the algorithm. Instead of looking for minimum cost assignments directly, we first focus on efficient assignments, for a given number of parameters. We can consider this problem independently of whether crosslinks are or are not internalized. This becomes clear when we see that the properties of the cost functions on which the result depends are the same in both cases.

We show next that the problem of finding efficient assignments reduces to the problem of finding efficient pairs, $c_2(a), \tau(a)$ for each fixed value of n, where a denotes the assignment. That is, for any number of individuals to whom modules may be assigned, we find the efficient combinations of the number of external communication links (whether external or internal) that result from the assignment, and the length of the computation that results from the assignment.

To see this we consider an assignment a, and note the following simple facts.[6] (To lighten notation we drop reference to a where it is possible without confusion, and indicate the effects of different assignments by other notation. We should note that Proposition 1 is valid for a broader class of cost functions than those for which it is stated. However, it is stated here for the class we are concerned with, and because of the interpretation, for cost parameters satisfying assumption (3.1).)

Proposition 1.) If (c_1, c_2, τ, n) is not efficient then (c_1, c_2, τ, n) does not minimize cost for any $(\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$ satisfying the assumption (3.1).

Proof of 1) Suppose (c_1, c_2, τ, n) is not efficient. Then there is some assignment that yields (c_1', c_2', τ', n') such that $c_1' \leq c_1$, $c_2' \leq c_2$, $\tau' \leq \tau$, $n' \leq n$, with at least one strict inequality. First, let C be the number of arcs in G (not counting projections), and note that

$$c_1 + c_2 = C,$$

i.e., each arc of G is assigned to be either an internal or an external communication link. Therefore, $c_1 = C - c_2$, and hence $\overline{\chi}(a) = \alpha_0 + \alpha_1 C + (\alpha_2 - \alpha_1)c_2 + \alpha_3\tau + \alpha_4 n$.

Therefore,

$$\overline{\chi}(a') - \overline{\chi}(a) = (\alpha_2 - \alpha_1)(c_2' - c_2) + \alpha_3(\tau' - \tau) + \alpha_4(n' - n) < 0,$$

which means that (c_1, c_2, τ, n) does not minimize cost.

2) It is also obvious that if, for a given assignment, (c_2, τ) is not efficient then neither is (c_2, τ, n).

Suppose (c_2, τ, n) minimizes cost. Then by Proposition 1) it is efficient. Denote by $(c_2^*(n), \tau^*(n))$ the number of crosslinks and delay that results from a cost minimizing assignment $a^*(n)$to n agents. I.e.,

$$\chi(c_2^*(n), \tau^*(n), n) = \chi(c_2(a^*(n)), \tau(a^*(n)), n) \leq \chi(c_2(a(n)), \tau(a(n)), n) \equiv \chi(c_2(n), \tau(n), n)$$

for every n and $a(n)$, such that $a(n)$ is an assignment to n agents. Let n* minimize $\chi(c_2^*(n), \tau^*(n), n)$ with respect to n. Then

4) $\left(c_2^*(n^*), \tau^*(n^*), n^*\right)$ minimizes cost over all feasible assignments.

By Proposition 1), it follows that $\left(c_2^*(n^*), \tau^*(n^*), n^*\right)$ is efficient. Hence, the assignment of modules to n* agents that results in the pair $\left(c_2^*(n^*), \tau^*(n^*)\right)$ is an efficient assignment.

As is the case in other familiar contexts, the search for cost minimizing assignments can be confined to search among efficient assignments.

Procedures for constructing efficient assignments, which it should be noted involves both assigning nodes to agents, and scheduling their execution in time subject to the parallel constraint and the precedence relations given by the modular network representing the algorithm, are presented in Appendices I and II respectively of [Reiter 1995].

Section 4. Assigned graph as a representation of organizational structure

Given the decision function, an algorithm for computing it represented by a minimal delay tree, G, and an assignment of the leaves of G, which specifies who can observe what, an assignment of the modules of G to agents results in a new graph that satisfies the three constraints in (2.1). The graph resulting from each such assignment may be interpreted as a (multiperson) organization for performing the managerial function of deciding what to do, given the restrictions on the dispersion of information, and the constraints on each individual's capacity to process information, including the Parallel Constraint. It is the formal entity in the model that is to be interpreted as one or more divisions. For this interpretation we need to know what properties distinguish an assigned graph that is interpreted as representing one division from one that represents two or more divisions.

Before presenting the definition, a few remarks may be helpful in explaining the intuition behind it.

An organization or firm is often likened to an organism in that it must adapt to its environment. If we consider from an informational viewpoint how an individual adjusts his or her behavior to the environment, as against how two or more persons do that, it seems clear that the internal mechanisms used for coordinating the behavior of an individual are different in kind from those used to coordinate the behavior of two persons. Neither neurons nor hormone flows extend through the skin of a person to the inside of another, nor do light or sound waves penetrate directly to the inside of a person.

In the case of multiperson organizations the distinctions between the mechanisms for internal coordination and those that mediate interactions with the outside environment analogous to the clear anatomical and physiological distinctions that apply to human beings individually are not immediately obvious. In the case of multiperson groups, written documents transmitted by mail or fax, conversations in person or via telephone, and the like are used for both internal communication and communication with those not in the organization. Unlike neurons, which do not go outside a person's skin, telephone lines cross the boundaries of firms. Moreover, units that are legally or descriptively parts of the same organization may be functionally separate in the sense that what coordinates their interactions with each other is not "anatomically or physiologically" different from what coordinates their interactions with what is outside the formal organization. Nevertheless, the rowing example of Section 2, and other observations, suggest that communication channels exist among members inside the same organization that are orders of magnitude higher in capacity and therefore in speed, and lower in cost, than the communication channels that exist between individuals in different organizations. Thus, what distinguishes mechanisms for coordination inside a multiperson organization from those that coordinate its interactions with the outside world is the existence of high capacity internal communication channnels. These should be understood to include the effects of shared knowledge and memory. Such channnels permit communication of highly complex and subtle information. This sort of channel is difficult to observe directly, because it operates through interactions of persons, and is not necessarily embodied in hardware, although persons may employ hardware (e.g., a telephone, or a local area network) in a process that flows naturally into and out of personal interactions. We have expressed this distinction in terms of selflinks and crosslinks, and the differences between them in terms of cost.

Given a set of environments, a decision function and an initial distribution of information about the environment, we construct efficient assigned graphs. An efficient assigned directed acyclic graph (EADAG) specifies how the agents involved compute the decision variables. These efficient graphs represent solutions to the problem of adapting to the constraints on information processing. If the solution graphs have many crosslinks, then it may be better (lower cost) to pay the fixed cost of setting up an organization to internalize some or all of those crosslinks so that the communication involved is done via the cheaper kind of crosslinks (the high capacity internal channels) and the remaining communication between agents inside and those outside the organization uses the more costly external channels.

Furthermore, we must recognize that the graph involved depends on the number of parameters. In the gear factory example the number of variables that appear as arguments of the decision function, which is the number of variables that characterize a set of orders, can vary. Since the system for computing the schedule must work for any set of orders received, we must look at the solution EADAGs over the entire set of environments under consideration. That is, if q represents the number of parameters characterizing an environment, we must consider how the efficient graphs vary with q, for the set of environments we are dealing with. So, suppose that for some value of q, an efficient assignment results in the vector $\left(c_1(q), c_2(q), \tau(q), n(q)\right)$. If we treat all agents connected by a crosslink as in different organizations, the resulting cost is

$$\overline{\chi}_2(q) = \alpha_0 + \alpha_1 c_1(q) + \alpha_2^{**} c_2(q) + \alpha_3 \tau(q) + \alpha_4 n(q),$$

while if we put all agents associated with crosslinks in the same organization, the cost is

$$\overline{\chi}_1(q) = \alpha_0 + \alpha_0' + \alpha_1 c_1(q) + \alpha_2^{*} c_2(q) + \alpha_3 \tau(q) + \alpha_4 n(q).$$

Costs favor internalizing crosslinks in one organization if

$$\overline{\chi}_1(q) - \overline{\chi}_2(q) < 0.$$

This reduces to

$$\alpha_0' < c_2(q)\left(\alpha_2^{**} - \alpha_2^{*}\right),$$

which, since $0 < \alpha'_{0},$ and $0 < \left(\alpha_2^{**} - \alpha_2^{*}\right),$ can be written

$$(4.1)\qquad \frac{\alpha'_{0},}{\left(\alpha_2^{**} - \alpha_2^{*}\right)} < c_2(q).$$

Thus whether to centralize the computation in one organization or not depends on how the number of crosslinks depends on the number of parameters. Since the computation must be carried out for any environment in the given class, and since the left hand side of the inequality is a constant, if the possible values of q forms a large set, say, an unbounded set as is typically the case in economic models, and if the number of crosslinks increases with q, then eventually (4.1) will be satisfied, and organization in one unit will be optimal.[7] We may take our cue here from a familiar example in economic theory, namely trade between two individuals. There the class of environments is customarily infinite dimensional.

For example, in an Edgeworth Box model representing trade in two goods, between two agents, an environment consists of the preferences of each agent, and the aggregate (or individual) resource endowments. (The commodity space and consumption sets are assumed to be fixed.) Any continuous and convex preference relation is allowed. Hence the set of environments includes infinite dimensional preferences, as well as those characterized by a finite number of parameters, such as a preference relation represented by a quasi-linear utility function, or one represented by a Cobb-Douglas utility. If environments are represented by real parameters, then the number of parameters specifying an environment in this class is not bounded. We say that a class of environments with this property is a large class. Suppose that a decision function is given, such as one that specifies some Pareto optimal trade (or allocation) for each environment. Then the computation of the decision as a function of the parameters can have an unbounded number of crosslinks. That is, the computation of the decision function imposes the requirement that an unbounded number of crosslinks may be needed. On the other hand, if the computation is such that all but a fixed number of crosslinks can be internalized by agents, then those agents are natural candidates for organizational units. In the Edgeworth Box model, the familiar result is that the computation splits into two parts, one internal to each agent. The internal computation of an agent may involve arbitrarily many parameters, but the communication between them depends only on the number of goods, and is independent of the number of parameters needed to specify preferences. As we show in Example 3 below, in the case of verifying a Pareto optimal trade in an Edgeworth Box economy, only two numbers need be transmitted (and in the case of a stable adjustment process no more than four) independent of the number of parameters.

In the case of the gear factory, the nature of the decision function leads to the conclusion that all efficient assigned graphs involve a number of crosslinks that increases without bound in the number of parameters. This suggests that for efficient coordination the entire calculation of production schedules be internal to one organization. There is no split such that the communication required between different agents is bounded.

Of course, even if the computation is organized in one unit, it would generally be the case that the cost of executing it would increase with the number of parameters, and in some cases reach a size for which it is no longer worth figuring out the decision.

These considerations motivate the following definition, presented next informally.

Suppose a coordination problem is given on a large class of environments. That is, we are given a function from the set of environments to a space of decision variables, and restrictions on the observability of environmental parameters expressed by an assignment of input variables (environmental parameters) to economic agents, (possibly multiperson agents). Consider the environments with a certain fixed number of parameters, q, and suppose we have a solution EADAG, G(q) for the class of environments with q parameters. Suppose that G(q) can be split into component subgraphs, each consistent with the initial assignment of information about the parameters. The components of G(q) represent candidates for organizational units. Consider the communications reguired within and between the components. Communication between components is represented by crosslinks, i.e., arcs of the graph that go from one component to another. Suppose that as q varies the component structure of G(q) remains the same. I.e., there is a one-to-one correspondence between the component subgraphs of G(q) and G(q′) for any two values of q and q′.

Of course, as q increases the number of vertices and arcs in each component may increase, and hence the amount of communication (and computation) within a component may increase. But it is possible that the amount of communication between components is independent of q, i.e., constant. If that is the case then there is a sharp distinction between the different components; communication between components remains constant, while communication (and computing) within each component may, and typically does, grow without bound. This property serves to define the boundaries between organizational units. These units may be interpreted as informationally distinct divisions.

To sum up so far, the analysis proposed here has the following structure. The basic economic unit is an individual or person. Individuals are equipped with (limited) capacities to observe, communicate and compute. There is given a set of possible environments, say, a set of possible technologies. These are specified parametrically. Information about a technology is dispersed among individuals in the sense that a particular person can observe only some parameters. There is a function that associates to each environment some optimal (with regard only to technological and resource constraints) or desired behavior. This function may be derived, say, by optimization, from the given data about the environment. The coordination problem is to determine the desired or optimal actions for a given environment from the dispersed information about the environment and subject to the information processing limitations of individuals. The analysis results in an algorithm for the required computation, and an assignment to individuals of the operations to be performed that is efficient in the sense that the determinants of cost form an efficient configuration. An efficient assignment of the computations is described by a multiperson modular network, an EADAG. The criterion for distinguishing separate entities is applied to the structure of this network (or networks) to determine whether the organization corresponding to the efficient network(s) should consist of one or more separate (multiperson) units, each of which may be interpreted as a separate division.

The preceding interpretation of a solution graph as organizational structure depends on communication and not on other determinants of cost, such as delay and the number of persons employed. However, these elements play a role in determining the size of the organizational unit in the static analysis, and an even more important role in analysis of dynamics. Dynamic analysis is reserved for another paper.

We may revisit the gear manufacturing case in light of the model just presented. First, in that case the central task involved in managing production is to compute a "good" shop schedule, given an assortment of orders. The pre-existing technology for doing this (the available algorithms) was the then standard system for determining Gant charts, and related systems such as Material Requirements Planning (MRP). The resources available to carry out the computations involved were people (production managers and foremen) with office equipment such as calculators, printed forms, telephones and the like, and the then existing commercial computer systems. The management of the gear company had determined that applying the new resources (existing computer systems) using the old algorithm would increase costs, but probably not result in a significant improvement in performance. They were probably correct in this assessment. The new system was based on a new algorithm for computing shop schedules, and used new resources for information processing, mainly new computing equipment which was announced, but not then immediately available. The nature of the decision to be computed was the same for each system. Expressed as a modular network, it does not split into subgraphs with limited communication between them. The old algorithm executed by people without computers becomes infeasible (very costly) at low values of q, i.e., a small set of orders. The old algorithm executed with new equipment (computers) does not yield significantly better performance and costs more even for low values of q. The new algorithm executed by people with the old resources does better then the old one for low values of q, but as q increases also quickly becomes infeasible. The new algorithm executed by people with computers produced the improvement in productivity described above and remained relatively cheap as q increased, thus enabling a tripling in the size of the firm.

In the next section we apply the model to examples . These examples may seem rather special, but, as we see in Section 6 below, they are prototypes of situations that prevail quite generally. The first example includes a version of the kind of coordination that occurs in rowing, as comparison between the functions to be optimized in the two examples shows. It also abstracts in a highly simplified form an essential feature of the gear manufacturing firm. As in that case, and in the Jordan-Xu model [Jordan and Xu [10)], in Example I, coordination requires that all parameters involved must be brought to bear on the decisions to be made.

Section 5. Three Examples

We present three examples in this Section. In analyzing these examples we suppose that each agent has a set $\mathcal{F}i$ of elementary operations, and is subject to the parallel constraint. For definiteness, we take $\mathcal{F}i$ to be the elementary operations of arithmetic of real numbers, together with the operation of raising a number to an arbitrary positive integer power. Thus, in particular an elementary operation can have at most two numbers as input.[8]

The set of possible environments is represented by a parameter space Θ, that has real coordinates. We say that Θ represents a large class of environments if for every natural number n there are points in Θ with at least n nonzero coordinates. That is, for each n there is at least one n-dimensional Euclidean subspace $\Re^n$ that is contained in Θ.

To avoid complexities that are not essential here, suppose that the subspaces $\Re^n$ are nested. I.e., for $n = 1,2,\cdots$ [9]

$$\Re^n \subset \Re^{n+1}$$

In that case all functions defined on the set of environments can be indexed by the parameter n, or q, representing the number of parameters.

Example 1.

Consider a two-stage production process. The first stage technology consist of two processes, P_1 and P_2. The second stage consist of a process P_3 which combines the outputs from P_1 and P_2. Thus, the process P_i is specified by a function

$$f^i : X^i \times \Theta^i \to Y \quad i = 1,2,$$

where Y is the space of properties of the outputs of P_i, including their quantities, X^i is the space of activity levels of process P_i, and Θ^i is a space of parameters relevant to process P_i. In this formulation

$$f^i\left(x^i,\theta^i\right) = y^i, \quad i = 1,2.$$

Suppose that $f^1\left(x^1,\theta^1\right)$ produces x^1 units of (intermediate) output 1 and that the vector $\theta^1 = \left(\theta_1^1,\cdots,\theta_q^1\right)$ of parameters of P_1 determines the characteristics of output that are not included in the specification of the commodity. For simplicity assume that the mapping from parameters of P_1 to characteristics of the product is the identity. I.e., process P_1 when conducted at the level x^1 produces x^1 units of output whose 'hidden' characteristics are θ^1. Similarly, $f^2\left(x^2,\theta^2\right)$ produces x^2 units of (intermediate) output 2, with hidden characteristics θ^2.

There is a third process P_3 in which the outputs of P_1 and P_2 are combined to produce a third (final) output. Thus, P_3 is given by $f^3\left(z,\theta^3\right)$ such that P_3 has the outputs of P_1 and P_2 as inputs. We suppose that the hidden components of the output of P_3 are determined by those of the outputs of P_1 and P_2 as follows. When the outputs of P_1 and P_2 are combined, the values of the parameters θ^3 are functions of θ^1 and θ^2. Specifically,

$$\theta_j^3 = \left(\theta_j^1 - \theta_j^2\right)^2, \quad j = 1,\cdots,q.$$

Suppose further that effort is expended in combining the outputs of P_1 and P_2. Suppose that when an amount of effort z is devoted to P_3 the effect is to determine the value of a unit of output as follows:

$$K - \left((1-z)\left(\sum_{j=4}^{q}\left(\theta_j^1 - \theta_j^2\right)^2\right) + wz^2\right).$$

Here K is the value of a unit of the output of P_3, (in a market this would be the price), and the second term is the deduction from value corresponding to the effort devoted to combining outputs of P_1 and P_2 when the parameters are θ^1 and θ^2, and the cost of the effort z is quadratic. The quantity $\sum_{i=1}^{q}\left(\theta_i^1 - \theta_i^2\right)^2$ measures the extent to which a unit of output of Process 1 and one of Process 2 are mismatched. Work done in Process 3 to correct the mismatch is measured by z, and the effect of the mismatch of characteristics depends on the value of z. Then, the net value of the output resulting from operating the three process with the intensities x^1, x^2, z is

$$V\left(x^1, x^2, z; \theta^1, \theta^2\right) = \min\left\{x^1, x^2\right\}\left(K - \left((1-z)\left(\sum_{j=4}^{q}\left(\theta_j^1 - \theta_j^2\right)^2\right) + wz^2\right)\right) - c\left(\left(x^1\right)^2 + \left(x^2\right)^2\right)$$

where c is a cost parameter, the same for processes P_1 and P_2, and z belongs to the interval [0,1].

With this technology efficient production corresponds to maximizing the value of V subject only to $x^i \geq 0$, i =1,2, and $0 \leq z \leq 1$. To simplify notation a little, write

$$\begin{aligned}
x &= \min\left\{x^1, x^2\right\}, \\
a &= \left(a_1, \cdots, a_q\right) = \left(\theta_1^1, \cdots, \theta_q^1\right) = \theta^1 \\
b &= \left(b_1, \cdots, b_q\right) = \left(\theta_1^2, \cdots, \theta_q^2\right) = \theta^2
\end{aligned}$$

and

$$D = \sum_{j=1}^{q} \left(a_j - b_j\right)^2,$$

in which case V can be written as

$$V\left(x^1, x^2, z, a, b\right) = x\left(K - \left((1 - z)D + wz^2\right)\right) - c\left(\left(x^1\right)^2 + \left(x^2\right)^2\right).$$[10]

Assume that

$$D = \sum_{j=1}^{q} \left(a_j - b_j\right)^2 \leq K,$$

and that

$$\frac{D}{2} \leq w \leq K.$$

Writing

$$\alpha(z) = K - (1 - z)D - wz^2$$

the first order conditions are,

$$\frac{\partial V}{\partial z} = x(D - 2wz) = 0$$

$$\frac{\partial V}{\partial x^i} = \frac{\partial x}{\partial x^i}\alpha(z) - cx^i = 0, \quad where\ \frac{\partial x}{\partial x^i} = \begin{cases} 1 & \text{if } x = x^i \\ 0 & \text{otherwise} \end{cases}, \ i = 1,2.$$

Solving for the optimal values $\hat{z}, \hat{x}^1, \hat{x}^2$, we get

$$\hat{z} = \frac{D}{2w}$$

and hence

$$\alpha(\hat{z}) = K - (1 - \hat{z})D - w\hat{z}^2 = K - \left(1 - \frac{D}{zw}\right)D - w\left(\frac{D}{2w}\right)^2 = K - D + \frac{D^2}{4w}$$

If x = x^i, then,

$$\hat{x}^i = \frac{\alpha(\hat{z})}{2c} = \frac{1}{2c}\left[K - D + \frac{D^2}{4w}\right]$$

and

$$\hat{x}^j = \hat{x}^i, \quad \text{for } j \neq i, \quad i, j = 1,2.$$

These values do yield the maximum of V; it is an interior maximum if the inequalities on D, K and w are strict.

As we have already said, the problem of coordination in its informational aspect may be viewed as that of computing the values of the three decision variables given the parameters, a and b, and subject to the conditions that limit the information processing capabilities of individuals.

We introduce the idea of an organizational Role- a collection of information processing functions that might be performed by an organizational subunit, such as a multiperson department, or by a single person, or by a free standing organizational unit, such as a firm.

We make two informational assumptions. First, that the parameters of P_1 and P_2 cannot be observed by the same organizational unit, whether a person or a group of people. Second, that the elementary operations available to individuals, and therefore to groups of them, are the binary operations of arithmetic together with the operation of raising a number to an arbitrary positive integer power. This last assumption is made only for convenience; nothing essential would change if it was omitted.

Given the assumption about the initial distribution of information about the parameters, there are at least two Roles, denoted 1,2. For i = 1,2, Role i observes θ^i (either a or b). For definiteness assume that Role 1 also observes K, w and c. Each of these Roles must employ at least one individual. The remaining information processing tasks may be assigned to either Role, or to additional Roles that might be introduced.

Two algorithms.

We begin with two algorithms obviously available for this problem. The first, Algorithm I, is to compute

$$\hat{z} = \frac{\sum_{j=1}^{q}\left(a_j - b_j\right)^2}{2w} \tag{5.1}$$

directly, and then to compute $\hat{x}^1$ and $\hat{x}^1$, as shown in the following modular network.

Under the assumption that the class of elementary operations consists of the binary operations of arithmetic together with the operation of raising a number to an arbitrary positive integer power, the modules of the network can have at most 2 input lines, i.e., the network is a (2,1)-network. Figure 1.1 shows the network for computing $\hat{z}$, and Figure 1.2 shows the network for computing $\hat{x}^1 = x^2$ from $\hat{z}$.

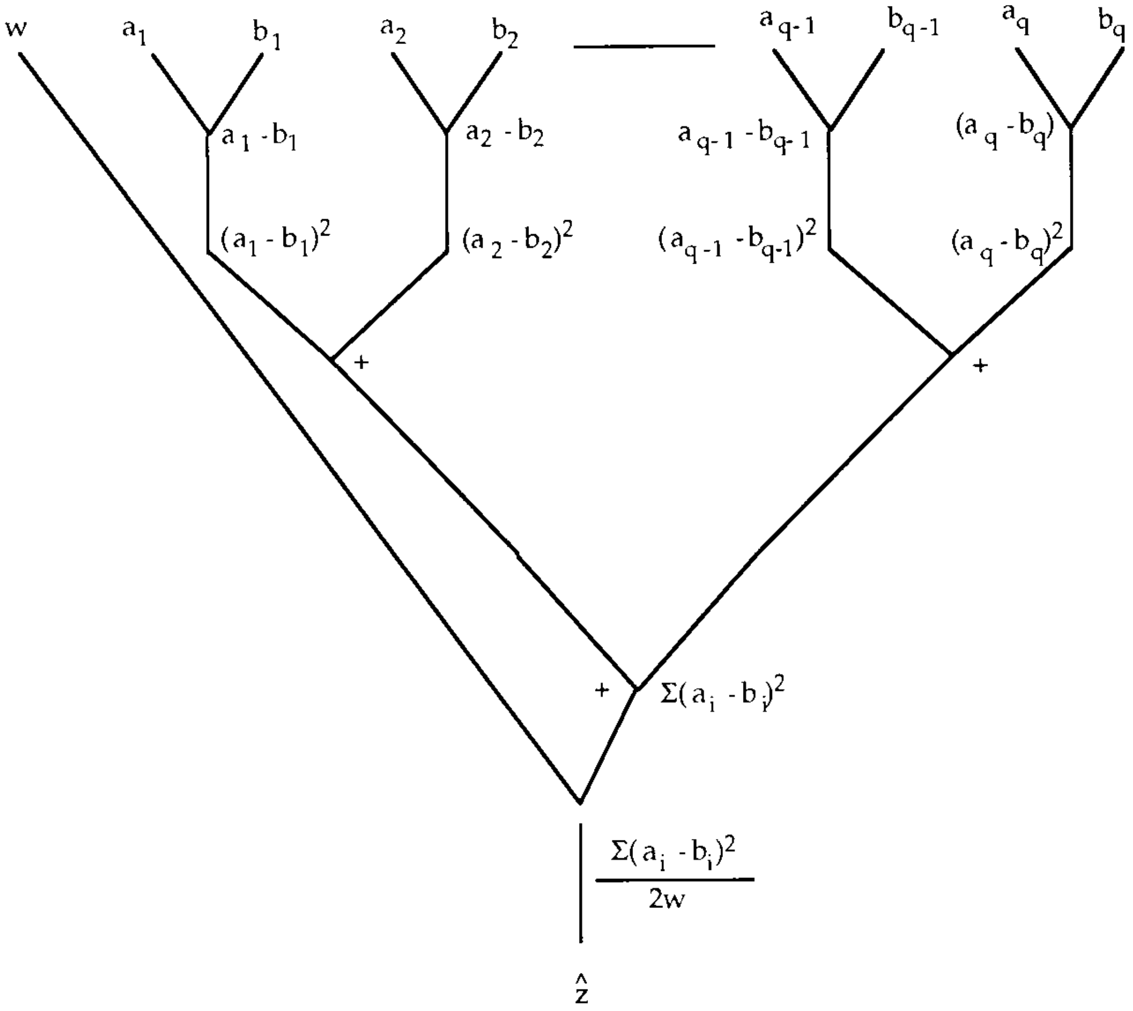

w
a_1
b_1
a_2
b_2
a_{q-1}
b_{q-1}
a_q
b_q
$a_1 - b_1$
$a_2 - b_2$
$a_{q-1} - b_{q-1}$
$(a_q - b_q)$
$(a_1 - b_1)^2$
$(a_2 - b_2)^2$
$(a_{q-1} - b_{q-1})^2$
$(a_q - b_q)^2$
+
+
+
$\Sigma(a_i - b_i)^2$
$\frac{\Sigma(a_i - b_i)^2}{2w}$
$\hat{z}$

Figure 1.1

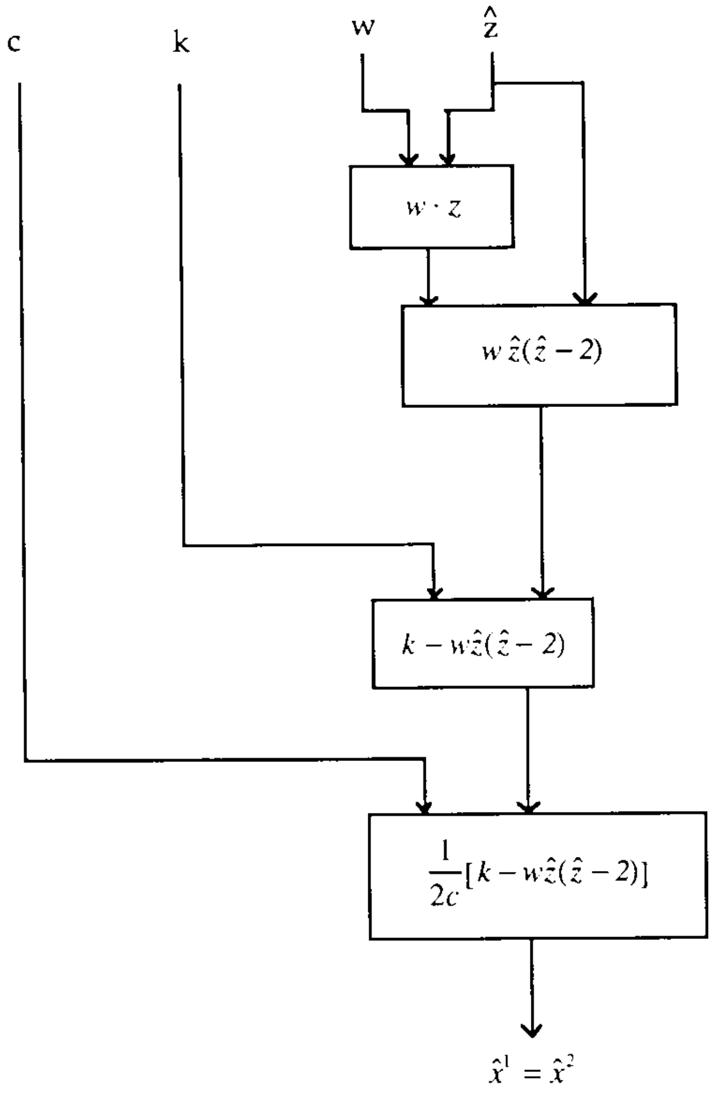


Figure 1.2

The second algorithm, Algorithm II results from writing the formula for $\hat{z}$ as follows.

$$\hat{z} = \frac{D}{2w} = \frac{\sum_{j=1}^{q}\left(a_j - b_j\right)^2}{2w} = \frac{\sum_{j=1}^{q}\left(a_j\right)^2 + \sum_{j=1}^{q}\left(b_j\right)^2 - 2\sum_{j=1}^{q} a_j b_j}{2w} = \frac{A(a) + B(b) - 2a \cdot b}{2w}.$$

The (2,1)-network shown in Figure 2.1 computes the values $\hat{z}$, $\hat{x}^1$ and $\hat{x}^2$ by computing A(a), B(b) and -2 $a \cdot b$ separately and combining them.

For each algorithm, the modules of the network must be assigned to individuals subject to the parallel constraint. Under the assumption that Role 1 observes a, K and w, the following conclusions emerge for Algorithm I.

(i) Every efficient assignment has the property that only Role 1 computes $\hat{z}$.

(ii) In every efficient assignment Role 2 communicates all its parameters b to Role 1.

(iii) Let $v_{sqr}(q,n_1,n_2)$ denote the minimum number of crosslinks required by an efficient assigned network that executes Algorithm I with n_i individuals employed in Role i, and where each process has q parameters. Then, for all q, n_1 and n_2,
$v_{sqr}(q,n_1,n_2) \geq$ q.

For Algorithm II, as in the case of Algorithm I, there are two symmetric equivalent schemes for computing $\hat{z}, \hat{x}^1, \hat{x}^2$, depending on who observes K, w and c. If, as in the case of Algorithm I, Role 1 observes a, K and w, then the efficient scheme is that Role 1 computes A(a), $\hat{z}$, $\hat{x}^1$ and possibly $\hat{x}^2$. Role 2 observes b, computes $B(b)$and possibly $\hat{x}^2$. The network shown in Figures 2.1, 2.2.1, 2.2.3 shows this scheme.

Since the assignment of the input nodes gives all input nodes labeled a_i to Role 1 and all those labeled b_i to Role 2, and the efficient graph assigns all other nodes to Role 1, the solution graph splits into two parts, one consisting of all the input nodes labeled b_i and the other consisting of all other nodes. This part of the solution EADAG has q crosslinks between the subgraph assigned to Role 2 and the one assigned to Role 1.

The graph shown in Figure 1.2 shows the algorithm for computing $\hat{x}^1$ (or $\hat{x}^2$) from $\hat{z}$ and $\hat{x}^2$ from $\hat{x}^1$ (they are equal) (or respectively $\hat{x}^1$ from $\hat{x}^2$). If this subgraph is assigned to Role 1, there is one additional crosslink, corresponding to the transmission of the value of $\hat{x}^2$ to Role 2. If the graph is assigned to Role 2, there are 2 additional crosslinks, corresponding to the transmission of $\hat{z}$ from Role 1 to Role 2 and to the transmission of $\hat{x}^1$ to Role 1.

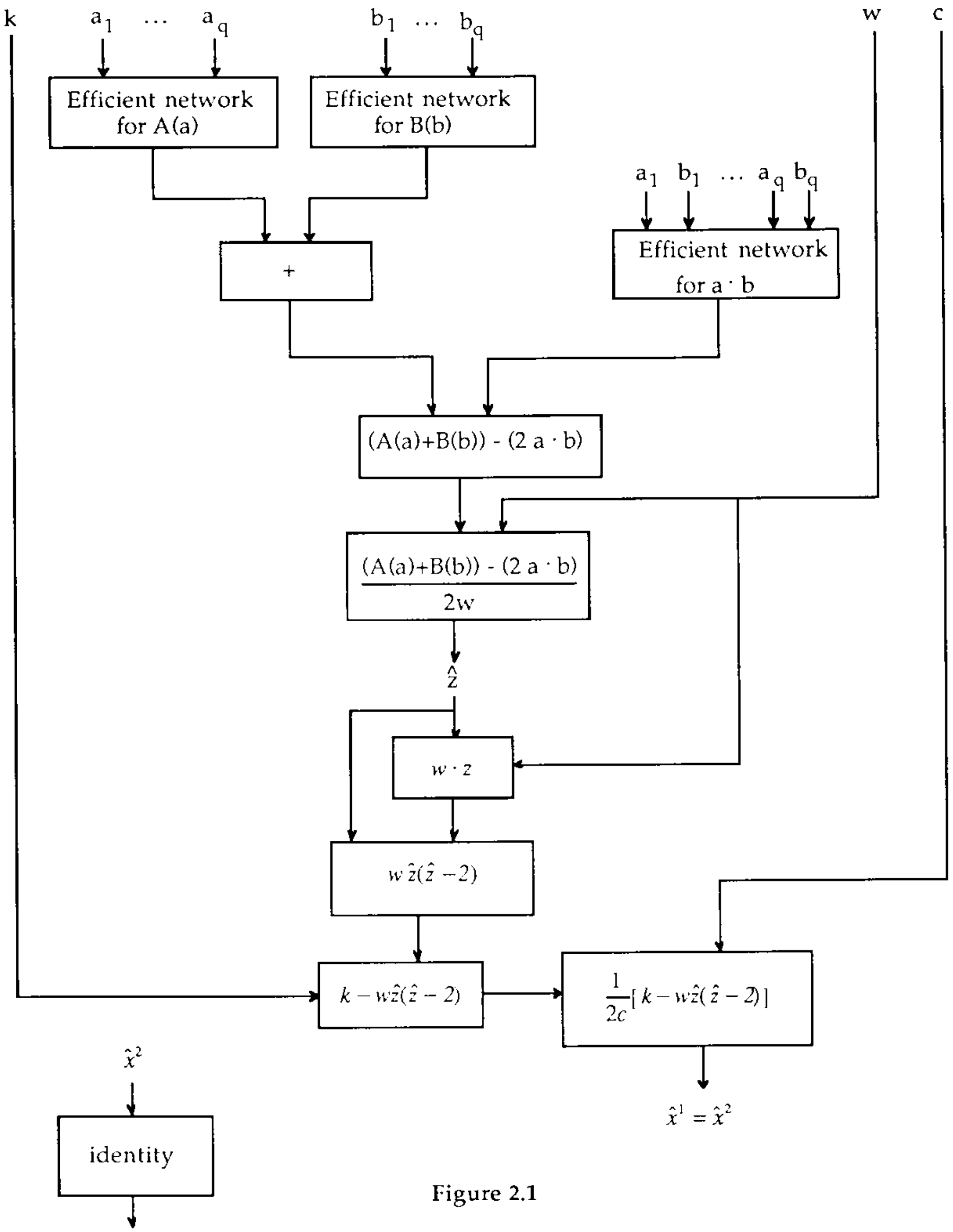

k
a1 ... aq
b1 ... bq
w
c
Efficient network for A(a)
Efficient network for B(b)
+
a1 b1 ... aq bq
Efficient network for a · b
(A(a)+B(b)) - (2 a · b)
(A(a)+B(b)) - (2 a · b) / 2w
ẑ
w · z
wẑ(ẑ − 2)
k − wẑ(ẑ − 2)
1/2c [k − wẑ(ẑ − 2)]
x̂¹ = x̂²
x̂²
identity

Figure 2.1

Efficient Network for A(a)

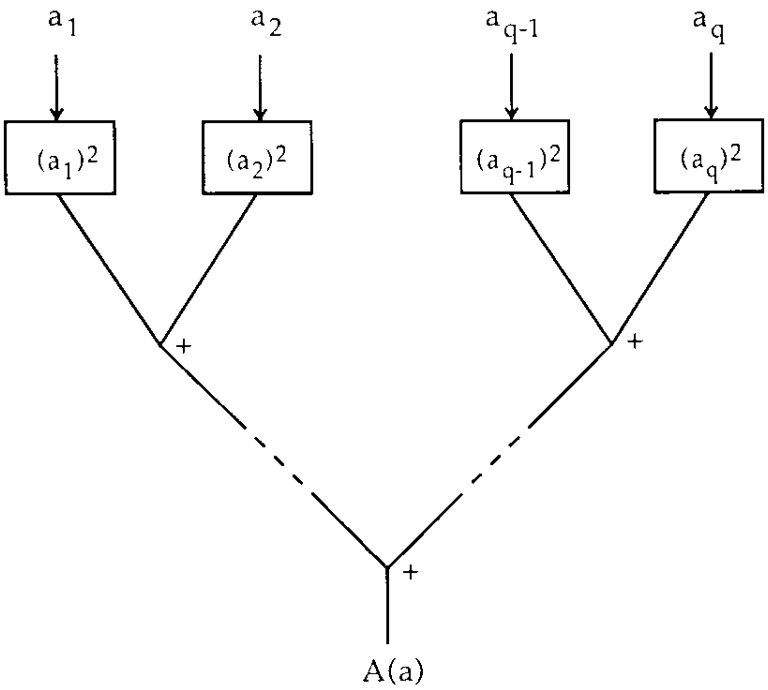


Figure 2.2.1

Efficient Network for B(b)

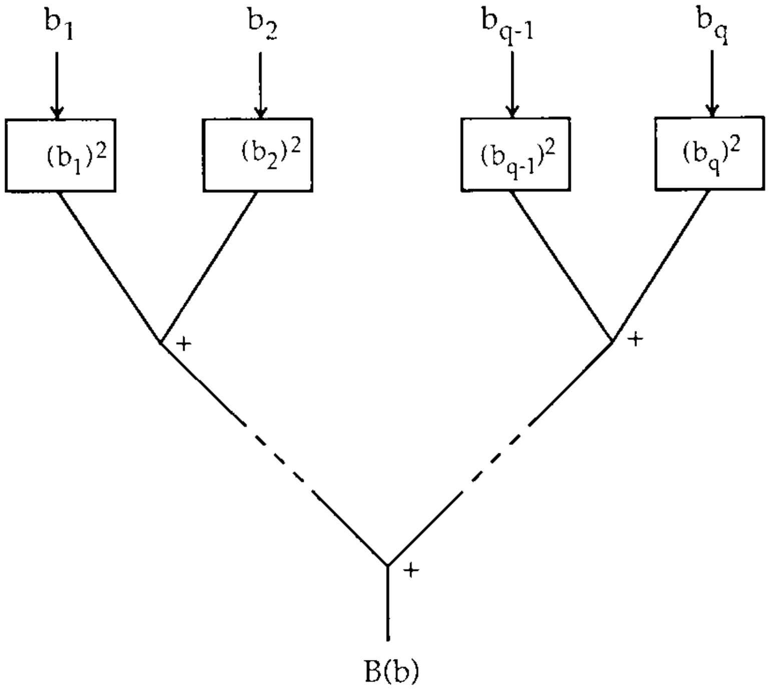


Figure 2.2.2

Efficient Network for 2 a · b

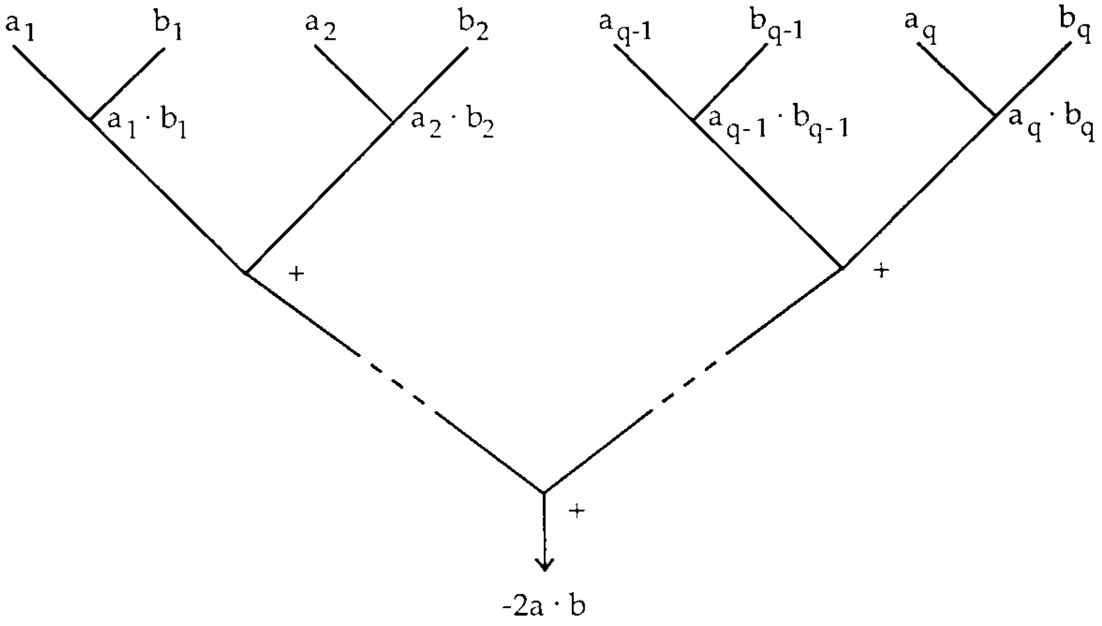


Figure 2.2.3

Analysis of the efficient assignments of modules in the case of Algorithm II yields the following conclusions.

(i) It is efficient for Role 1 to compute A(a) and for Role 2 to compute B(b). As in the case of Algorithm I, in every efficient assignment, only Role 1 computes $\hat{z}$.

(ii) In every efficient assignment, Role 2 communicates all the parameters b to Role 1.

(iii) Let $v_{ip}(q,n_1,n_2)$ denote the minimum number of crosslinks required to compute $\hat{z}$ by Algorithm II. Then, for all q, n_1 and n_2, $v_{ip}(q,n_1,n_2) \geq q+1$.

Let $\tau_z(q,n_1)$ denote the delay as a function of q and n_1 for an efficient network that computes the inner product. Similarly, let $v_1(q,n_1)$ and $\tau_1(q,n_1)$ denote the corresponding quantities for the networks that compute $\hat{x}^1$ from $\hat{z}$ and d, and let $v_2(q,n_2)$ and $\tau_2(q,n_2)$ denote the corresponding quantities for the computation of $\hat{x}_2$ from z and c, where n_2 is the number of individuals employed in Role 2.

Analysis of efficient networks for computing the inner product, which is what the computation of $\hat{z}$ entails, tells us that $v_z(q,n_1,n_2) = q$, while the delay, $\tau_z = \tau_z(q,n_1,n_2)$ varies with q and n_1. In scheme I, the overall time to compute all three decisions is $\tau(q,n_1,n_2) = \tau_1(q,n_1) = \tau_z(q,n_1) + 1$, which is the same as the overall time to compute $\hat{x}_2$ since $\tau_2(q,1) = 1$, and $\hat{x}_1$ and $\hat{x}_2$ are computed in parallel from $\hat{z}$, in one unit of time.

Role 2 might also compute $\hat{z}$, but this would be inefficient, since it would increase the number of crosslinks from q to 2q, and have the same delay.

In Example 1 any assignment of nodes to Role 1 and Role 2 results in at least q crosslinks, and exactly q crosslinks is attained only if all the nodes are assigned to Role 1 and none to Role 2, or the reverse. Therefore, as in the case of algorithm I, the number of crosslinks between the subgraph assigned to Role 1 and that assigned to Role 2 grows without bound.

One might speculate about the possibility that there is some other algorithm for computing the decision variables, especially $\hat{z}$, which might result in a less centralized organization. There is no such algorithm, a fact that will be discussed in Section 6 below.

Size of the firm in Example 1

The preceding analysis of the implications of the coordination requirement in Example I, expressed by the decision rule (or the goal function from which it is derived) tells us that production in this class of environments is best organized in one firm, for a class of environments where large values of q are possible. When production is organized in one unit, the cost per crosslink is α_2^*, which is strictly less than α_2, and the fixed cost is $\alpha_0 + \alpha_0'$ instead of α_0. When the class of environments is large, then it is possible that organization into one firm cannot work for all environments. The size of the firm or organization will be bounded if the net value function goes from positive to negative for some value of q. We next compare the cost of computing optimal decisions with the benefit of doing so, as functions of the number of parameters.

We write

$$V_q(\hat{x}^1, \hat{x}^2, \hat{z}, a, b)$$

for the value of output when the optimal decisions are taken and the vectors a and b have q components. This value is net of production costs, but not of the costs of computing the optimal decisions. The full net value is

$$(5.2) \qquad V_q(\hat{x}^1, \hat{x}^2, \hat{z}, a, b) - \overline{\chi}_q$$

when $\overline{\chi}_q$ is the cost of computation associated with a minimal EADAG. We show that the value function, the first term in (5.2)is bounded by a constant, independent of q. Therefore, if for some value of q the cost of computing the optimal decisions exceeds that constant, then it is not worthwhile to compute optimal decisions for a coordination problem with larger values of q. We begin by showing that there is a constant such that for all q, V_q is bounded by that constant.

Note that

$$V_q\left(\hat{x}_q^1,\hat{x}_q^2,\hat{z}_q,a,b\right)=\hat{x}_q\left(k-\left(1-\hat{z}_q\right)D_q-w\hat{z}_q^2\right)-c(\hat{x}_q)^2$$

where $\hat{x}_q^i$ is the optimal value of x^i when the parameter vectors are q -dimensional and similarly for $\hat{z}_q$ and D_q.

We know from the conditions assumed in the original optimization problem that

$$\frac{D_q}{2}\leq w\leq k,$$

which implies

$$D_q\leq 2w.$$

Now

$$\begin{aligned}k-(1-\hat{z}_q)D_q-w\,\hat{z}_q^2&=k-\left(1-\tfrac{D_q}{2w}\right)D_q-w\frac{D_q^2}{(2w)^2}\\&=k-D_q+\tfrac{D_q^2}{4w}<k+\frac{D_q^2}{4w}<k+w.\end{aligned}$$

Furthermore, from this and the formula for $\hat{x}_q$, it is immediately evident that

$$\hat{x}_q = \frac{1}{2c}\left[k - D_q + \frac{D^2}{4w}\right] \leq \frac{1}{2c}(k+w)$$

Therefore,

$$V_q\left(\hat{x}_q^1, \hat{x}_q^2, \hat{z}_q, a, b\right) \leq \hat{x}_q\left(k - (1-\hat{z}_q)D_q - w\ \hat{z}_q^2\right) \leq (\tfrac{1}{2c}+1)\ k + w \equiv \text{ constant.}$$

We turn now to the analysis of the growth of cost as a function of the number, q, of parameters. It is sufficient to show that the cost of computing the inner product is unbounded in q. The computation of A(a) and B(b) can only add costs. For this we must analyze the other determinants of cost, namely delay, and the number of individuals (processors) used, $n = n_1 + n_2$. This is a little more complicated than the preceding analysis.

By assigning enough individuals to perform Roles 1 and 2 it is possible in Example 1 to achieve the minimum possible delay, namely $t^*(q) = q$ units of time.

More generally, let $\mathcal{E}(q,n_1,n_2) = \{(v,\tau) : (v,\tau) \text{ is efficient given } q,n_1,n_2\}$. Let

$$\underline{v}(q,n_1,n_2) = \min_{\varepsilon(q,n_1,n_2)} \{v\}$$

and let

$$\bar{\tau}(q,n_1,n_2)=\max_{\varepsilon(q,n_1,n_2)}\{v\}.$$

If $(v,\tau)\in\mathcal{E}(q,n_1,n_2)$ and $v=\underline{v}(q,n_1,n_2)$, then $\tau=\bar{\tau}(q,n_1,n_2)>t$ for all t for which there exists v such that $(v,t)\in\mathcal{E}(q,n_1,n_2)$. That is, if (v,τ) is efficient for (q,n_1, n_2), and if v is the minimum number of crosslinks among efficient assignments, then τ is the maximum delay among efficient assignments.

The following properties hold in Example 1.

(5.3)

For $q\geq 2,\ \ \underline{v}(q+1,n_1,n_2)=\underline{v}(q+1,1,1),=\underline{v}(q,1,1)+1=\underline{v}(q,n_1,n_2)+1,$ where $\underline{v}(2,1,1)=2$

(5.4)

For $q\geq 2,\ \ \bar{\tau}(q+1,n_1,n_2)=\bar{\tau}(q+1,1,1)=\bar{\tau}(q,1,1)+2=\bar{\tau}(q,n_1,n_2)+2,$ where $\bar{\tau}(2,1,1)=3$

Let $\tau^*(q)$ denote the minimum delay for computing $\hat{z}$ in Example 1 by a modular network without regard to the parallel constraint. Then

(5.5) $\quad \tau^*(q)=INT\left[\log_2 q\right]+1,\ q\geq 2$

where $INT[x]$ denotes the smallest integer larger than x. It follows that, for p a (nonzero) natural number, if $2^{p-1} < q \le 2^p$, then $\tau^*(q) = p+1$. Furthermore,

(5.6) The number of persons required to carry out the Roles 1 and 2 while attaining the minimum delay satisfies the following conditions. For $n = n_1 + n_2$,

(5.7) $2^{p-1} < q \le 2^p \Rightarrow 2^{p-2} < n < 2^p$,

and

(5.8) $q = 2^p \Rightarrow n = q$ and $n_1 = n_2$.

Thus, n (q) does not grow monotonically with q, but, for $n(q) = (n_1 + n_2)(q)$ such that $\tau(q, n_1, n_2) = \tau^*(q)$, $q \in (2^{p-1}, 2^p] \Rightarrow 2^{p-2} < n(q)$. And further, since for $q = 2^p$, $n(q) = q$, $n(q)$ is bounded above by q.

Thus, both n(q) and $\tau^*(q)$ grow roughly as $\log_2 q$.

The number of crosslinks grows linearly with q. It is at least q when $q = 2^p$, and at such values of q, $n_1 = n_2$ and $n_1 + n_2 = n = q$. Then, at such values of q, the number of crosslinks is

$$\nu(q, n_1, n_2) = \nu\left(2^p, \frac{q}{2}, \frac{q}{2}\right) = \nu(2^p, 2^{p-1}, 2^{p-1}) = 2^p + 1.$$

In addition to the number of crosslinks, and the minimum delay as functions of q, we also need the total number of arcs $C(q)$ in a $(2,1)$ -network that computes the inner product of q - dimensional vectors. For even values of q there is a simple formula for $C(q)$. For $q = 2n$, where $n = 2,3,\cdots$

$$(5.9) \quad C(q) = C(2) + \left(\frac{q}{2} - 1\right)8,$$

where $C(2) = 6$.

Since 2^p is even for p a positive integer, (5.9) is valid for $q = 2^p$.

The formulas for $v(q,n_1,n_2)$ and for the delay $\tau(q,n_1,n_2)$ together with the formula (5.9) for $C(q)$ give a convenient indicator how the cost of computing grows with the number of parameters. Because the cost of a unit of delay is assumed (Section 3; (3.2)) to be at least as large as the cost of hiring another computer (person) it is useful to look at the growth of cost when enough people are hired for each q to achieve the minimum delay. Recall that $\tau*(q)$ is the minimum possible delay, and that it is equal to $\tau*(q, n*(q))$. For this analysis we may ignore the constant term. Then,

$$\overline{\chi}(q) = \alpha_1 C + c_2(q)(\alpha_2 - \alpha_1) + \alpha_3 \tau*(q) + \alpha_4 n^*(q).$$

The following formulas, are all valid when $q = 2^p$

$$C(q) = 6 + \left(\frac{q}{2} - 1\right)8 \quad \text{when } q \text{ is even,}$$

$$c_2(q) = v(q, n_1, n_2) = q \qquad \text{for all } q,$$

and $\tau^*(q) = p + 1$ when $\mathrm{q} = 2^{\mathrm{P}}$, (hence $\tau^*(q) \cong 1 + \log_2 q$,)

and $n^*(q) = q$.

When $q = 2^p$, writing $\hat{\chi}(p) \equiv \overline{\chi}(2^p)$ and substituting from the four formulas above, yields

$$\hat{\chi}(p) = \alpha_1\left(6 + \left(\frac{2^p}{2} - 1\right)8\right) + (\alpha_2 - \alpha_1)2^p + \alpha_3(p+1) + 2^p$$

$$= 2^P\left((\alpha_2 - \alpha_1) + 4\alpha_1 + \alpha_4\right) + \alpha_3(p+1) - 2\alpha_1$$

Now, because the cost of a crosslink within a single firm is α_2^*, the term $(a_2 - \alpha_1)$ is replaced by $(\alpha_2^* - \alpha_1)$, which from ($) is nonnegative. Hence, the cost grows exponentially in p, which means that as q increases, the cost of computing exceeds all bounds, although as shown above not all determinants of cost increase monotonically in the intervals of q between successive values of 2^p. This remains true even if it is the case that within a firm there is no difference in cost between communication via crosslinks and via selflinks, i.e., when $\alpha_2^* = \alpha_1$.

The total cost of processors (persons or other units capable of carrying out an elementary operation) can be reduced by trading off increased delay for reduction in the number of processors. The relations (5.3) and (5.4) of this section tell us the extent to which such a tradeoff can be made, since the maximum delay is achieved when there is only one processor of each type, i.e. when $n = n_1 + n_2 = 1 + 1 = 2$. In that case the delay is roughly equal to half the number of vertices in the graph. The number of vertices, $N(q)$ in the graph of a (2,1) - network that computes the q-dimensional inner product is

$$N(q) = 1 + (q - 1) \cdot 2$$

and

$$\begin{aligned} \bar{\tau}(q,1,1) &= INT\left[\frac{1}{2}N(q)\right] = INT\left[\frac{1 + 2(q-1)}{2}\right] = INT\left[\frac{1}{2} + q - 1\right] \\ &= q. \end{aligned}$$

Thus, trading off processors for increases in delay can reduce the cost of processors to $2\alpha_4$ independently of q, but the cost of the corresponding delay is unbounded, since for $q = 2^p$

$$\bar{\tau}(q,1,1) = 2^p.$$

The effect of the tradeoff on cost is given by

$$\alpha_3(\tau*(q,n*) - \bar{\tau}(q,1,1)) + \alpha_4(n*(q) - 2),$$

since the differences in the first two terms of the cost function add to zero.

Substituting $q = 2^p$ yields

$$\alpha_3(p+1-2^p)+\alpha_4(2^p-2)$$
$$=2^p(\alpha_3-\alpha_4)+\alpha_3(p-1)-2\alpha_4$$

Recalling assumption (3.1) which is that $\alpha_3 \ge \alpha_4$,

If $\alpha_3 = \alpha_4$, then

$$\alpha_3(p-1)-2\alpha_4 = \alpha_3(p-1)-2\alpha_3 = \alpha_3(p-3)$$

which is positive if $p>3$, (or $q>8$).

If $\alpha_3 > \alpha_4$. Then,

$$2^p(\alpha_3-\alpha_4)+\alpha_3(p-1)-2\alpha_4>0$$

is equivalent to

(5.10) $2^p(\alpha_3-\alpha_4)>2\alpha_4-\alpha_3(p-1)$

If $p=2$; the inequality reduces to

$$\alpha_3 > \tfrac{3}{5}\alpha_4,$$

which is satisfied in view of the assumption that $\alpha_3 > \alpha_4$.

(If $p=1$, then the condition $\alpha_3 > 2\alpha_4$ is equivalent to the inequality (5.10)).

Thus, $p \geqq 3$ (or $q \geqq 8$) is sufficient for the inequality (5.10) to be positive, or equivalently for the tradeoff to increase cost.

One might think that the decision rule in Example I is special in the sense of being rare, but this is far from the case; functions like that are ubiquitous in the space of smooth functions. William's genericity theorem, discussed in Section 6, tells us that there is an open dense set of functions which require that all but one of the Roles transmit all their parameters to the remaining Role. If we take the inner product as the prototypical representative of this class of decision rules, we see that it represents a maximal requirement of coordination and that efficient organizations for computing it are fully centralized. This fact may go some way toward explaining the informational basis for the prevalence of centralized organization of firms.

Example 2. (Abelson (1980))

We consider a second example involving the same parameters and decision variables, but a different value function, W instead of V, where

$$W\left(x^1, x^2, z; a, b\right) = x^1 x^2 \left(z \left(\sum_{j=1}^{q} a_1^j b_j + \sum_{i=1}^{q} a_i b_1^i \right) - \frac{z^2}{2} \right) - cx^1 - dx^2 .$$

Let

$$\beta(z) = \left(z \left(\sum_{j=1}^{q} a_1^j b_j + \sum_{i=1}^{q} a_i b_1^i \right) - \frac{z^2}{2} \right),$$

and let

$$k = \left(\sum_{j=1}^{q} a_1^j b_j + \sum_{i=1}^{q} a_i b_1^i \right).$$

Then the first order conditions for maximizing W can be written as

$$\begin{aligned} &x^1 x^2 (k - z) = 0 \\ &x^2 \left(zk - \frac{z^2}{2} \right) - c = 0 \\ &x^1 \left(zk - \frac{z^2}{2} \right) - d = 0, \end{aligned}$$

which yield

$$\begin{aligned} &\hat{z} = k = \left(\sum_{j=1}^{q} a_1^j b_j + \sum_{i=1}^{q} a_i b_1^i \right) \\ &\hat{x}^1 = \frac{2c}{\hat{z}^2} \\ &\hat{x}^2 = \frac{2d}{\hat{z}^2} \end{aligned}$$

An efficient network for computing $\hat{z}$ in this example is shown in Fig. 3.3. In this scheme Role 1 observes the parameters a,c, and computes $\sum_{j=2}^{q} a_j b_1^j \equiv A$, the final term

A + B, and $\hat{x}^1$; Role 2 observes the parameters b,d, and computes $\sum_{i=1}^{q} a_1^i b_i \equiv B$, and $\hat{x}^2$.

Figure 3.1 and 3.2 show networks for A(a) and B(b) respectively.

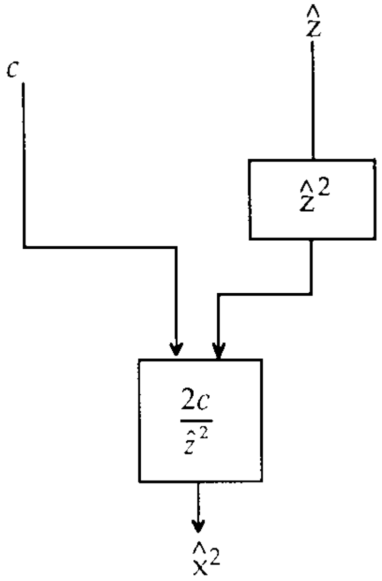


Figure 3.1

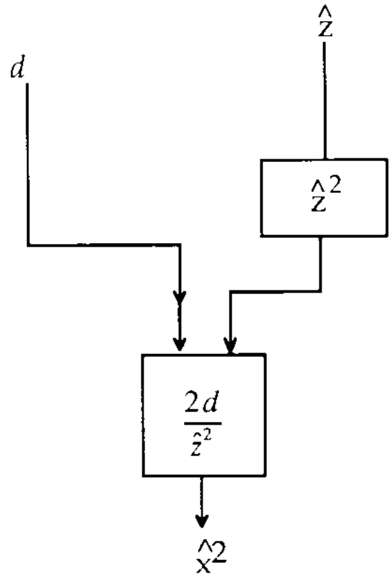


Figure 3.2

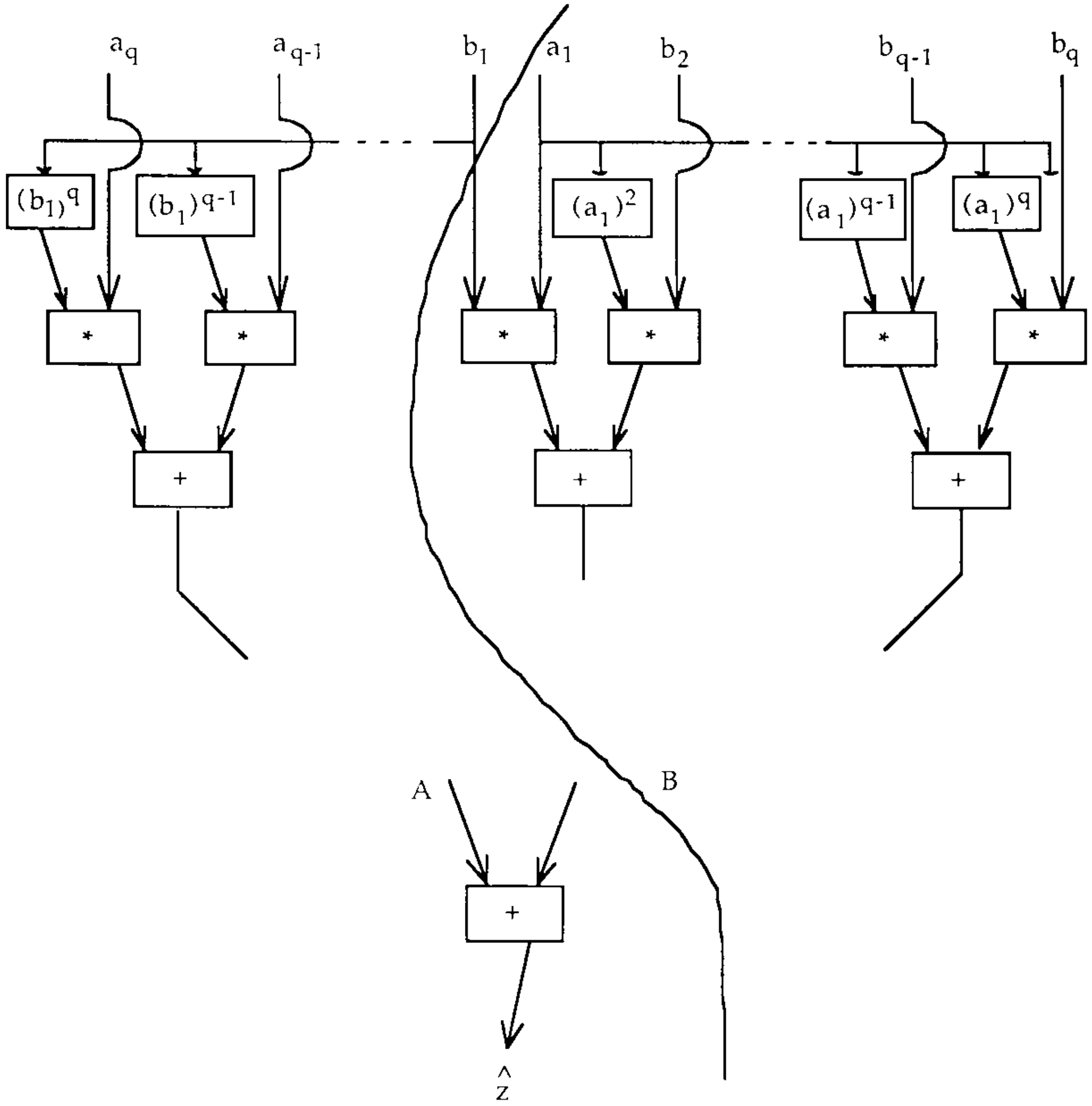


Figure 3.3

In this scheme Role 1 computes the term $\sum_{j=2}^{q} a_j b_1^j = A$, and Role 2 computes $\sum_{i=1}^{q} a_1^i b_i = B$. The final term A+B is computed by Role 1. The computation of $\hat{z}$ requires exactly 3 crosslinks, since Role 2 transmits the value of b_1, and eventually B to Role 1, who transmits the value of a_1 to Role 2. Thus,

$$v_{\hat{z}}^2(q,n_1,n_2) = 3.$$

The computation of $\hat{x}^1$requires no additional crosslinks, while that of $\hat{x}^2$requires 1, either if Role 2 computes it, because Role 2 needs the value of $\hat{z}$ to compute $\hat{x}^2$, or if Role 1 computes it and transmits the result to Role 2. Finally, if process P_3 is controlled by someone other than Role 1, the value of $\hat{z}$ must be transmitted to that Role, entailing one more crosslink. Thus,

$$v^2(q,n_1,n_2) = v^2_{\hat{z},\hat{x}^1,\hat{x}^2}(q,n_1,n_2) = 4, \text{ or } 5.$$

And, letting

$$\tau_{\hat{z}}^2(q,n_1,n_2)$$

be the time required to compute $\hat{z}$ when n_i individuals belong to Role i, i = 1,2, it follows that the total time required to compute all the decision variables is

$$\tau_{\hat{z}}^2,\hat{x}^1,\hat{x}^2(q,n_1,n_2) = \tau_{\hat{z}}^2(q,n_1,n_2) + 1\ (or\ 2)$$

depending on whether the computation of $\hat{x}^1$and $\hat{x}^2$requires 1 or 2 units of time.

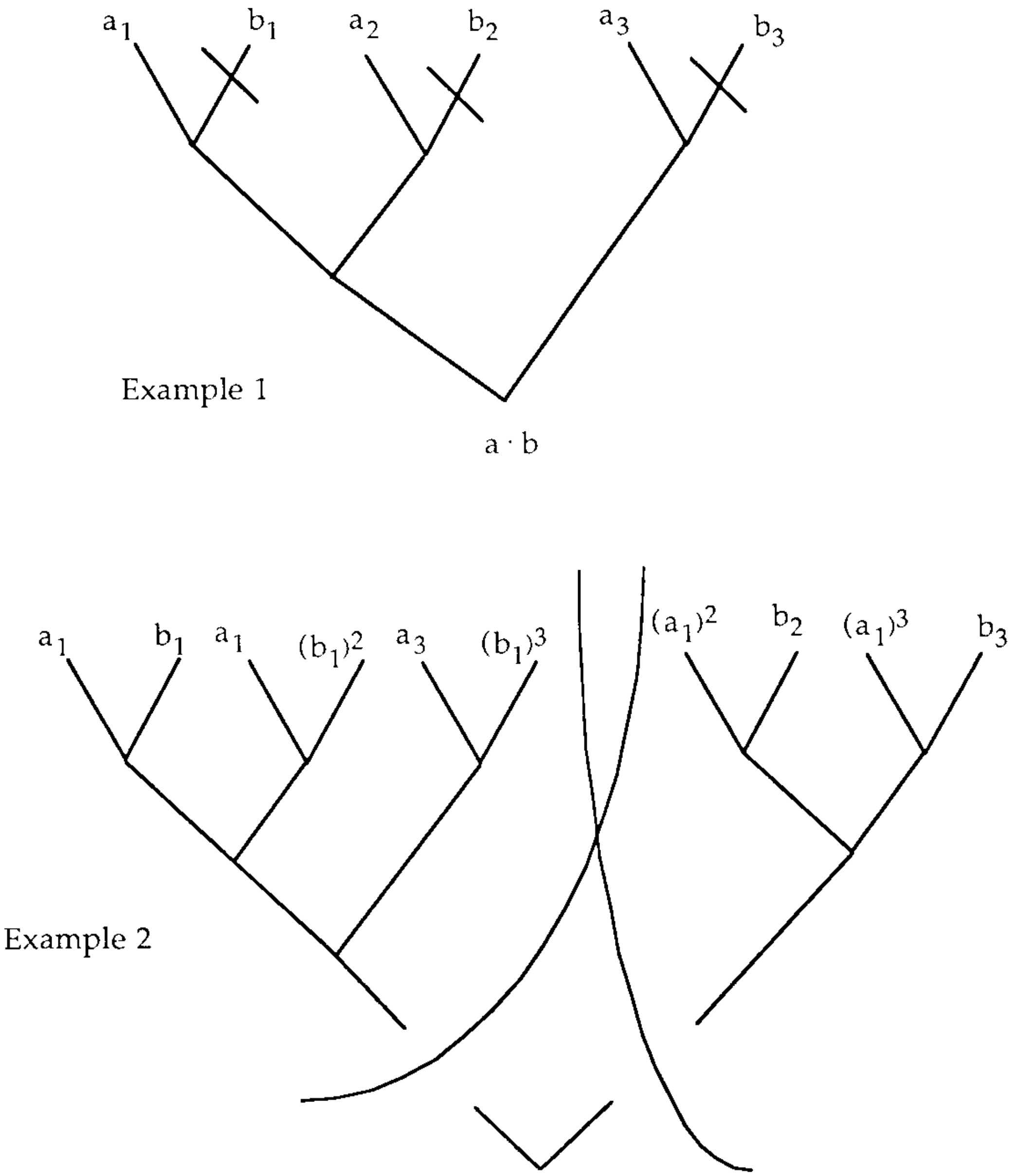
A second scheme simply interchanging Role 2 and Role 1 in the sense that Role 2 computes $\hat{z}$ instead of Role 1 leads to symmetric results.

Comparison of Examples 1 and 2

In Example 2 the number of crosslinks between Role 1 and Role 2 is independent of the value of q and of n_1 or n_2. Thus, no matter how many more parameters are needed to specify the technology, coordination requires only that the organizational unit corresponding to one of the Roles send one real number to the other, and the other Role send two numbers to the first, one to compute the decision variable $\hat{z}$, and one more to compute the remaining decision variables. Internal coordination of each of the units corresponding to Roles 1 and 2 is more complex than is the coordination between them, and involves internal communication of the values of many parameters. The number of individuals n_i who make up Role i may vary with the number of parameters in order to reduce delay. The amount of internal computing and communication will grow with q, but the communication between the units carrying out Roles 1 and 2 will be the same for every value of q. To an outside observer, the units corresponding to Roles 1 and 2 will appear functionally the same for all environments in the class generated by different values of q. One the other hand, in Example 1 the communication between units corresponding to Roles 1 and 2 grows with q, and is hardly distinguishable from the communication within the units.

Figure 4 presents in a graphic way the different structures that emerge in the two examples when q = 3. In Example 1 Roles 1 and 2 form what is essentially one unit; the communication that Role 1 has with itself is not very different from what it has with Role

2. In Example 2 the graph splits into two units, as the alternative cuts that partition the graph show. Communication between these units remains the same as q grows.



Example 1

Example 2

Figure 4

Example 3

Consider an environment given by q-dimensional parameter vectors a and b, as in Examples 1 and 2, but with a different function specifying optimal decisions. We begin with the case of two parameters, i.e., $q = 2$ and take the function to be

$$F(a,b) = \frac{b_1 - a_1}{a_2 - b_2}.$$

We suppose that, as in Examples 1 and 2, a and b cannot both be observed by the same Role. Therefore there will be at least two Roles in any organization for coordinating action in this class of environments.

Before proceeding further we note that the function F represents a class of functions for which the structure of organization would be the same. We know for instance from application of the Method of Rectangles (Hurwicz and Reiter [8]) that F can be written as the composition of the function, denoted ϕ(a,b), whose value at (a,b) is the solution of the equation system

$$(5.11) \qquad \begin{array}{l} m_1 - a_2 m_2 - a_1 = 0 \\ m_1 - b_2 m_2 - b_1 = 0 \end{array},$$

followed by the function

$$h(m_1, m_2) = m_2.$$

Solving the system (5.11) we obtain

$$\hat{m}_1 = \frac{\begin{vmatrix} a_1 & -a_2 \\ b_1 & -b_2 \end{vmatrix}}{\begin{vmatrix} 1 & -a_2 \\ 1 & -b_2 \end{vmatrix}} = \frac{a_2 b_1 - a_1 b_2}{a_2 - b_2}$$

$$\hat{m}_2 = \frac{\begin{vmatrix} 1 & a_1 \\ 1 & b_1 \end{vmatrix}}{\begin{vmatrix} 1 & -a_2 \\ 1 & -b_2 \end{vmatrix}} = \frac{b_1 - a_1}{a_2 - b_2},$$

and,

$$h \circ \phi(a,b) = h(\hat{m}_1, \hat{m}_2) = \hat{m}_2 = \frac{b_1 - a_1}{a_2 - b_2} = F(a,b).$$

The function F is representative of functions F*(a,b) which can be written as the composition of a function h* with ϕ*, where $\phi^*(\alpha_1(a), \alpha_2(a), \beta_1(b), \beta_2(b)) = (\tilde{m}_1, \tilde{m}_2)$, gives the solution $(\tilde{m}_1, \tilde{m}_2)$ of equations

$$(5.12) \qquad \begin{aligned} m_1 - \alpha_2(a) m_2 - \alpha_1(a) = 0 \\ m_1 - \beta_2(b) m_2 - \beta_1(b) = 0, \end{aligned}$$

where the coefficients are given functions of a and b respectively, the dimension of a (and b) is $q \geq 2$, and h* is an arbitrary (regular) function of (m_1, m_2)).

We can see from (5.11) or (5.12) that it is possible for the solution to be computed by algorithms such that communication between the two Roles is independent of q, the dimension of a and b. We consider two types of algorithms, direct and iterative. Among the direct algorithms for F there are two with the following efficient assigned networks.

1) The individual filling Role 1 computes $\alpha_1(a)$ and $\alpha_2(a)$ and transmits them to the individual filling Role 2, who computes everything else. This results in 2 crosslinks and a delay of 3 units of time, when there is just one individual in each Role, ($n_i = 1$, for $i=1,2$). The graph for this is Figure 5.2. (Figure 5.1 shows the algorithm before assignment.)

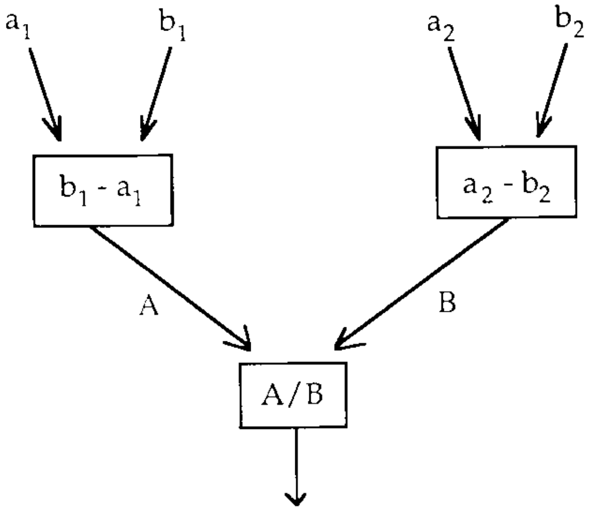


Figure 5.1

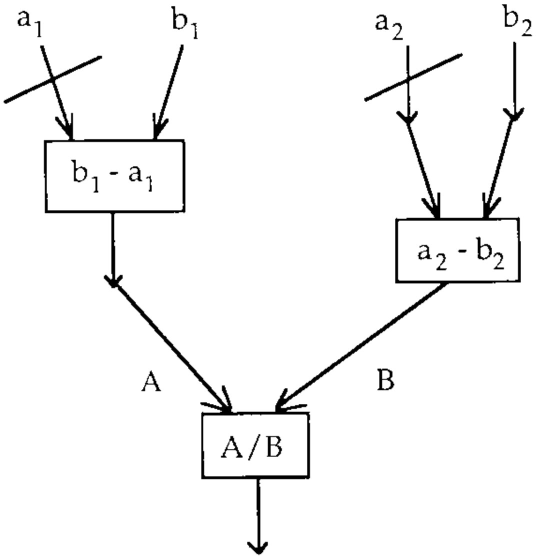


Figure 5.2

2) The individual who plays Role 1 sends a_2 to the one who plays Role 2; Role 2 sends b_1 to Role 1, who computes, b_1- a_1 while Role 2 computes a_2-b_2. This saves a unit of time, but costs a crosslink, because either a_2 - b_2 must be sent from Role 2 to Role 1 or b_1- a_1 must be sent from Role 1 to Role 2 in order to perform the final division. The assigned network for this procedure is shown in Figure 5.3.

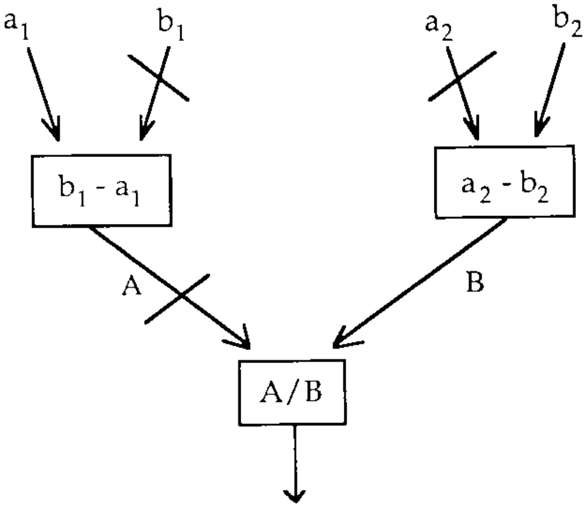


Figure 5.3

Both these assigned graphs also apply to the computation of F*, when a_i is replaced by α_i and b_i by β_i i=1,2, and the networks for computing α_i from a_i and β_i from b_i are included. Although the delays of the internal computations of the α's and β's increase with increases in q, the number of crosslinks does not. Thus, the number of crosslinks is independent of q, the number of parameters.

Both of these efficient assigned networks represent organization into two separate units. The example includes a situation which is generally regarded as the prototype of decentralized organization, namely, two independent units whose actions are coordinated by prices. This becomes even clearer when we consider a related iterative computation of F* , rather than the direct computations just presented.

The iterative process involves three Roles; Role 1 observes a, Role 2 observes b, and Role 3 observes no parameter directly, but can communicate with Roles 1 and 2. The three Roles together compute the value of F* by a convergent iterative process specified as follows.

(i) At step t Role 3 computes $\overline{m}(t) = (\overline{m}_1(t), \overline{m}_2(t))$ according to the rule

(6.0) for t= 0, $\overline{m}(0) = \overline{m}_0$, and for t > 0 $\overline{m}(t) = \frac{1}{2}\left(m^1(t) + m^2(t)\right)$;

(ii) At step t for i = 1,2 Role i computes $m^i(t+1) = \left(m_1^i(t+1), m_2^i(t+1)\right)$according to the following rules:

$$m_1^i(t+1) = \frac{-1}{(\alpha_2)^2+1}\left((\alpha_2)^2\overline{m}_1(t) + \alpha_2\overline{m}_2(t) + \alpha_1\right) \tag{6.1}$$

$$m_2^i(t+1) = \frac{-1}{(\alpha_2)^2+1}\left(\alpha_2\overline{m}_1(t) + \alpha_2\overline{m}_2(t) - \alpha_1\alpha_2\right) \tag{6.2}$$

$$m_1^2(t+1) = \frac{-1}{(\beta_2)^2+1}\left((\beta_2)^2\overline{m}_1(t) + \beta_2\overline{m}_2(t) + \beta_1\right) \tag{6.3}$$

$$m_2^2(t+1) = \frac{-1}{(\beta_2)^2+1}\left(\beta_2\overline{m}_1(t) + \beta_2\overline{m}_2(t) - \beta_1\beta_2\right). \tag{6.4}$$

It has been shown [Reiter (1979)] that

$$\lim_{t\to\infty} \overline{m}(t) = \lim_{t\to\infty} m^1(t) = \lim_{t\to\infty} m^2(t) = (\tilde{m}_1, \tilde{m}_2) = \phi(a,b)$$

This process is a classical tatonnement, in which the first equation of (5.1) (or of (5.2)) is the equilibrium condition of Role 1 and the second of Role 2. Role 3 acts as the 'market institution,' receiving the analogue of 'excess demands' from the other two Roles and replying with signals that play the role of tentative prices. The number of signals per iteration is larger than in the classical market demand adjustment process where price is assumed to adjust to excess demand. This is unavoidable because that price adjustment process is not (locally) stable for all environments for which the equilibrium exists, in this case, for which $\alpha_2^2 - \beta_2^2 \neq 0$ (Jordan [9]).

The obvious network for Role 1's computation is shown in Figure 6.1.

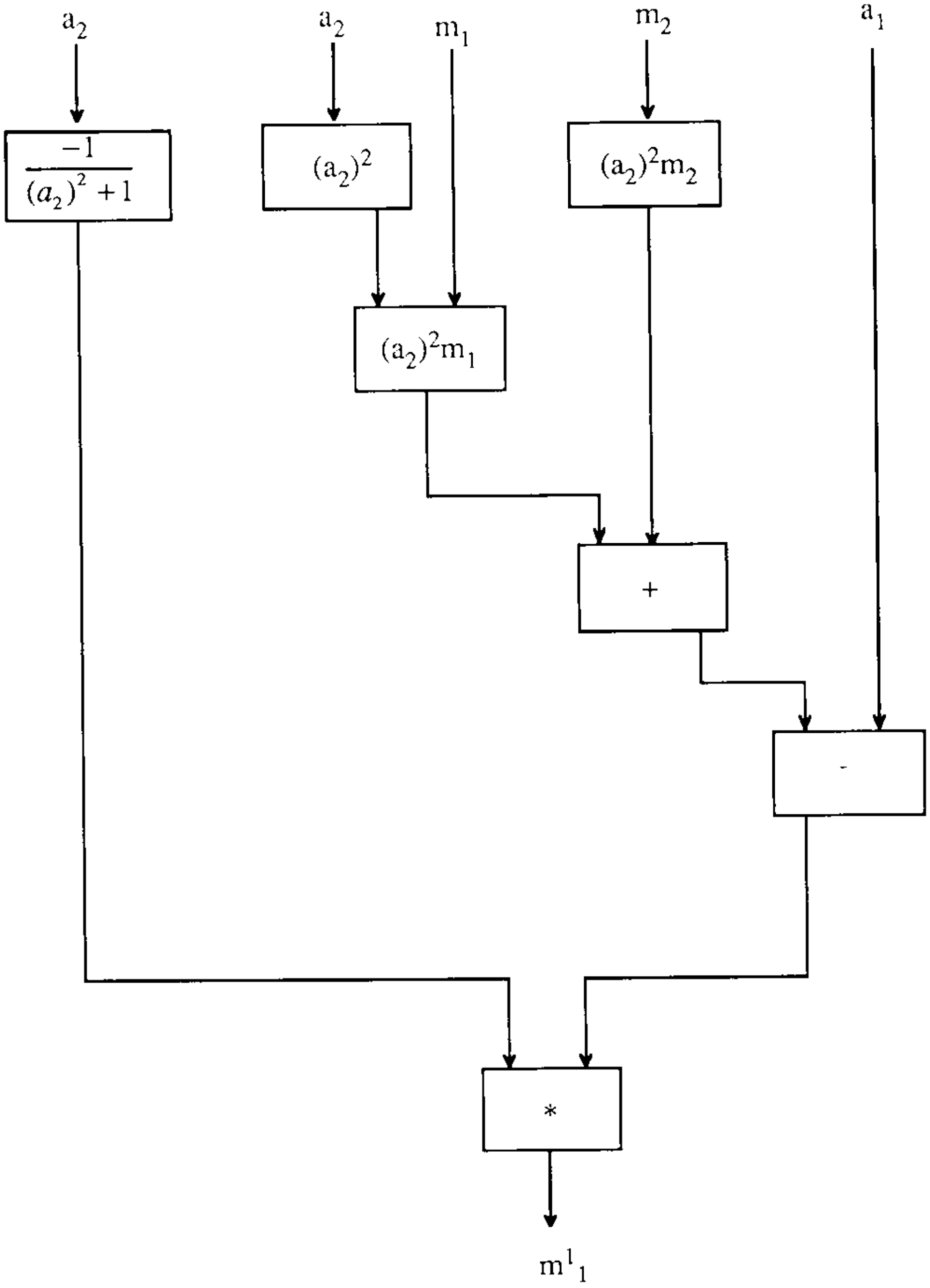


Figure 6.1

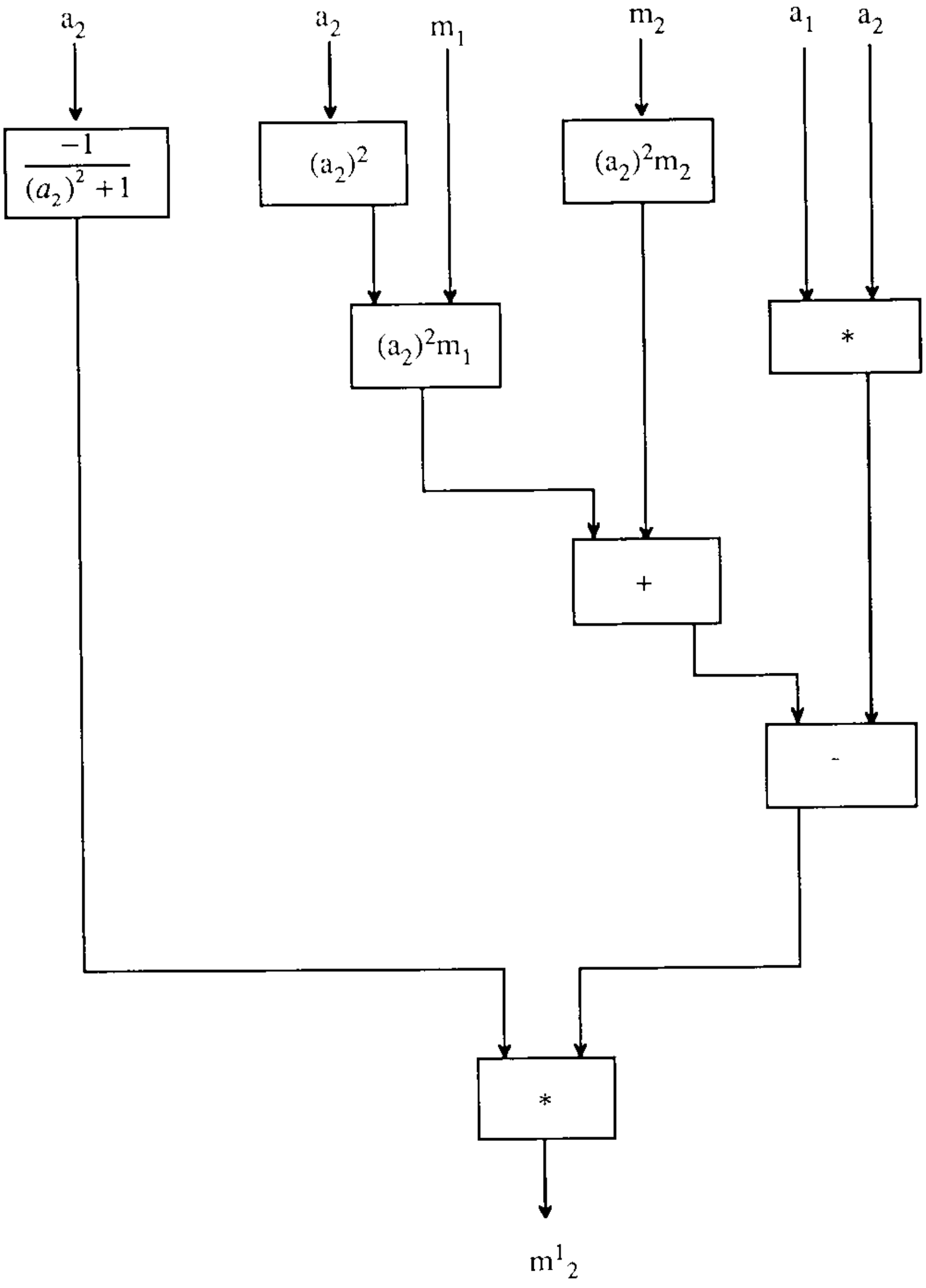


Figure 6.2

The same networks apply to Role 2 with β in place of α.

The networks for Role 3 are

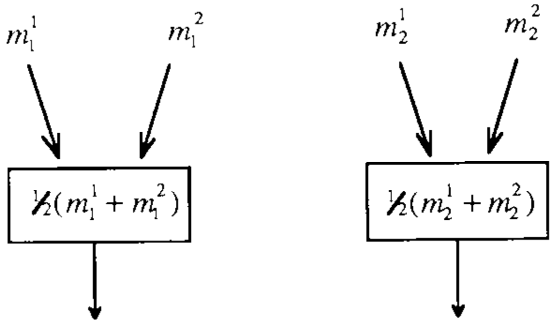


Figure 6.3

When each Role has just one individual in it, these networks show that the delay required for one iteration is 16 units of time. The number of crosslinks per iteration is 6. The number of crosslinks depends only on α and β and not on a and b.

Section 6. Some general results useful for analyzing external communication

Communication complexity

Example 1 shows that it is possible given the initial distribution of information to compute the value of $\hat{z}$ while transmitting q variables from one Role to another. Is there another way to compute $\hat{z}$ that requires transmission of fewer than q variables? The answer to this question is obtained from several related results.

First, Abelson [1] addressed the question of how much communication is required to compute a function F when knowledge of the values of its arguments is distributed among processors. Abelson's result is most easily presented when there are two processors P_1 and P_2. He considered real-valued functions of n + m real variables, where n variables are in the memory of processor P_1 and m are in the memory of P_2. A lower bound on the number of variables whose values must be transmitted between the two processors in a multistage computation, when communication may be in both directions, is given by the rank of the Hessian matrix of F. More precisely, let $F : R^n \times R^m \to R$, let $A \times B$ be a neighborhood of a point $(\bar{a},\bar{b})$ in $R^n \times R^m$. If F can be computed on $A \times B$ with (two-way) communication of k real variables between the processors, then the matrix

$$\mathrm{H}(F) = \begin{pmatrix} F_{a_1 b_1} & \cdots & F_{a_1 b_m} \\ \vdots & \cdots & \vdots \\ F_{a_n b_1} & \cdots & F_{a_n b_m} \end{pmatrix}$$

has rank less than or equal to k at every point of $A \times B$.

A more complete treatment of the result announced by Abelson is given in Mount and Reiter [17]. The necessary and sufficient conditions that k be a lower bound on the communication complexity of F involve two matrices associated with F. These are the Hessian of F, and the Full Bordered Hessian of F, abbreviated FBH(F).

$$\mathrm{FBH}(F)=\begin{pmatrix} 0 & F_{b_1} & \cdots & F_{b_m} \\ F_{a_1} & F_{a_1 b_1} & \cdots & F_{a_1 b_m} \\ \vdots & \vdots & \cdots & F_{a_1 b_m} \\ F_{a_n} & F_{a_n b_1} & \cdots & F_{a_n b_m} \end{pmatrix}.$$

If F can be computed as described above with interchange of no more than k variables, then the rank of H (F) is no more than k (on the relevant neighborhood). On the other hand if the rank of H(F) is k and is equal to the rank of FBH(F) on the relevant neighborhood, then F can be computed on the neighborhood with transmission of no more than k variables between the processors. [11]

Message spaces

The same question has been addressed in another form in the literature on size of message spaces of privacy preserving mechanisms. In this setting, the size of the message space describes the number of variables that must be transmitted among economic agents in order to verify equilibrium conditions in a privacy preserving manner. There are several types of results in that literature that are useful in the present context.

For the case of two agents, each with 2 parameters $\theta^1=(a_1,a_2)$, $\theta^2=(b_1,b_2)$, corresponding to q = 2 in Examples 1 and 2, Hurwicz, L. [6], gives necessary and sufficient conditions that there exists a privacy preserving static mechanism that realizes F(a_1,a_2, b_1, b_2) and uses a two dimensional message space. In this case we know that the message space of the parameter transfer mechanism has dimension 3, and Hurwicz's condition tells us that there is no privacy preserving mechanism with a message space of dimension 2. The equilibrium equations of the parameter transfer mechanism in this case are:

$$\begin{aligned} &m_1^1 - a_1 = 0 \\ &m_2^1 - a_2 = 0 \\ &m^2 - F\left(m_1^1, m_2^1, b_1, b_2\right) = 0. \end{aligned}$$

Hurwicz's necessary (and sufficient) condition that there exist a privacy preserving mechanism that realizes F with a message space of dimension 2 is that

$$\begin{vmatrix} 0 & F_{b_1} & F_{b_2} \\ F_{a_1} & F_{a_1 b_1} & F_{a_1 b_2} \\ F_{a_1} & F_{a_2 b_1} & F_{a_1 b_2} \end{vmatrix} = 0\pi$$

for all a and b.

In the case of the inner product, when $F(a,b) = a \cdot b$, The determinant is

$$\begin{vmatrix} 0 & a_1 & a_2 \\ b_1 & 1 & 0 \\ b_2 & 0 & 1 \end{vmatrix} = -a \cdot b,$$

which means that there can be no such mechanism for the inner product.

For the case of the function $g(a,b)=\sum_{i=1}^{q}(a_i-b_i)^2$ the Hurwicz condition, when q =2, is

$$\begin{vmatrix} 0 & -2(a_1-b_1) & -2(a_2-b_2) \\ 2(a_1-b_1) & -2 & 0 \\ 2(a_2-b_2) & 0 & -2 \end{vmatrix} \neq 0.$$

Hence, there is no mechanism that realizes g with communication of fewer than 3 variables, including the value of the function g. Therefore one of the agents must transmit at least two variables.

Chen [2] has generalized Hurwicz's necessary condition to the case of more than two parameters per agent.

Privacy preserving mechanisms that realize a given function F and Abelson's model of a distributed computation of F are related as follows. The processors in a distributed computation of F can be identified with agents in a privacy preserving mechanism that realizes F. The initial distribution among processors (or agents) of knowledge of its arguments is therefore the same for processors (or agents). Privacy is preserved in either case, because agents or processors can only base their calculations or responses on knowledge of parameters which they have either directly, or via messages received from others. Abelson considers an iterative process in which at each stage processors exchange messages based on the parameter values that reside in their respective memories, and on the messages received at earlier stages. If after a finite number of stages one of the processors computes the value of F, then the computation ends. The communication complexity is the total number of values of variables transmitted between the processors in all preceding stages.

For each such distributed computation of F there is a privacy preserving mechanism that realizes F with a message space whose dimension is 1 more than the number of real numbers exchanged in the distributed computation of F. The size of the minimal message space of mechanisms that realize F minus 1 is a lower bound on the communication complexity of F.

Rectangles method

Knowledge of the equilibrium equations of a privacy preserving mechanism that realizes a given goal function or decision rule can be helpful in designing an algorithm for computing the decision rule. This is illustrated in Example 3. The equilibrium equations can in principle be obtained from the goal function by constructive methods. When the goal function is smooth, the methods reported in Hurwicz, Reiter and Saari [7], permit construction of mechanisms using methods of differential topology. These include methods based on Frobenius' theorem for integrable distributions, and methods based on differential ideals. They involve solving systems of partial differential equations in one guise or another.

The Rectangles Method has been developed for the same purpose (Hurwicz and Reiter, [8]). This method uses elementary algebraic constructions, and given the distribution of knowledge of parameters among the agents, constructs the message space, the equilibrium equations and outcome function of the mechanism, using only properties of the given goal function F.

Williams genericity theorem

Williams considered a parameter space $\Theta = \Theta^1 \times \cdots \times \Theta^N$ where Θ^i is an open subset of R^{k_i}, the Euclidean space of k_i dimensions, for $i \in \{1,\cdots,N\}$. The set Θ^i consists of parameters that may be observed together; an agent who may observe parameters in Θ^imay not also observe parameters in Θ^j if j≠i.

Williams considered functions on Θthat are continuously differentiable of every order. The space of these C^∞functions, denoted $C^\infty(\Theta)$ is given the Whitney topology. Because Williams' analysis is local, for any point $\theta \in \Theta$he defines as equivalent all functions that are the same on some neighborhood of θ. For $\overline{\theta} \in \Theta$, the space of all such functions is denoted $C^\infty_{\overline{\theta}}(\Theta)$. In the present interpretation, these are the possible goal functions or decision rules. Williams supposes that given a function $F \in C^\infty_{\overline{\theta}}(\Theta)$,(considered on some neighborhood Θ' of $\overline{\theta}$) can be realized by a privacy preserving mechanism in equation form, with agent i having q_i equations. He imposes conditions that ensure that the dimension of the message space of the mechanism is $\sum_{i=1}^{N} q_i$.

Williams genericity theorem says that for any $\overline{\theta} \in \Theta$,satisfying his assumptions, there is an open dense set Ω of $C^\infty_{\overline{\theta}}(\Theta)$ such that if $F \in \Omega$ is realizable as above, then there is at most one value of the index i, say i = j, such that $q_i \neq k_i$.

In the context of this paper, Williams's result tells us that for an open dense subset of decision rules, all but one of the (minimal) set of Roles defined by the initial distribution of observed parameters will have to transmit all observed parameter values to the Role (or Roles) involved in computing the decision rule. This in turn means that for an open dense set of functions, the task of coordination presented by that decision rule requires that communication among Roles grows with the dimensions of the parameter spaces. The resulting organizational structure will look like one centralized firm.

It is a question whether the functions that define coordination tasks that permit a higher degree of separation among subunits, which are relatively rare in $C^{\infty}_{\overline{\theta}}(\Theta)$ are as rare in the economic situations that exist.

FOOTNOTES

[1] Indeed, some manufacturing in Sheffield, and Leeds in England was organized in this way in the the 19th century. Alternatively, since gears are an intermediate product used in the manufacture of something else, any purchasers of gears could make their own gears. This is sometimes done, but the indivisibility of machines and the fact that some advantages of specialization are lost militate against this. Often a manufacturer of a final product, say earth moving equipment will choose to make the gears they require, encurring significantly higher costs in order to get more reliable control of delivery time.

[2] The Modular Network model is an extension of the neural network model of McCulloch and Pitts [13]. The relationships between our model and standard models of computation, namely Turing machines a nd finite state machines is discussed in Mount and Reiter [16], where it is shown via certain limit theorems, that the Modular Network model is an idealization of standard computing in the same sense in which measurement, say, of length, using real numbers is an idealization of actual measurement, which can be at best rational.

[3] Alternatively, if deviations from optimal decisions could be weighed against costs of computation, then 'optimal adaptation' could mean 'maximization of net performance.'

[4] In fact it is shown that G may be replaced by an equivalent regular tree, i.e., a fan-in, in which every path from a leaf to the root has the same length.

[5] Generally that cost would depend on the scale of the organization. In the present model the scale or size of the organization depends on the number of parameters needed to specify an environment. Hence, we would expect the capital cost to be a function of the

number of parameters. A step function, with the steps depending on the number of parameters, would be a good candidate for that function. The present formulation is the simplest case of such a function, namely one with one step of size α_0' for any non zero number of parameters

6. Note that Proposition 1) and remarks 2) and 3) which follow it hold whether crosslinks are or are not internalized.

7. It is also possible for a given coordination problem that if the set of environments is small, a form of organization into several independent units might be preferred or indifferent to organization into one unit, even though the latter would be strictly better when the set of environments is large.

8. For the distinction between 1 and 2 real numbers to be meaningful requires a regularity condition that has the effect of excluding dimension-increasing continuous maps, such as the Peano function. See [Mount and Reiter [14].

9. (Without this simplifying assumption, given any parameter vector $\theta(n)$ with n nonzero coordinates, the projection of $\theta(n)$ on an n -dimensional subspace $\Re^n$ can be written

$$\tilde{\theta}(n) = (\theta_{i_1}, \theta_{i_2}, \ldots, \theta_{i_n})$$

where i_j is the j^{th} non-zero coordinate of $\theta(n)$. In this language the simplification is to assume that $i_j = j \quad$ for $j = 1, \ldots, n$.)

10. Note that D and V are also functions of q.

11. Abelson's result is based on a theorem of Leontief [12].

REFERENCES

1. H. Abelson, Lower bounds on information transfer in distributed computations, *J. Assoc. for Computing Machinery* **27** (1980), 384-392.

2. P. Chen, A lower bound for the dimension of the message space of the decentralized mechanisms realizing a given goal, *J. Math. Econ.* **21** (1992), 249-270.

3. R. Coase, The nature of the firm, *Economica* **4** (1937), 385-405.

4. J. Y. Halpern and Y. Moses, knowledge and common knowledge in a distributed environment." *J. Assoc. for Computing Machinery* **37** (1990), 549-587.

5. B.R. Holmstrom and J. Tirole "The Theory of the Firm," Handbook of Industrial Organization, (1989) **I**, eds., R. Schmalensee and R.D. Willig, Elsevier Science Publishers, NV, 63-127.

6. L. Hurwicz, "On Informational Decentralization and Efficiency in Resource Allocation Mechanisms," in ed., S. Reiter, Studies in Mathematical Economics, MAA Studies in Mathematics, **25** (1986), Mathematical Association of America, 237-238.

7. L. Hurwicz, S. Reiter and D. Saari, "On Constructing an Informationally Decentralized Process Implementing a Given Performance Function," Mimeo, (1972), presented and distributed at the Econometric Society World Congress, Aix en Provence.

8. Hurwicz, L., and S. Reiter (1989) "Designing Mechanisms by the 'Method of Rectangles,' Mimeo, presented at the NBER-NSF Conference on Decentralization, Northwestern University, Evanston, Illinois, 1990, and at the NBER-NSF Conference on Decentralization, University of California, Berkeley, 1993.

9. J.S. Jordan, "The Informational Requirements of Local Stability in Decentralized Resource Allocation Mechanisms," in eds., Groves, Radner, Reiter, Information, Incentives, & Economic Mechanisms, (1987), University of Minnesota Press, Minneapolis.

10. J.S. Jordan and D. Xu, "On the Communication Complexity of Expected Profit Maximization," Mimeo, (1994), Department of Economics, University of Minnesota, Minneapolis, presented at NBER Decentralization Conference April 1995.

11. T.C. Koopmans and M.J. Beckmann, Assignment problems and the location of economic activities, *Econometrica* **25** (1957), 53-76.

12. W. Leontief, A note on the interrelation of subsets of independent variables of a continuous function with continuous first derivatives" *Bull. Amer. Math. Soc.* **53** (1947), 343-350.

13. W. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity, *Bull. Math. Biophysics* **V** (1943), 115-133.

14. K.R. Mount and S. Reiter, The informational size of message spaces, *J. Econ. Theory* **6** (1974), 161-192.

15. K.R. Mount and S. Reiter, "Computation, Communication, and Performance in Resource Allocation," paper presented at the CEME-NBER Decentralization Seminar, (1982), University of Minnesota, Minneapolis (unpublished).

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

17. K.R. Mount and S. Reiter, (1993) A lower bound in computational complexity given by revelation mechanisms, *Econ. Theory* **7** (**2**) (1996), 237-266.

18. R. Radner, Hierarchy: The economics of managing, *J. Econ. Lit.* **30** (1992), 1382-1415.

19. R. Radner, The organization of decentralized information processing, *Econometrica* **61** (1993), 1109-1146, p.1109.

20. S. Reiter, A system for managing job shop production, *J. of Business of the Univ. of Chicago* (1966), 371-393.

21. S. Reiter, "There is No Adjustment Process With Two-Dimensional Message Space for 'Counter Examples'," Mimeo (1979), Northwestern University.

22. S. Reiter, "Coordination and the Structure of Firms," Discussion Paper No. 1121 (1995), Center for Mathematical Studies in Economics and Management Science, Northwestern University (unpublished).

23. S.R. Williams, Implementing a generic smooth gunction, *J. Math. Econ.*, **13** (1984), 273-288.

24. O. Williamson, "Markets and Hierarchies; Analysis and Antitrust Implications", The Free Press, 1975.

25. O. Williamson, (1985) "The Economic Institutions of Capitalism; Firms, Markets, Relational Contracts," NY, The Free Press, 1985.

26. O. Williamson, "Transaction Cost Economics," Chapter 3, in R. Schmalensee and R. Willigs, eds., Handbook of Industrial Economics, NY, North Holland, 1989.

Appendix I

Introduction

In the Appendix we consider the problem constructing efficient assignments of modules to agents. To begin with we consider networks that are trees. That is, we are given a modular network in the form of a tree, together with specification of observability of inputs-the initial dispersion of information-and we want to construct all efficient assignments of modules to agents and schedules of execution of modules in time subject to the parallel constraints. The construction of efficient assignments is done in two parts. The first is the assignment of modules to agents so as to minimize the number of crosslinks, without regard to the parallel constraint. This is presented in Appendix I. The second begins with a modular network whose modules are assigned to agents so as to minimize the number of crosslinks, and schedules the evaluation of the modules in time subject to the parallel constraint. This is presented in Appendix II.

Appendix I. Assigning Modules to Agents so as to Minimize Crosslinks

To avoid heavy notation we begin with the problem of assigning a (2,1)-network to two agents. For the present purpose it is not necessary to specify the set of elementary operations, because the properties of the graph that matter do not involve distinctions among modules.[1] We assume here that a real valued function $P : X \to Y$ is given, and that we are also given a (2,1)-network (with some class of elementary operations) that computes P in time t*. Recall that every such

[1] However, under certain conditions the minimal delay network for a given function P is unique (up to equivalence). This is the case when the function P is real analytic to be computed up to degree M, and the elementary functions are truncated (to degree M) real analytic functions of two variables. See [Mount and Reiter (1993)] for the full details.

network has an equivalent network with the properties: (A) its directed graph is a tree; (B) it uses the same modules as the original network; and (C) it computes P in time t*. Since we are dealing with (2,1)-networks that are trees, the graphs have the property that:

(i) each node has input degree equal to 2, except for the input nodes, which have input degree 0;

(ii) each node has output degree 1, except for the root, which has output degree 0;

(iii) there is a unique walk (directed path) from each node to the root.

The problem of finding efficient assignments is approached in two steps. The first step is to find an assignment of nodes to agents that is "good" in the sense that it results in a small number of crosslinks. Such an assignment becomes the starting point for the second step, scheduling the operations in time so as to minimize delay. Then the resulting assignment is revised to generate all efficient assignments. We begin with the first step, which focuses on crosslinks.

To achieve a more colorful exposition, let the two agents be named "red" and "blue", respectively and denote them r, and b. (Since the parameter r in the (r,d)-network has the value 2 throughout this section, there is no ambiguity in this notation.) Since the effects of assigning modules to agents depend only on the (2,1)-tree we start from, we may suppress reference to the modules associated with the vertices V(T) of the tree T. In the case of (2,1)- trees the problem of finding assignments that give a minimal number of crosslinks may be approached in two different ways, one in terms of finding minimal cuts, the other in terms of properties of coloring functions or colorings. We begin with the latter approach.

Preliminaries and Notation. Denote the set of agents by $C = \{r,b\}$.

A function $\Psi : V(T) \to C \equiv \{r,b\}$ from the set of vertices of T to the set of agents, called a coloring of T, is an assignment of the vertices of T to agents. The restriction of Ψ to the set of input vertices of T, denoted Ψ_I, is called an input coloring. It is assumed unless otherwise stated that Ψ_I, is given. (This corresponds to the assumption either that each agent has been given the task of observing certain variables--arguments of P--or that each agent has private information about (environmental) parameters that are the arguments of P.)

An arc of T from vertex x to vertex y is denoted (x,y).

A coloring ψ of T induces a function ν_ψ: $\mathbf{A} \rightarrow \{0,1\}$ on the arcs of T defined by the condition

$$\nu_\Psi(x,y) = \begin{cases} 0 & \text{if } \Psi(x) = \Psi(y) \\ 1 & \text{otherwise} \end{cases}$$

If $\mathbf{A} \supset A$ is a subset of the arcs of T, let

$$\nu_\Psi(A) = \sum_{a \in A} \nu_\Psi(a).$$

A coloring ψ of a tree T is minimal if it minimizes the number of crosslinks $\nu_\psi(T)$.

Recolorings and Changes in the Number of Crosslinks

If $N \subseteq V(T)$ is a subset of nodes, the operation Δ_N carries the coloring ψ to ψ_{Δ_N} i.e. $\psi \rightarrow \psi_{\Delta_N}$ by

$$\psi_{\Delta_N}(i) = \begin{cases} \psi(i) & \text{if } i \notin N \\ x \text{ where } x \in C \setminus \psi(i) & \text{if } i \in N \end{cases}$$

If N = {j}, we shall write Δ_j for Δ_N. The operation Δ_N changes the color of each node in the subset N.

(Recall $C = \{r,b\}$. Therefore if ψ(i) = r and i ∈ N, then ψ_{Δ_N} (i) = b; if ψ(i) = b and i ∈ N then ψ_{Λ_N} (i) = r.)

If A and B are subsets, then A Δ B = (A ∩ $\tilde{B}$) ∪ ($\tilde{A}$∩ B), where $\tilde{X}$ denotes the complement of X; A Δ B is the symmetric difference of A and B.

In particular, if $N_1,\dots,N_q$ form a partition of N, then $\Delta_N = \Delta_{N_{p(1)}} o \ldots o \Delta_{N_{p(q)}}$ where $p_{(1)} \cdots p_{(q)}$ is a permutation of 1,...,q, and "o" denotes composition.

A recoloring Δ_N of T induces a change in the number of crosslinks $v_\Psi \rightarrow v_{\Psi_{\Delta_N}}$ as follows.

Let a = (x,y) be an arc of T. Then

$$v_{\Psi_{\Delta_j}}(a) = \begin{cases} [v_\Psi(a) + 0] \text{ (mod 2)} & \text{if} \quad x \neq j \text{ and } y \neq j \\ [v_\Psi(a) + 1] \text{ (mod 2)} & \text{if} \quad x = j \text{ and } y = j \end{cases} .$$

Define

$$\Delta_j v_\Psi(a) = v_{\Psi_{\Lambda_j}}(a) - v_\Psi(a)$$

The expression $\Delta_j v_\psi(a)$ is the change in the status of the arc a (counted as 0, +1 or -1 according to whether a remains a selflink, changes from a selflink to crosslink or vice versa) as a result of the recoloring Δ_j.

Note that if a = (x,y), and j ≠ x and j ≠ y, then $v_{\Psi_{\Lambda_j}}(a) - v_\Psi(a) = 0$, and if x = j or y = j, then $v_{\Psi_{\Delta_j}}(a) - v_\Psi(a) = \pm 1$. Hence,

$$\Delta_j v_\Psi(a) = \begin{cases} 0 & \text{if } j \notin x \text{ and } j \notin y \\ v_{\Psi_{\Delta_j}}(a) - v_\Psi(a) = +1 & \text{if } v_\Psi(a) = 0 \text{ and } j = x \text{ or } j = y \\ v_{\Psi_{\Lambda_j}}(a) - v_\Psi(a) = -1 & \text{if } v_\Psi(a) = 1 \text{ and } j = x \text{ or } j = y \end{cases}$$

It follows directly from the fact that $\Delta_A o \Delta_B = \Delta_B o \Delta_A$, for every A, B, that,

<u>Lemma 1</u>: $v_{\Psi_{\Delta_{jo\Delta i}}} = v_{\Psi_{\Delta_{io\Delta j}}}$

Let **N** = {1,2,...,q}, and let $\Delta_1, \Delta_2, ..., \Delta_q$ be a sequence of recolorings of nodes 1,2,...,q. The total effect of these changes on the crosslink count for any subset A of arcs of T is as follows.

<u>Lemma 2</u>: Let $V(T) \supset A$.. Then,

$$\begin{aligned} v_{\Psi_{\Lambda_N}} A = v_{\Psi_{\Delta_q o \ldots o \Lambda_1}}(A) &= \sum_{a \in A} \left[\sum_{i=1}^{q} \Delta_i v_{\Psi_{\Delta_{i-1}}}(a) \right] + v_{\Psi}(a) \\ &= \sum_{a \in A} \left[\sum_{i=1}^{q} \Delta_i v_{\Psi_{\Lambda_{i-1}}}(a) \right] + v_{\Psi}(A) \end{aligned}$$

Proof: By definition of Δ_j v, for any $a \in A$

$$\Delta_1 v_{\Psi}(a) = v_{\Psi_{\Lambda_1}}(a) - v_{\Psi}(a)$$

$$\Delta_2 v_{\Psi} \Delta_1(a) = v_{\Psi_{\Lambda_2}}(a) - v_{\Psi} \Delta_1(a)$$

.

.

.

$$\Delta_q v_{\Psi_{\Lambda_{q-1}}}(a) = v_{\Psi_{\Delta_q}}(a) - v_{\Psi_{\Delta_{q-1}}}(a)$$

Summing these equations yields

$$\Delta_1 v_{\Psi}(a) + \Delta_2 v_{\Psi_{\Lambda_1}}(a) + \ldots + \Delta_q v_{\Psi_{\Delta_{q-1}}}(a) = v_{\Psi_{\Delta_q}}(a) - v_{\Psi}(a)$$

or

$$\sum_{i=1}^{q} \Delta_i v_{\Psi_{\Delta_{i-1}}}(a) + v_{\Psi}(a) = v_{\Psi_{\Lambda_q}}(a)$$

Summing over a $\in$ A gives the statement to be proved;

$$\sum_{a \in A} \left[\sum_{i=1}^{q} \Delta_i v_{\Psi_{\Delta_{i-1}}}(a) \right] + v_{\Psi}(A) = v_{\Psi_{\Delta_N}}(A).$$

Because of Lemma 1, any permutation of 1,...,q gives the same results, i.e. the changes of color of nodes may be done in any order, with the same result.

Consider a connected subgraph G of the graph T such that G has a root. I will refer to such a subgraph as a subtree of T. Examples are:

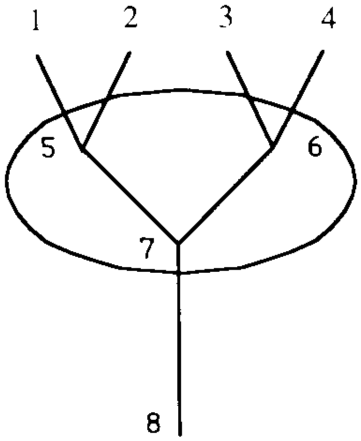


FIGURE AI-1

The subset of nodes and directed arcs are indicated in the enclosed areas in Figure AI-1.

Those nodes that receive inputs from nodes outside the subset are input nodes of G and the node outside of G to which the output of the root node of G is sent is the output connection of G. E.g. in figure a, G consists of nodes 5, 6, and 7 and arcs (1,5), (2,5), (3,6), (4,6), (5,7) and (6,7); nodes 5 and 6 are input nodes of G, 7 is the root and 8 is the output connection of G. The given input colors of G are the colors of nodes 1, 2, 3, and 4 i.e. of all nodes i not in G such that there is an arc (i,x) where x is in G; for such an arc, x is an input node of G.

Levels or Tiers of Nodes or Vertices in a Tree

The level of a vertex v in T is defined as follows. A path $p = n_0, n_1, n_2, \ldots, n_q$ is a complete path through v in T if:

(i) n_0 is a leaf of T

(ii) n_q is the root of T

(iii) for each $i = 0, \ldots, q-1$, (n_i, n_{i+1}) is an arc of T

(iv) for some $i \in \{0, \ldots, q_{-1}\}$ $n_i = v$.

We define the height of a vertex v in a complete path $p = n_0, n_1, \ldots, n_{q-1}$ through v in T to be the value i* of i such that $n_{i^*} = v$, where $i \in \{0, 1, \ldots, q_{-1}\}$.

The level of a vertex v in T is, $l(v) \equiv \max \{i^* | i^*$ is the height of v in p, and p is a complete path through v in T.}

The ordering of the nodes of G by level or tier is the ordering inherited from the ordering of T by level or tier.

The first level nodes of G are those whose level in T is the minimum over G. Second level nodes are all those in G whose level in T is one more than the (common) level in T of the first level nodes of G. The i^{th} level nodes in G are all those whose level in T is one more than the $(i-1)^{st}$ level nodes of G.

Conditions for Minimal Colorings

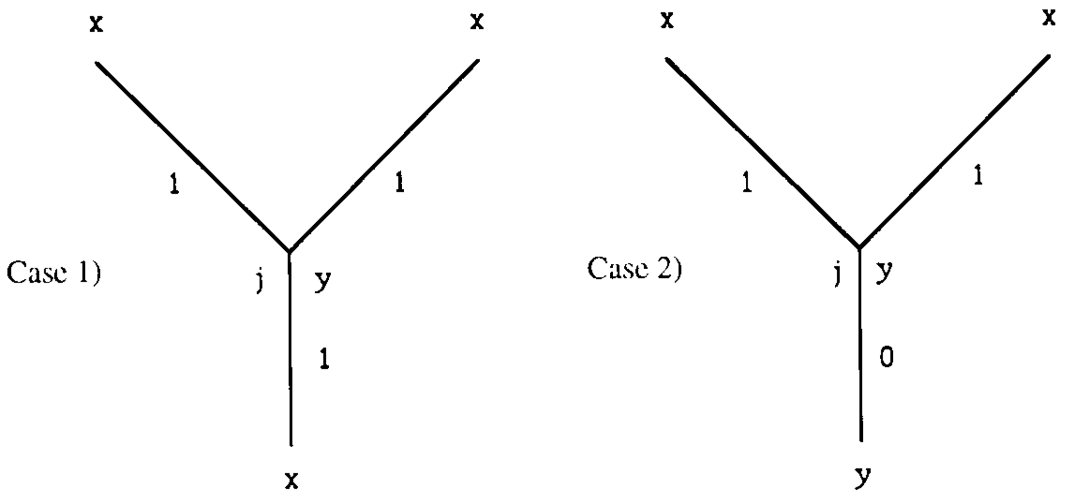
Given a coloring of the inputs of G, the first level nodes of G may be classified as (x,x)-nodes, if both inputs are colored x, and (x,y)-nodes if there is one input of each color $x,y \in C$.

Recall that a coloring ψ of a tree T (or subtree) is minimal if it minimizes the number of crosslinks $v_{\psi}(T)$ over all colorings.

The following are necessary conditions for minimality.

Lemma 3.(i): If G is a subtree of T all of whose input nodes are (x,x)-nodes, then in a minimal coloring of G every node in G is colored x.

Proof: Suppose there is an (x,x)-node, say j, colored $y \neq x$. Then the two possibilities are:



Case 1)

Case 2)

FIGURE AI-2

In case 1) there are 3 crosslinks and $\Sigma_{a\in \{(x,j),(y,j),(j,z)\}}\Delta_j \nu(a) = +3$. Hence if $\psi(j) = \psi = x$ then the total of cross links goes down by 3 with no change in the rest of the network.

In case 2) changing node j from y to x leads to a reduction of 1 in the number of crosslinks with no change in the rest of the network. Hence if $\psi(j) = y$, ψ cannot minimize $\nu(T)$. As a Corallary, Lemma 3.(i) implies Lemma 3.(ii).

Lemma 3.(ii): In a minimal coloring of T every (x,x)-node is colored x, $x \in C$.

Lemma 4: If G is a subtree of T all of whose input nodes are (x,y)-nodes, and whose root i has the output connection j whose color is $z \in \{r,b\}$ then in a minimal coloring the color of every node in G must also be z.

Proof: Suppose G has p input nodes, all of which are (x,y)-nodes, as in Figure 3.

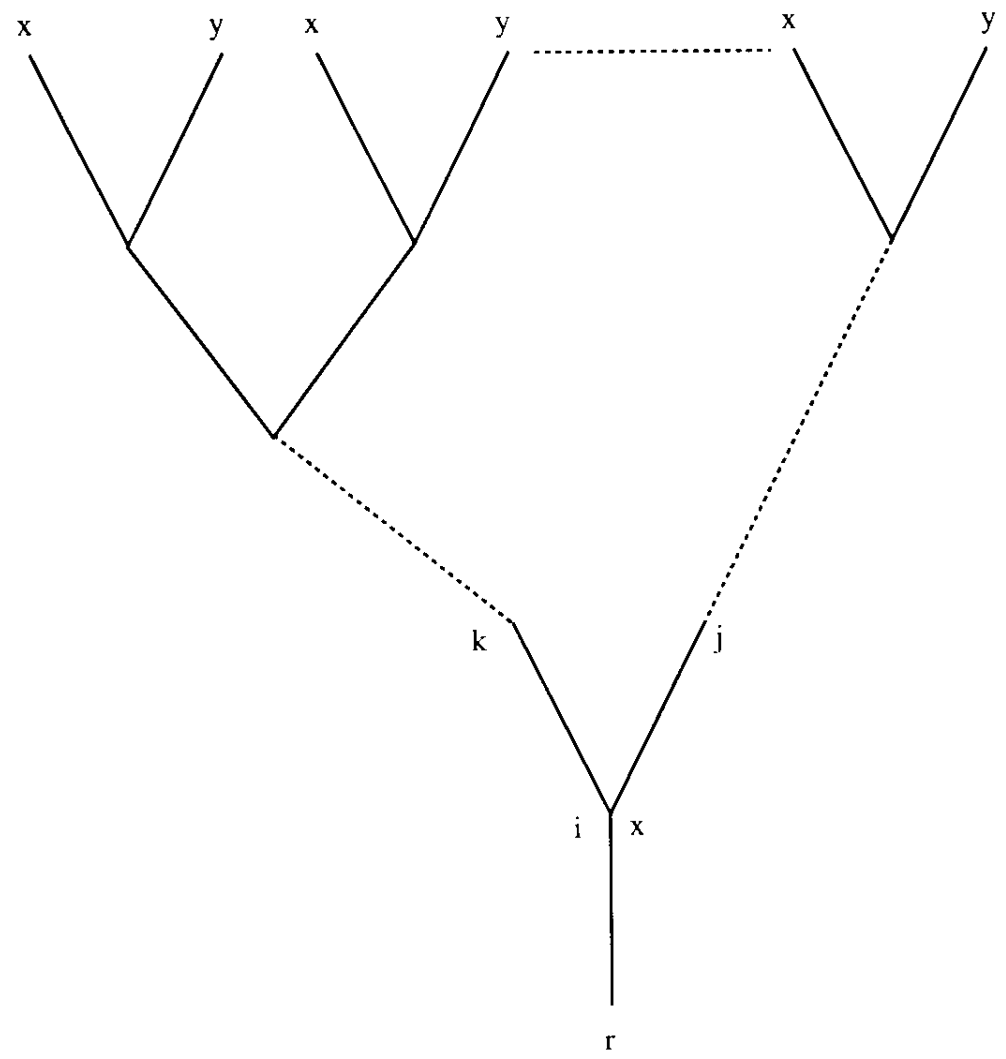


FIGURE AI-3

First note that the number of crosslinks from input connections to (input) nodes is in this case independent of the coloring of first tier nodes, because each first tier node is an (x,y)-node, there is exactly one crosslink per node, whether its color is x or y.

Second, if all the nodes of G are assigned the same color, then there are no additional crosslinks between the input nodes and the root. If some node in G is assigned a different color, then there is at least one additional crosslink in G. If the common color assigned to all nodes is x, and x is also the color of the output connection of i, i.e., if x = z, then there is no crosslink between the root and the output connection; if the color is y ≠ z, then there is one crosslink between the root of G and the output connection. It follows that the minimum number of crosslinks, equal to the number of input nodes of G, is attained by assigning all nodes of G the color z. []

We turn now to sufficient conditions for minimality.

Lemma 5: Let G be a subtree whose input nodes are either (x,x)-nodes or (x,y)-nodes and whose output connection is colored x. Then,

(1) the constant coloring $\overline{\Psi}|_G\left(j\right) = x$ for all j $\in$ G minimizes the number of crosslinks,

(2) the number of crosslinks $v_{\overline{\Psi}|_G(G)}(G)$ is equal to the number of (x,y)-nodes among the input nodes of G, and

(3) the minimizing coloring $\overline{\Psi}|_G$is unique.

Proof: Let the number of (x,y)-nodes among the input nodes of G be q. If each such node is colored x then, since (x,x)-nodes must also be colored x (as required by Lemma 3.(i), a necessary condition for a minimizing coloring), all input nodes are colored x. Consider the subtree G' consisting of the nodes of G other than its input nodes. The subtree G' has input nodes consisting entirely of (x,x)-nodes. Since its output connection is the same as that of G, it is colored x. Hence assigning the color x to all nodes of G' yields a total of zero crosslinks. This is minimal.

Thus, in the original subtree G there will be one crosslink for each input colored y. Therefore the number of crosslinks is equal to the number of (x,y)-nodes among the input nodes of G.

To see that this coloring is the unique minimizer, suppose there is another. Then some node j must be colored y ≠ x. The path from j to the root of G must contain only nodes colored y, for if any node on this path were colored x, there would be at least one crosslink added by the arc from the last node colored y to the first node on the path colored x. This implies that the root must be colored y. But then the arc from the root of G to the output connection is a crosslink. Thus, the number of crosslinks determined by such a coloring function is larger by at least 1 than $\nu_{\overline{\Psi}|_G}(G)$. []

The proof of Lemma 5 can be slightly extended to establish:

Lemma 6: Let G be a subtree whose input nodes are either (x,y)-nodes or (x,x)-nodes and whose output connection is colored y ≠ x. Then:

(1) If there is at least one (x,x)-node, then the minimum number of crosslinks is one more than the number of input nodes that are (x,y)-nodes. (Note that if there are no (x,x)-nodes, then Lemma 3.(i) applies and shows that the minimizing coloring is the constant coloring $\overline{\overline{\Psi}}|_G(j)$= y and the minimum number of crosslinks is the number of input nodes).

(2) The constant coloring $\overline{\Psi}|_G(j)$ = x for every j ∈ G is a minimizing coloring.

Lemma 7: Let G be a subtree whose input nodes are either (x,y) or (x,x)-nodes, and whose output connection is colored y ≠ x.

1) If the minimum subtree of G generated by the (x,x)-input nodes of G is not G itself, then the minimizing coloring is not unique. Moreover,

2) if $\tilde{\Psi}|_G$ is a coloring such that the set of nodes colored x is a subtree, H, that contains all input nodes of G that are (x,x)-nodes, and whose complement in G, consisting of all nodes colored y, is also a subtree, G\H, such that its root is the root of G, then $\tilde{\Psi}|_G$ is also a minimizing coloring of G, and is not a constant coloring.

Proof: If all nodes are colored x, then the there are as many crosslinks as there are (x,y)-nodes among the input nodes, plus one for the arc from the root, colored x, to the output connection colored y ≠ x.

To see that this is minimal given that there is at least one (x,x)-node among the input nodes, note that the number of crosslinks is at least the number of (x,y)-nodes, and that all (x,x)-nodes, of which there is at least one, say, node j, must be colored x. Hence the path from node j to the root either, (i) consists entirely of nodes colored x, including the root, or (ii) there is somewhere in that chain a node colored y ≠ x. If (i), then the arc from the root to the output connection is a crosslink. If (ii), then the arc from the last node in the chain colored x to the node colored y is a crosslink.

To see that the coloring $\overline{\Psi}\,|_G$ is not always the unique minimizer when there is at least one (x,x)-node, color all the (x,x)-nodes x and let H be the smallest subtree with these nodes among its input nodes.

If H = G, then the constant coloring $\overline{\Psi}\,|_G\,(j) = x$, j ∈ G is the unique minimizer.

If H ≠ G, then the root of H is not the root of G. Color all nodes of G\H the color y, including the root of G. Then, all nodes of H are (x,x)-nodes except for the input nodes of H that are (x,y)-nodes, and all nodes of G\H are (y,y)-nodes except for the input nodes of G that are not input nodes of H and are (x,y)-nodes, and except for the additional (x,y)-node of G\H which is the output connection of the root of H. Thus the number of crosslinks is the total of (x,y)-nodes of G plus 1, which is the minimum. []

Some examples.

Example 1

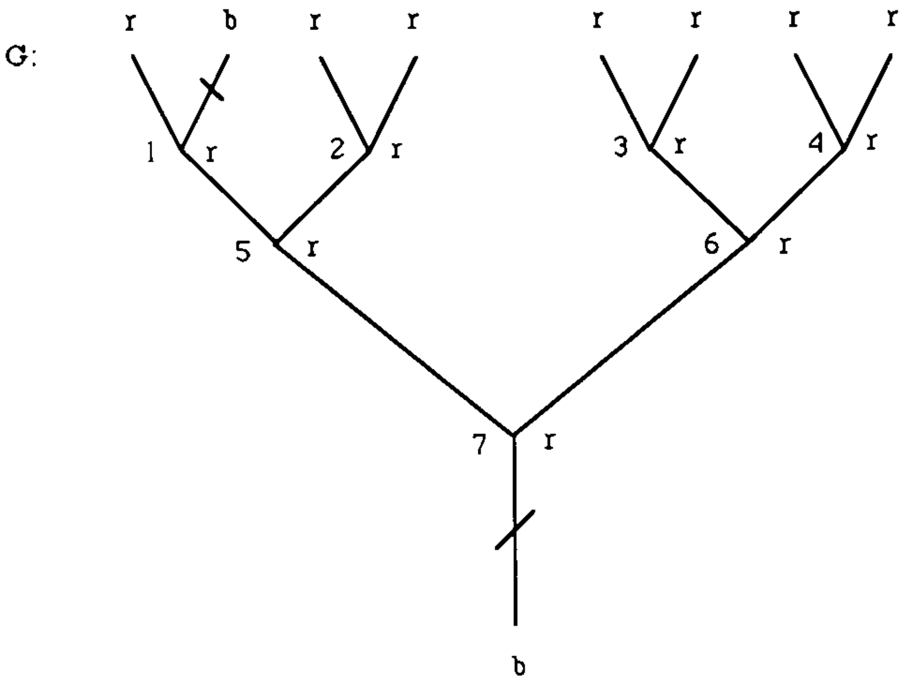


FIGURE AI-4

The minimal subtree of G generated by nodes 2, 3 and 4 is equal to G. The unique minimizing coloring is $\overline{\psi}|_G\,(j) = r$, for j = 1,...,7. The minimum number of crosslinks is 2.

Example 2

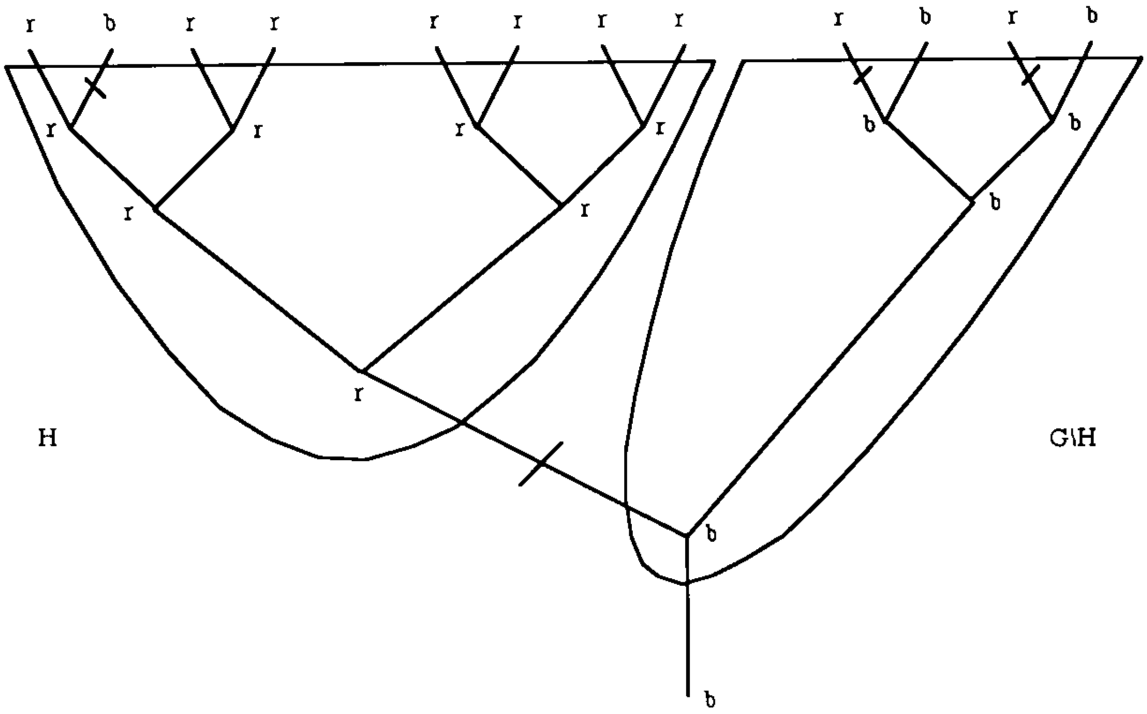


FIGURE AI-5

In Figure AI-5 the number of crosslinks is 4, equal to the minimum.

The constant coloring r, shown in Figure AI-6, also gives 4 crosslinks.

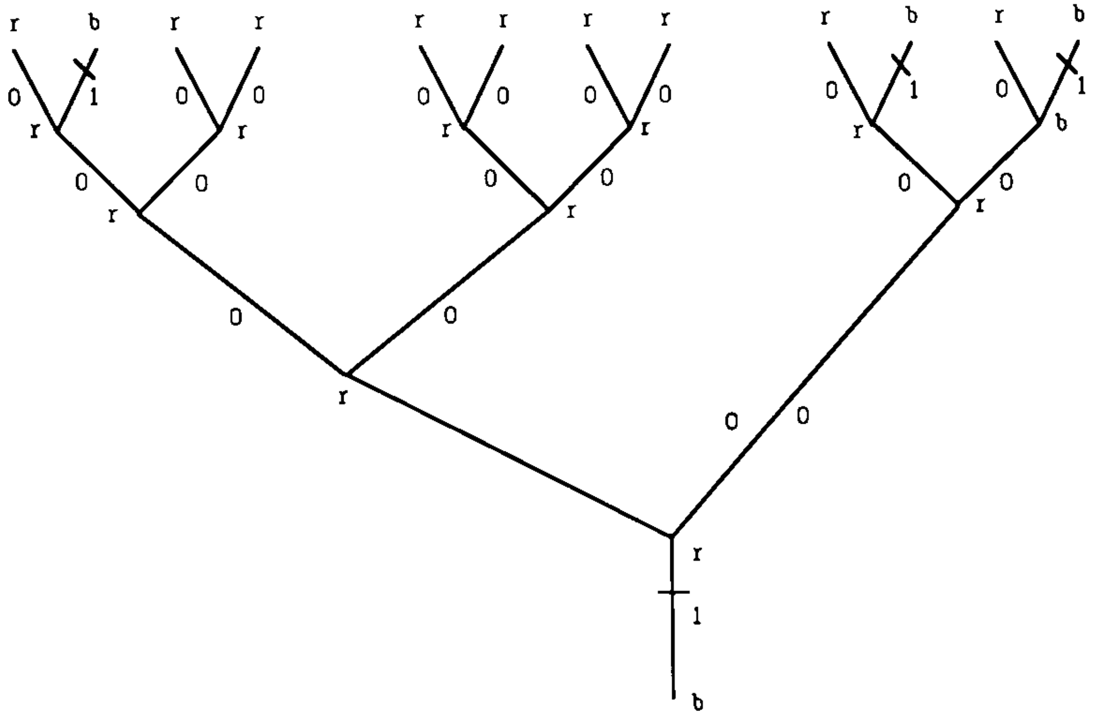


FIGURE AI-6

We turn now to sufficient conditions for a minimal coloring.

Theorem 1. A coloring function ψ defined on a (2,1)-tree T is a minimal coloring of T if and only if it satisfies Lemmas 3 to 7. I.e., the conditions of Lemmas 3 to 7 are each necessary and together sufficient.

Proof. Necessity is already established by the Lemmas.

Sufficiency: If given a coloring ψ there is a subtree G of T that can be recolored to reduce the number of crosslinks below v_ψ, then there is a subtree G' of G which satisfies the hypotheses of one of Lemmas 3 to 7, but not the conclusions. This is the case, because, by Lemmas 1 and 2 recoloring of G can be reduced to successive recoloring of subsets of the nodes of G. []

Next we present an algorithm for constructing minimal colorings.

An Algorithm for minimally coloring a (2,1)-tree T

A minimal coloring function ψ can be constructed by a two-pass process. The first pass starts with the leaves of the tree T and goes level by level to the root; the second pass starts with the root and goes in reverse order level by level to the leaves. The process makes use of a provisional color, an indeterminate or variable denoted u, whose domain is the set C. This color is used in the first pass and is replaced in the second pass by one of the colors r or b.

First Pass.

1) A node may be colored during the first pass whenever the color of its inputs or parent nodes is known.

2) A node whose parents (or inputs) have the same color (either r,r, or b,b or u,u) is given the same color as the parents.

3) A node whose parents (or inputs) have colors r,b (or b,r) is given the provisional color u.

4) A node whose parents have the colors x,u (or u,x), where x is either r or b, is given the color x.

5) The first pass is completed when all nodes have been assigned a color, (including u).

Second Pass.

6) If the root is colored u, then it is recolored arbitrarily either r or b.

7) A node colored u on the first pass may be recolored if its child is colored r or b; a node colored u is recolored with the color of its child. Thus, once the root is recolored, if necessary, as provided by 6), the parent nodes of the root may be recolored, and so on through the levels of the tree to the leaves, until all nodes are colored either r or b.

8) The coloring process ends when all nodes have been assigned one of the colors r or b.

It is straightforward to verify that the coloring function defined by this process satisfies the necessary and sufficient conditions for a minimal coloring given in Theorem 1.

More Than Two Agents:

Consider next assignment of nodes to more than two agents. The aim is still to find an assignment that minimizes the number of crosslinks. The algorithm presented above can be extended to more than two agents as follows.

A coloring algorithm for (2,1)-trees with more than 2 colors.

Let the set of colors be $C = \{c_1,...,c_p\}$. Let (T, Ψ_I) be a (2,1)-tree T with input coloring $\Psi_I : I \rightarrow C$. The algorithm, like the one for p=2, consists of two-passes.

First Pass.

Let $\tilde{\Psi}$: V(T) → 2^C denote a provisional coloring of (T,ψ_I). The provisional coloring must satisfy

(i) $\tilde{\Psi}\,|_I = \Psi_I$.

A node n of (T,ψ_I) is called an A,B-node if u and v are the parents of n and $\tilde{\psi}(u) = A \quad \tilde{\psi}(v) = B$, where A and B are subsets of C. (If B = A, then n is an A,A-node.)

(ii) If n is an A,B-node, then

$$\tilde{\psi}(n) = \begin{cases} A \cap B & \text{if} \quad A \cap B \neq \phi \\ A \cup B & \quad \text{otherwise.} \end{cases}$$

<u>Second Pass</u>.

If n is the root of T, let

$$\psi(n) \in \tilde{\psi}(n),$$

i.e., choose any element of $\tilde{\psi}(n)$ as the color of the root. If n is not the root of T, let

$$\psi(\mathrm{n}) = \begin{cases} x \in \tilde{\psi}(n) \cap \tilde{\psi}(\xi(n)), \text{ if } \tilde{\psi}(\xi(n) \cap \tilde{\psi}(\xi(n)) \neq \varnothing, \text{ where } \xi(n) \\ \qquad\qquad\qquad\qquad \text{denotes the child of n in T,} \\ x \in \tilde{\psi}(n) \qquad\qquad\qquad \text{otherwise,} \end{cases}$$

i.e., the color assigned to n is an arbitrary element of $\tilde{\psi}(n) \cap \psi(\xi(\mathrm{n}))$ if that set is not empty and is an arbitrary element of $\tilde{\psi}(n)$ otherwise.

Notice that this procedure reduces to the coloring algorithm described above when there are just two colors.

<u>Minimal colorings of DAG's in terms of c-cuts</u>

We now take up the second approach, which is based on the characterization of minimal colorings in terms of <u>c-cuts</u>, defined below. The approach via c-cuts is treated in the more general context of finite directed acyclic graphs, rather than (2,1)-trees.

Let G be a directed acyclic graph (<u>DAG</u>). Let V(G) denote the set of nodes (vertices) of G, and **A** the set of directed arcs. The arc from node m to node n is denoted (m,n). A *path* of G is a connected sequence of distinct arcs of G with distinct vertices, i.e. $(n_0,n_1)(n_1,n_2),\ldots,(n_{k-1},n_k)$, where for each arc $(n_i,n_{i+1}), n_i \neq n_{i+1}$, is a path from n_0 to n_k. That G is acyclic means there is no path from a node n_0 to a node n_k with $n_k = n_0$.

DAG G has a *root* if there is exactly one node n_k of G such that there is no arc of G of the form (n_k,m).

We associate with a DAG, G, an undirected graph $\tilde{G}$, such that $\tilde{G}$ has the same nodes as G; the arcs $\tilde{A}$ of $\tilde{G}$ are the arcs of G considered as undirected, i.e. $(m,n) \in \tilde{A}$ if and only if either $(m,n) \in A$ or $(n,m) \in A$. The graph $\tilde{G}$ is not in general acyclic, even if the graph G is acyclic.

A graph G is 2-DAG if G is a DAG such that for each node n of G there are at most 2 arcs of G of the form (m,n) where m is a node of G. A DAG G is connected if for any two nodes m and n of G there is a path from m to n in the undirected graph i.e. if there is a sequence $(m_0,m_1),\ldots,(m_{k-1},m_k)$ of nodes of G such that

$$m_0 = m$$

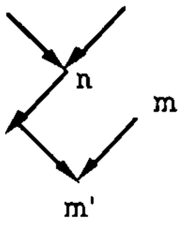
$$n_k = n$$

and, for each m_i, $i = 0, 1,\ldots,k-1$, either (m_i,m_{i+1}) or (m_{i+1},m_i) is an arc of G.

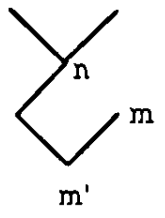
A DAG G is disconnected if it is not connected, i.e. if there are nodes m and n with no path between them in $\tilde{G}$.

An input node of G is a node n of G such that there are no arcs of G of the form (m,n), where m is a node of G. Let *I* denote the set of input nodes of G.

Note that in the following example there is no path between nodes n and m in G, but there is one in $\tilde{G}$, because (m',m) is an arc of $\tilde{G}$, but not of G.



G



$\tilde{G}$

A cut, K, of a graph G is a subset of the arcs of G whose removal disconnects G. More formally, let $A \supseteq K$ and let G_K be the graph whose nodes are V (G), the nodes of G, and whose arcs are A\K, then K is a cut of G if G_K is disconnected.

A cut disconnects a finite graph G into a finite number of connected components.

An input coloring of G is a function ψ_I: $I \to$ C, where C is the set of colors, and I the set of input nodes of G.

The pair (G,ψ_I) is an I-colored graph (resp., DAG) if G is a graph (DAG) and ψ_I an input coloring of G.

A colored graph (DAG) is a pair (G,ψ) where ψ: N $\to$ C maps the set of nodes of G to the set of colors, and G is a graph (DAG).

A coloring of an I-colored graph (DAG), (G,ψ_I) is a colored graph (DAG), (G',ψ'), where G' = G and $\psi|_I = \psi_I$, i.e. the coloring ψ agrees with ψ_I on the input nodes of G.

When G is given we can speak of the coloring function ψ as a coloring of G.

Let (G,ψ_I) be an I-colored graph (DAG), let K be a cut of G, and let $G_1,...,G_p$ be the components of G_K, where V(G_i) is the set of nodes of G_i, and A_i the set of arcs of G_i and I_i the set of input nodes of G_i, for i = 1,...,p.

Note that

$$V(G) = \bigcup_{i=1}^{P} V(G_i)$$

and

$$\bigcup_{i=1}^{P} I_i \supseteq I$$

The cut K is a color cut (c-cut) of (G,ψ_I) if:

1) for each i = 1,...,p $G_i \cap I \neq \emptyset$. I.e., every component of G(K) contains input nodes of G;

2) If m and n ∈ V(G) and m, and n ∈ $G_i \cap I$ for some I = 1, ..., p, then $\psi_I(m) = \psi_I(n)$. I.e., all inut nodes of G that belong to the same component of G(K) have the same color (are assigned to the same agent).

The size of a cut K, denoted either card K, or $|K|$, is the number of arcs in K.

Let (G,ψ_I) be an *I*-colored graph (DAG) and let ***K*** be the set of all c-cuts of (G,ψ_I). A cut K of (G,ψ_I) is a *minimal* cut if

1) K ∈ ***K***
2) | K | ≤ | K' | for all K' ∈ ***K***.

I.e., K is a c-cut of (G,ψ_I) and there is no other c-cut of (G,ψ_I) consisting of fewer arcs than K.

Let (G, ψ) be a colored digraph G, where ψ is the coloring function. A c-cut of (G, ψ) is a cut K that separated G into monochrome components, $G_1, \cdots, G_p$. I.e., for $i = 1, \cdots, p$, if u and v are elements of G_i then $\psi(u) = \psi(v)$.

A pair of vertices u, v in a digraph G form a parental pair, or briefly a p-pair if there are arcs (u,x) and (v,x) in the arcs,**A**(G), of G. I.e., vertices u and v are a parental pair if they have a child in common.

Let (T,ψ_I) be an I-colored(2,1)-tree T with input connections set I, let V(T) be the set of nodes of T, and **A** the set of arcs.

A partition P = $I_1, ..., I_p$ of I is an eligible partition of I if:

(i) If for some i ∈ {1,...,p}, v ∈ I_i, and there exists w ∈ I such that (v, w) form a p-pair (i.e., have a common child in T), then w ∈ I_i;

(ii) For every i ∈ {1,...,p}, if v and w ∈ I_i and (v, w) form a p-pair such that ψ_I(v) = ψ_I(w) = x, x ∈ C, and if u and z ∈ I_i also form a p-pair (u, z), then either ψ_I(u) = x or ψ_I(z) = x.

An eligible partition of I is a partition whose component sets consist of parental pairs that are x,x or x,y pairs, or of nodes that do not form a parental pair with any other node in I. The coloring of such "isolated" nodes of I is not restricted.

Given an eligible partition P = $I_1, ..., I_p$ of (T,ψ_I), a cut K of (T,ψ_I) is a P-cut if:

(i) K cuts T into p components $T_1, ..., T_p$,

(ii) $T_i \cap I = I_i$, for i = 1,...,p.

A component T_i of the P-cut K of T is an x-component if $T_i \cap I$ contains a p-pair colored x,x; it is an (x,y) component if all the p-pairs are colored x,y, for $x,y \in C$, $x \neq y$.
The following Lemma tell us that if (i) a (2,1)-tree T is given, (ii) a coloring of its input nodes is given, and a P-cut is given, then the coloring of input nodes can be extended to a coloring of T, and the P-cut K can be extended to a cut that disconnects T into monochrome components, i.e., a c-cut.
Lemma 8. A P-cut K of (T,ψ_I) can be augmented to form a c-cut of (T,ψ_I).

Proof. Let K be a P-cut of (T, ψ_I), and let $P = T_1,...,T_p$ be the partition determined by the P-cut K. For $i \in \{1,...,p\}$ if T_i is an x-component, color all its vertices not already colored with the color x. I.e., if v is a node in $T_i \backslash I_i$, let $\psi(v) = x$.

If T_i is an (x,y)-component, and the number of vertices in $T_i \cap I = I_i$ colored x is larger than the number colored y, then give all uncolored nodes in $T_i \backslash I_i$ the color x. I.e., if, $|\{v \in I_i | \psi_I(v) = x\}| \geq |\{v \in I_i | \psi_I(v) = y\}|$ then for all $n \in T_i \backslash I_i$; let $\psi(n) = x$; otherwise $\psi(n) = y$ for all $n \in T_i \backslash I_i$. (Note that the number of nodes in I_i colored x is larger than the number colored y if and only if the number of x-nodes in I_i not part of a p-pair is larger than the number of y-nodes not part of a p-pair.)

We now augment the P-cut K to a cut $K' \supset K$ as follows.

1. For all i such that T_i is an x-component include in K' all arcs that originate at nodes $v \in I_i$ with $\psi_I(v) = y \neq x$.

2. Recall that if T_i is an (x,y)-component then all its nodes not in I_i have been assigned the same color.

For all i such that T_i is an (x,y)-component colored x, augment K by including all arcs originating at y-nodes in I_i; if the nodes of $T_i \backslash I_i$ are colored y, then augment K by including all arcs originating at x-nodes in I_i.

This process increases the number of components determined by the augmented cut K' by exactly the number of nodes that originate arcs in the augmenting cut $K' \backslash K = \{a \in A | a \in K' \text{ and } a \notin K\}$.

It is easily verified that K' satisfies all the conditions for a c-cut. []

The process described in the proof of Lemma 8 provides a natural and convenient way of going from an eligible partition to a c-cut. This appears to be a good way of finding minimal c-cuts, and in graphs in which minimal c-cuts are not unique, of finding all minimal c-cuts.

In *I*-colored binary (rooted) trees, it seems often to be the case that the number of eligible partitions that need to be explored is relatively small.

We turn now to some examples to illustrate the concepts and procedure described in Lemma 8.

Some examples:

Example 1.

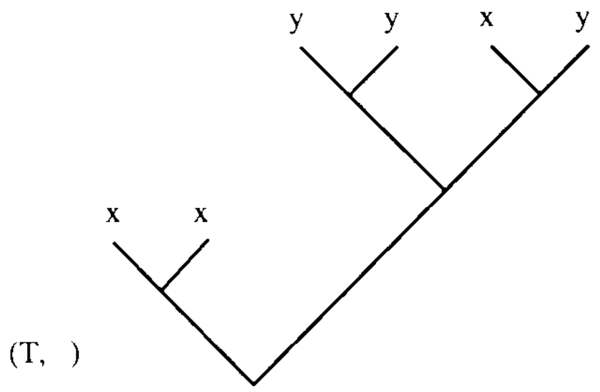


FIGURE AI-7

It is convenient to represent this colored binary (rooted) tree in the form:

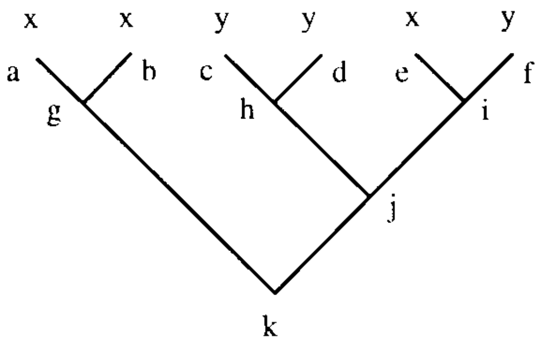


FIGURE A-8

so that the set *I* and the tiers of nodes are clearly represented.

The nodes are labelled a, b, c,...,k. The nodes {a,b,c,...,f} = *I*, are the input connections, g, h and i are input nodes.

Eligible partitions of *I* are:

P_1 = {ab} {cd} {ef}

P_2 = {abef} {cd}

P_3 = {ab} {cdef}.

The associated P-cuts are as follows:

1. There are three P_1-cuts:

(i) {(h,j),(i,j)}

(ii) {(h,j),(j,k)}

(iii) {(h,j),(g,k)}.

2. There is one P_2-cut, namely {(h,j)}.

3. There are two P_3-cuts:

   (i) {(j,k)}

   (ii) {(g,k)}.

It is clear that P_1 cannot lead to a minimal c-cut, and that both P_2 and P_3 lead to minimal c-cuts when augmented as prescribed in the proof of Lemma 8. The partition P_2 leads to the augmented cut K'_2 = {(j,k),(f,i)}, with |K'| = 2, and hence to the coloring,

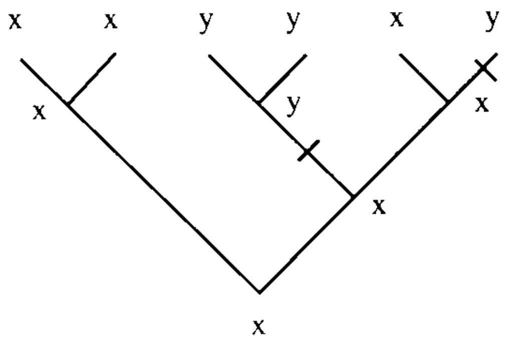
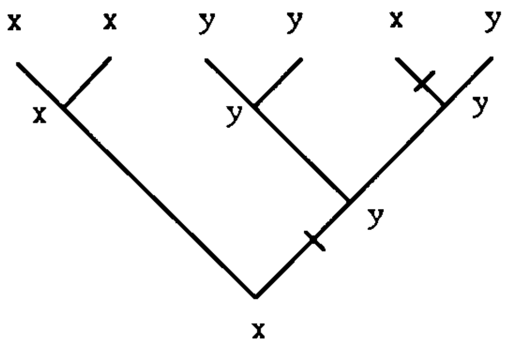


FIGURE AI-9

Here |x| = 4, |y| = 1.

Partition P_3 leads to K'_3 = {(e,i),(j,k)}, and hence to the coloring,



$|K_i' 3 = 2, |x| = 2, |y| = 3$

FIGURE AI-10

and K= {(e,i),(g,k)}, shown below

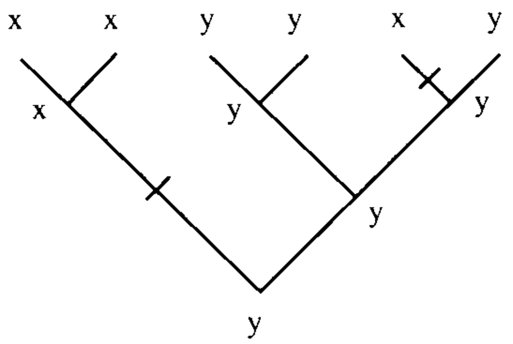


FIGURE AI-11

where |K"$_3$)| = 2, and |x| = 1, |y| = 4.

While all three c-cuts are minimal, we will see below that only the cut K$_3$' leads to an efficient assignment, when the time of computation is taken into account.

Application of the 2-pass coloring algorithm described in section II(1) to this example gives the following coloring.

1st pass.

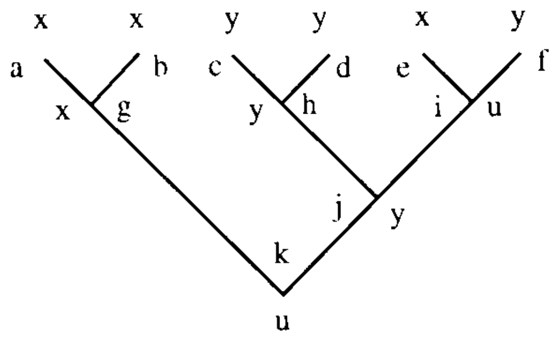
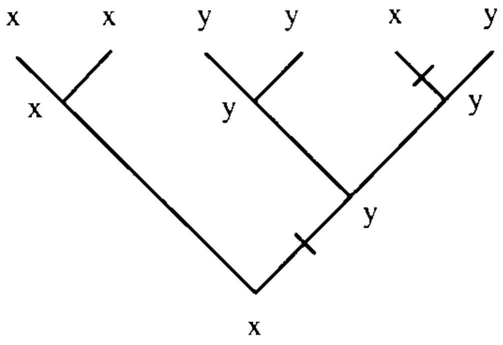


FIGURE AI-12

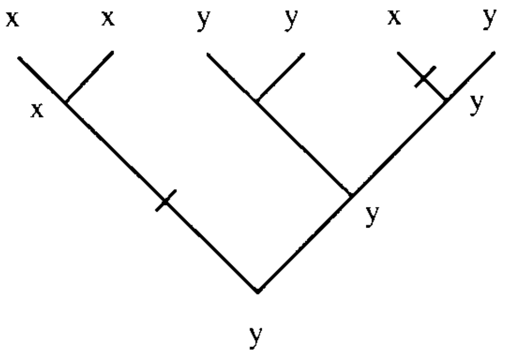
2nd pass. If $\psi(k) = x$, we get



$= 2, \ |x| = 2, \ |y| = 3$

FIGURE AI-13

If $\psi(k) = y$ we get

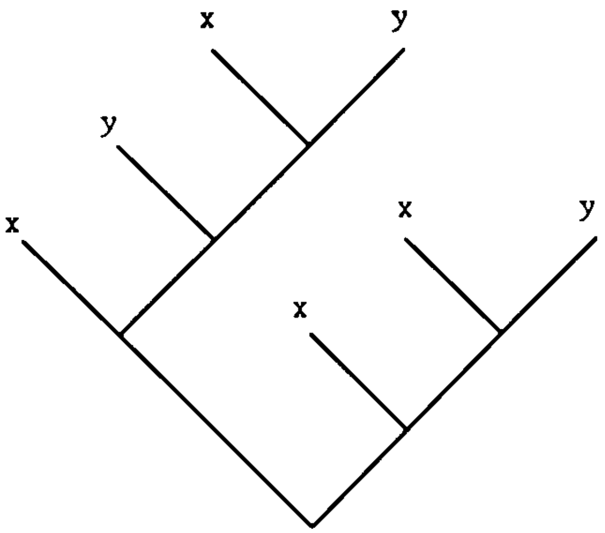


$= 2, \ |x| = 1, \ |y| = 4.$

FIGURE AI-14

Note that the 2 pass algorithm cannot yield the coloring corresponding to K_2'.

Example 2. Starting with the graph,



we represent it in the form,

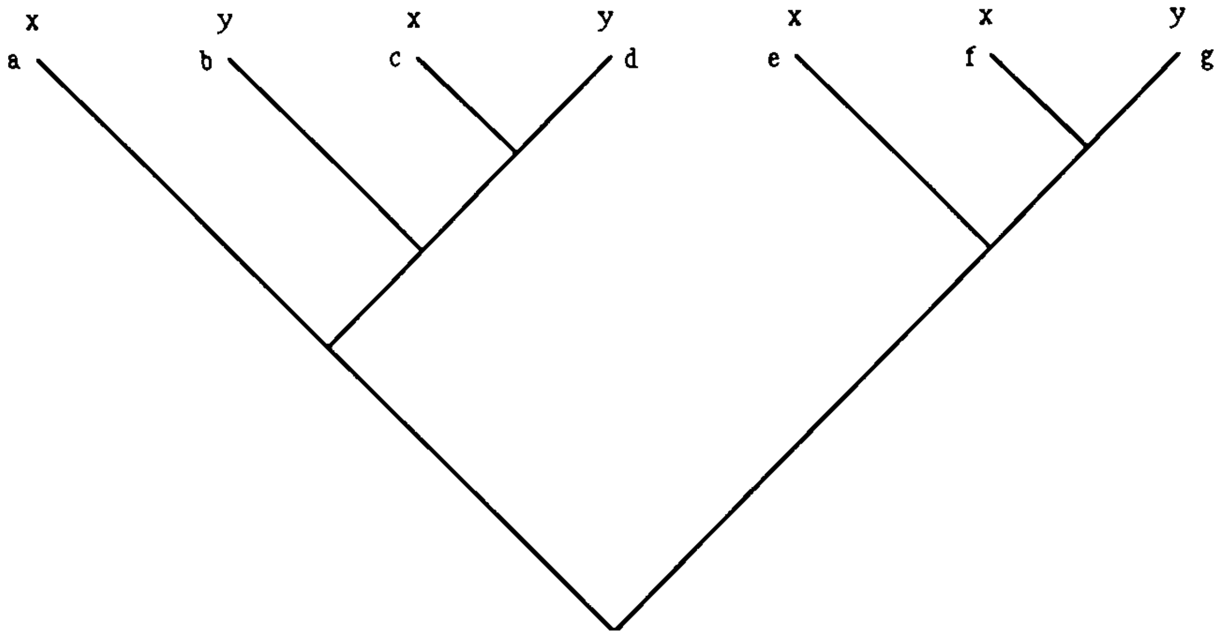


FIGURE AI-15

Some possible partitions are:

1. $P_1 = I$
2. $P_2 = \{abcd\}, \{efg\}$
3. $P_3 = \{a\}\ \{bcd\}\ \{efg\}$

P_1 determines the augmented c-cut K, and coloring shown in Fig. AI-16.

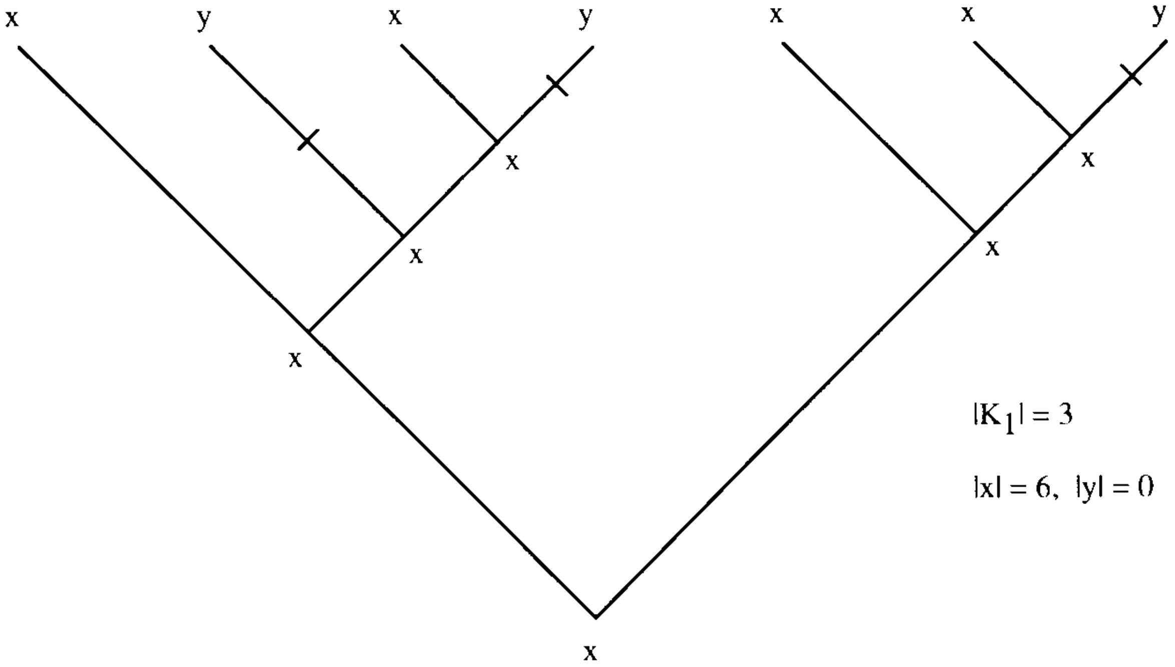
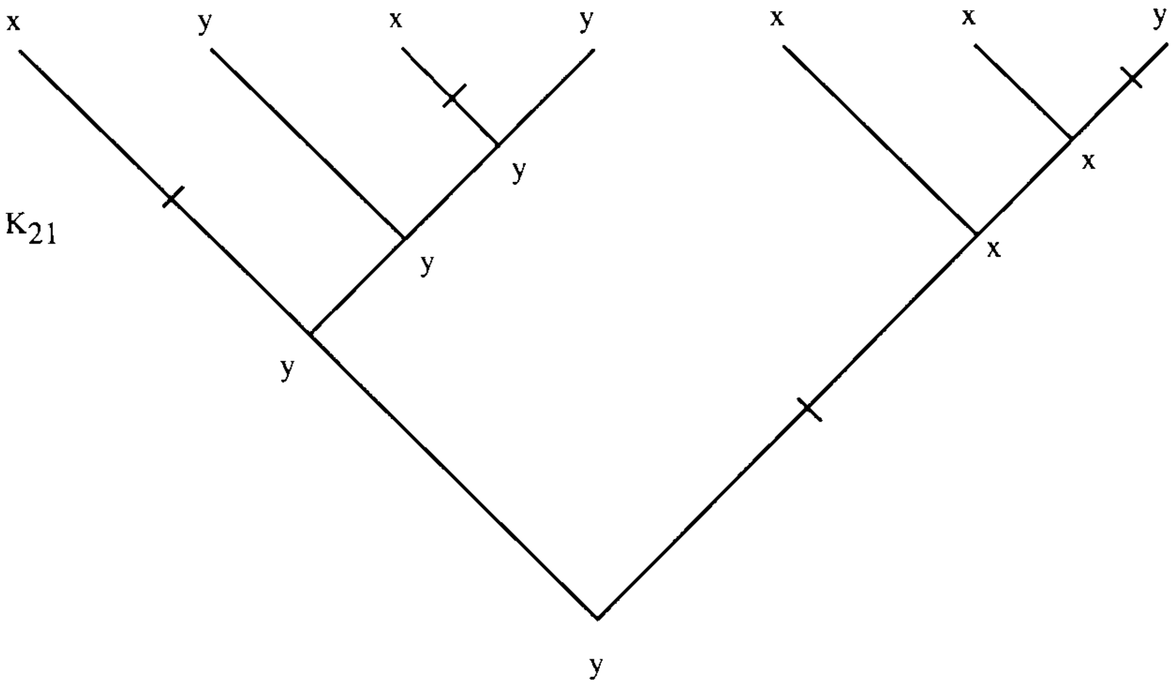


FIGURE AI-16

P_2 determines K_{21}, or K_{22}, shown in Fig. AI-17.



or

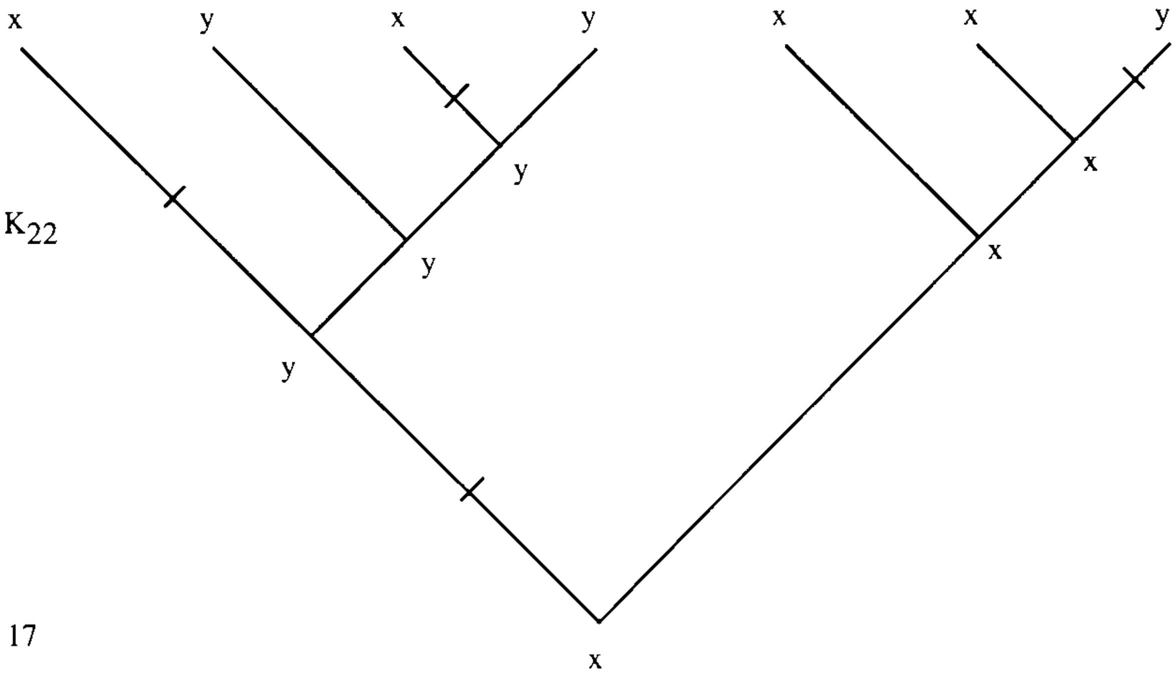
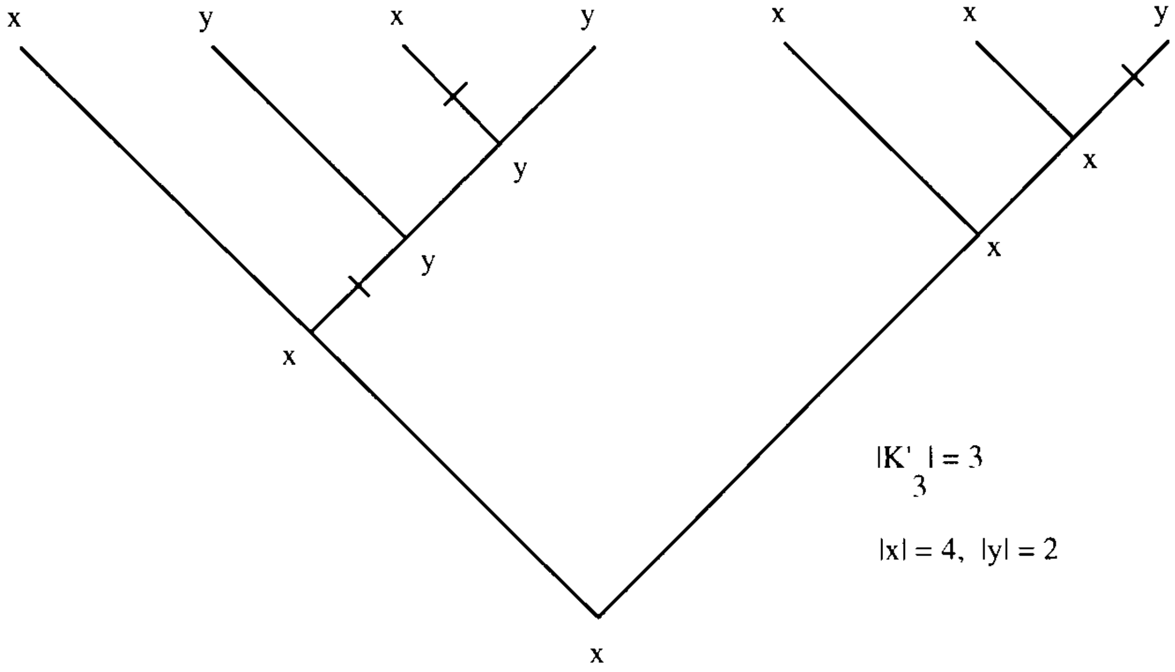


FIGURE AI-17

|K| = |K| = 4.

P_3 leads to Kshown in Fig. AI-18.



$|K'_3| = 3$

|x| = 4, |y| = 2

FIGURE AI-18

The 2-pass coloring procedure applied to this example leads to the coloring shown in Fig. AI-20. (Figure AI-19 shows the result of the first pass:)

1st pass.

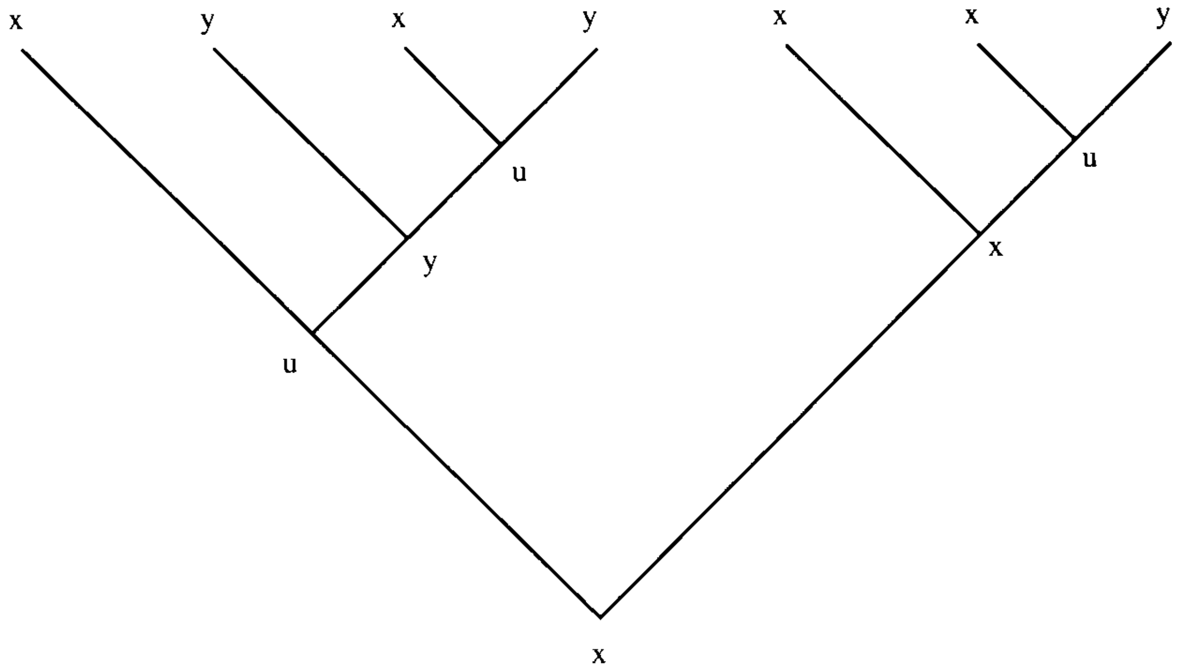
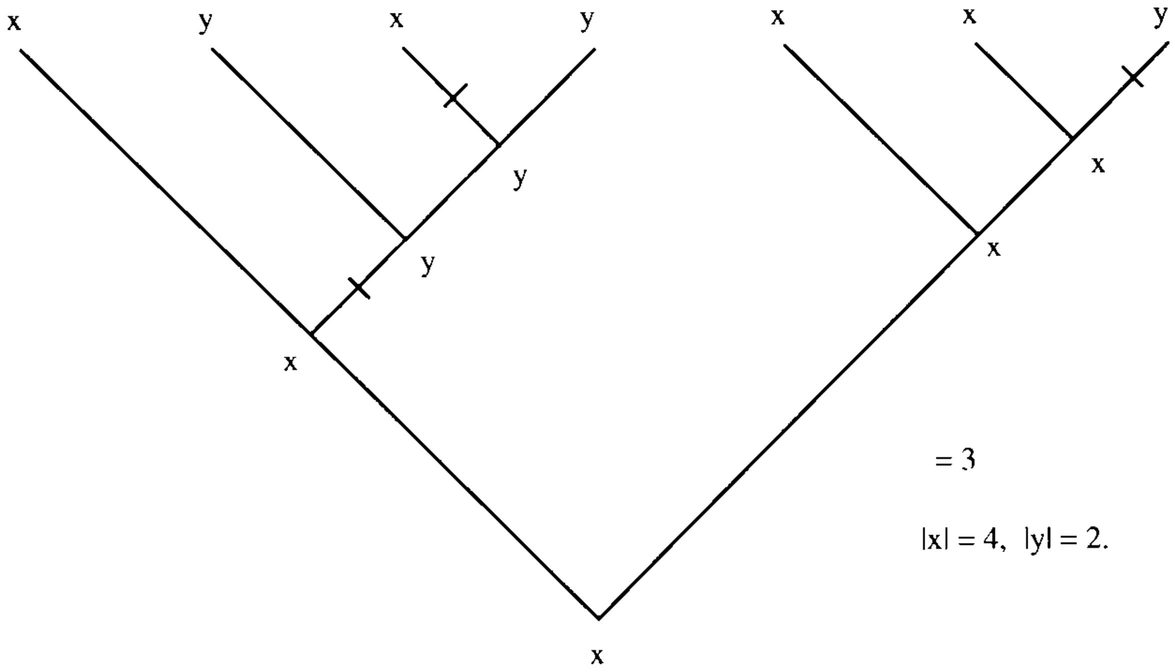


FIGURE AI-19

2nd pass.



= 3

|x| = 4, |y| = 2.

FIGURE AI-20

I.e., the c-cut determined by the 2-pass coloring procedure is K. It leads to an efficient point in this example.

Example 3:

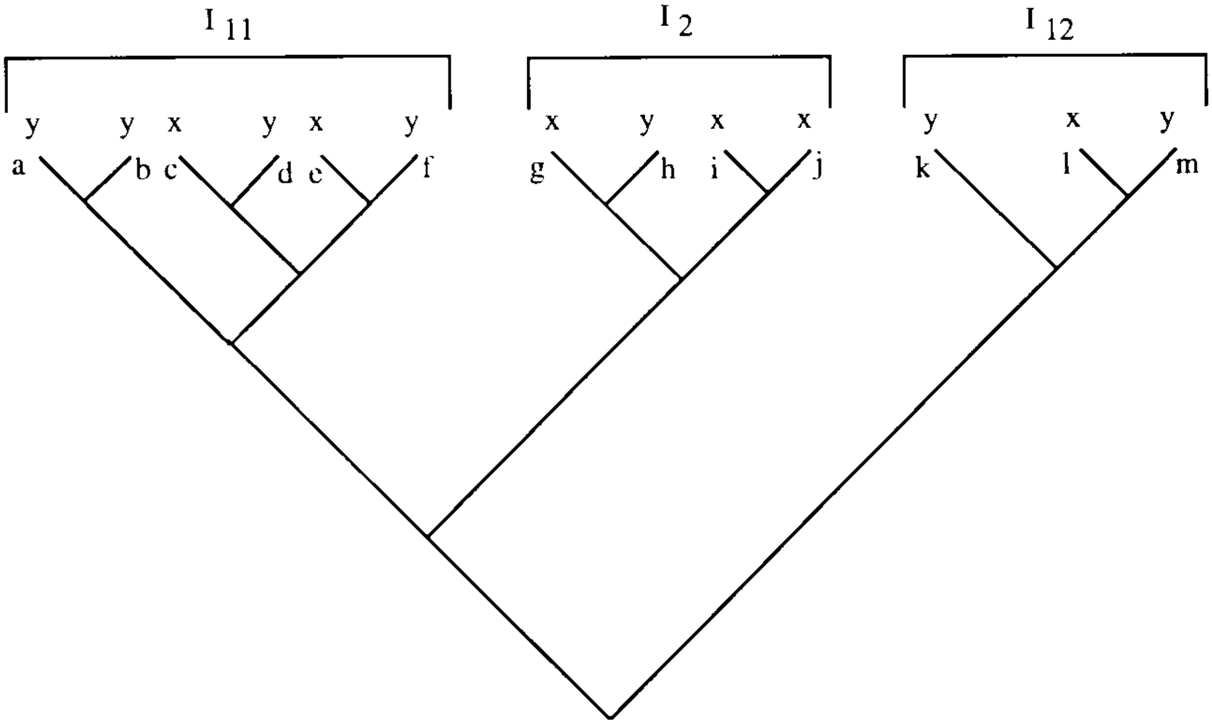


FIGURE AI-21

In this example the following eligible partition of *I* is particularly interesting

$$P_1 = \{abcdefklm\}, \{ghij\} = (I_{11} \cup I_{12})), I_2,$$

where

$$I_{11} = \{abcdef\}, I_{12} = \{klm\}, I_2 = \{ghij\}$$

This leads to the c-cut K, and the coloring function ψ shown in Figure AI-22.

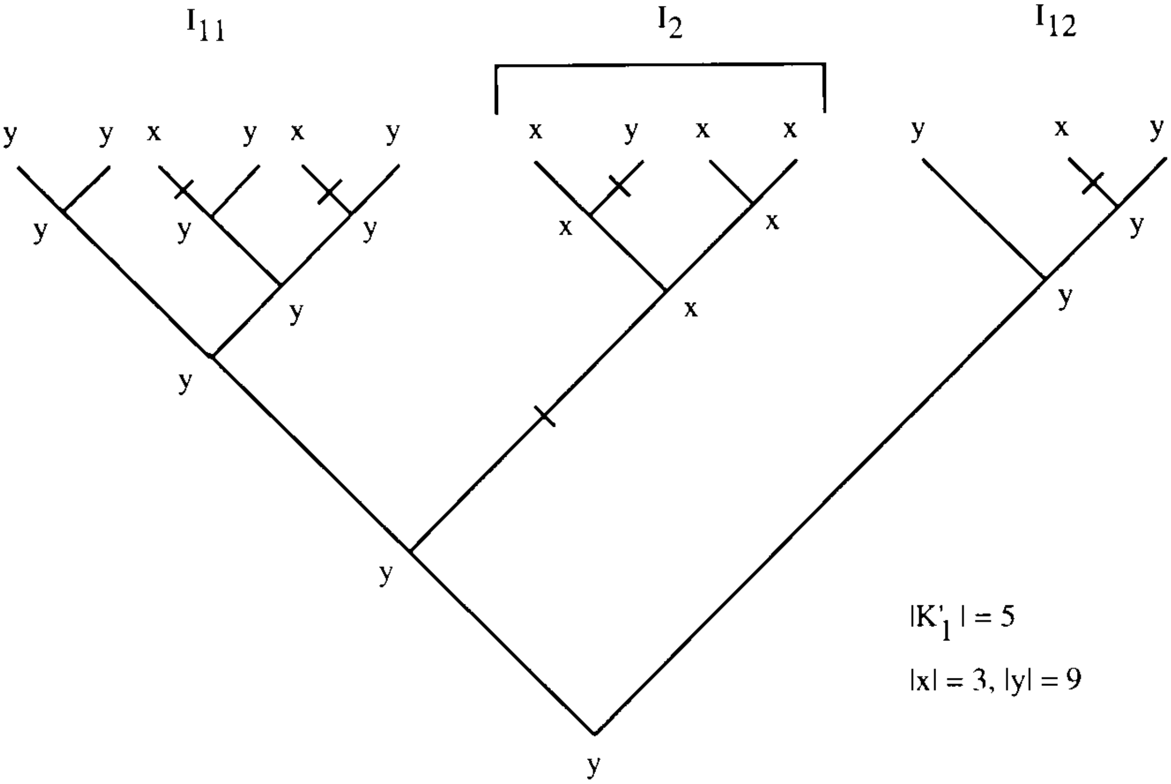


FIGURE AI-22

The 2-pass coloring algorithm applied to this example yields the coloring in Figure AI-24. (Figure AI-23 shows the result of the first pass.)

1st pass

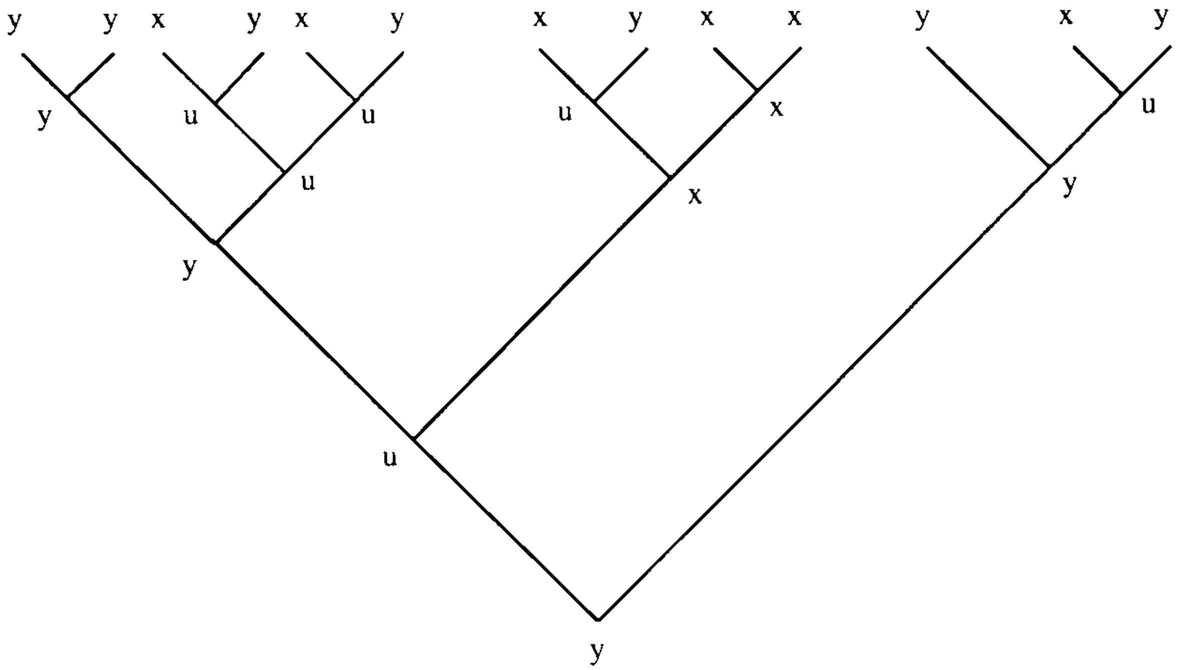
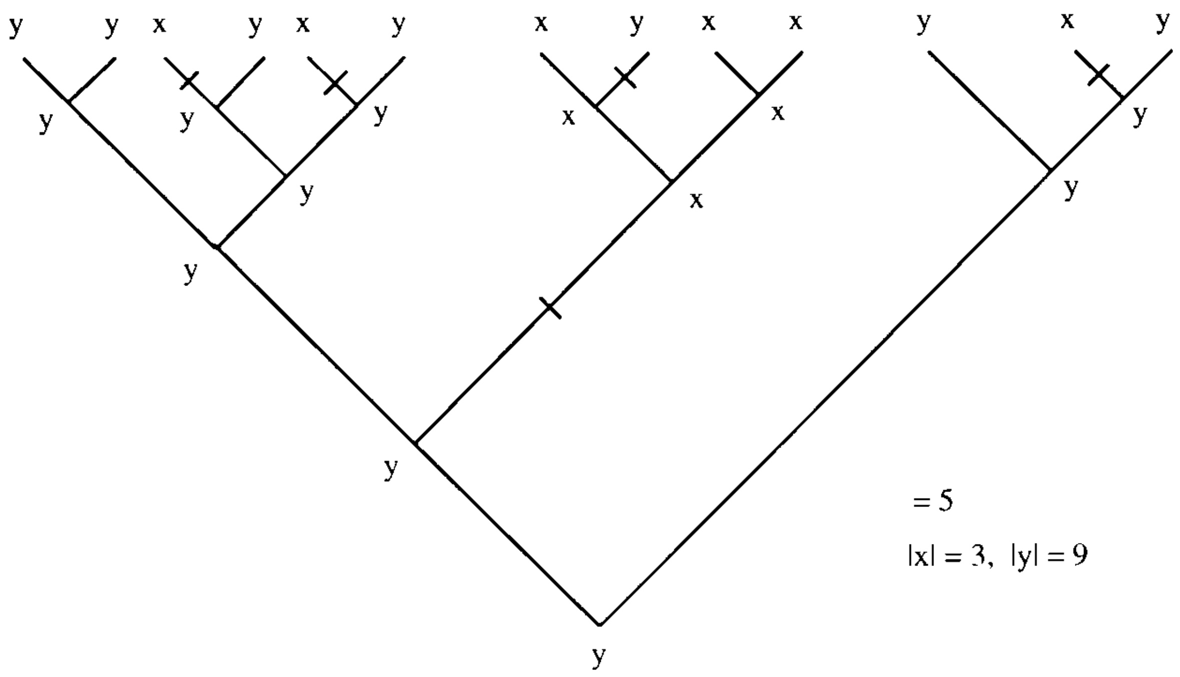


FIGURE AI-23

2nd pass



= 5

|x| = 3, |y| = 9

FIGURE AI-24

This is, of course the same coloring as ψ, and hence determines the same c-cut K.

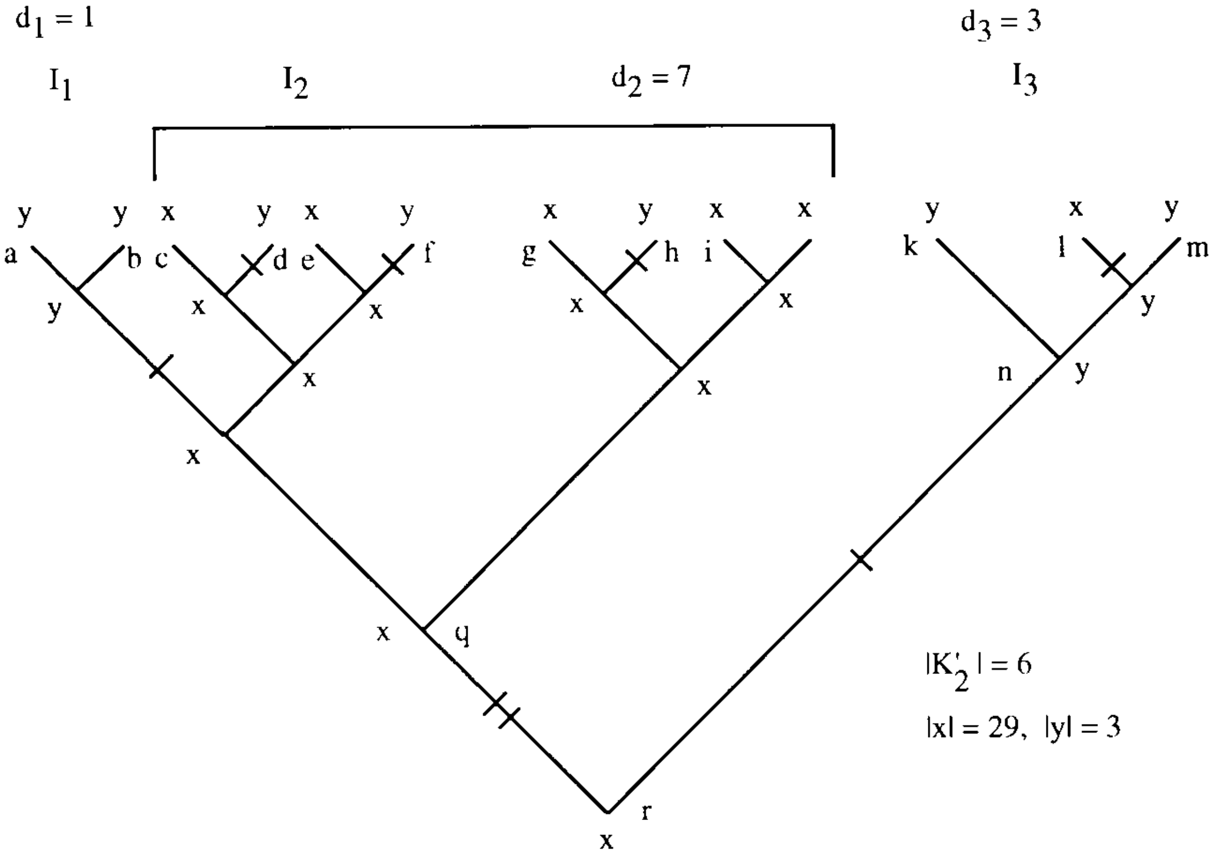


FIGURE AI-25

Another partition of *I* of interest is

$$P_2 = I_1, I_2, I_3 = \{ab\}\ \{cdefghij\}\ \{klm\}.$$

Here I_1 is a y-component, I_2 an x-component and I_3 an (x,y)-component. This partition leads to two c-cuts, K_2^1 consisting of the arcs marked with a slash in Figure 25, and K_{21}^1 which differs from K_2^1 in that the arc from node n to r is replaced by (q,r). This changes the color of r from x to y. Neither of these c-cuts is minimal, each containing 6 arcs, but K_{21}^1 has |x| = 8, |y| = 4.

The difference between the eligible partitions P_1 and P_2 is that P_2 has more components than P_1.

Definition. An eligible partition

$$P = I_1, I_2, \ldots, I_p$$

is a minimal eligible partition if it has no more components than any other eligible partition, i.e., $P = I_1, I_2, \ldots, I_p$ is a minimal eligible partition of I if

(i) P is an eligible partition of I and;

(ii) if $P' = I'_1, \ldots, I'_{p'}$ Is an eligible partition of I, then $p \leq p'$.

Conjecture: Let (T, ψ_I) be an I-colored (2,1)-tree. If P is a minimal eligible partition of I, then the augmented P-cut K' is a minimal c-cut of (T, ψ_I).

Notice that the P-cut determined by an eligible partition $P = I_1, \ldots, I_p$ must divide T into exactly p components $T_1, \ldots, T_p$.

Thus, in Example 3 above, the partition, P, of I given by

$$I_1 = \{abcdk\}, I_2 = \{efghijlm\}$$

and

$$P = I_1, I_2,$$

is not an eligible partition of I, because although I_1 is a y-component and I_2 an x-component, we cannot find 2 connected components T_1 and T_2 with

$$T_1 \quad I = I_1 \text{ and } T_2 \quad I = I_2.$$

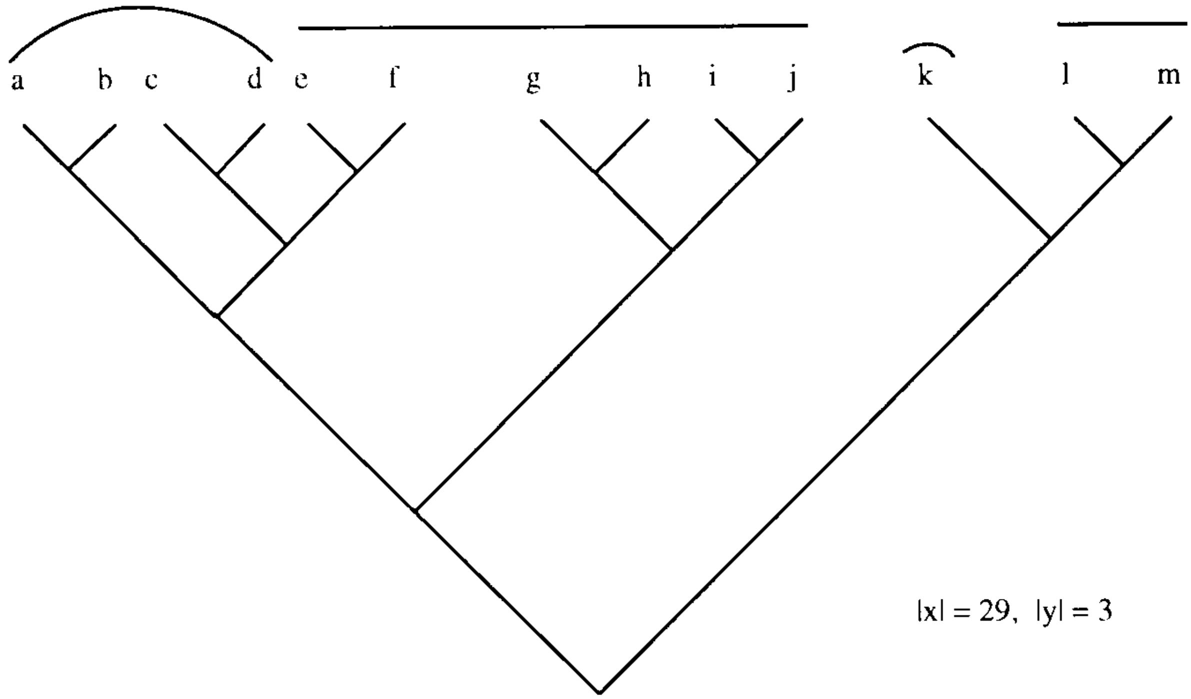


FIGURE AI-26

We can see in Figure 26 that any connected component that includes {abcd} and {k} must disconnect {lm} from {ghij}. Alternatively, any connected component that includes {ghijlm} must disconnect {k} from the remaining nodes in I_1.

The next section of this Appendix pursues the problem of finding minimal colorings in a special class of directed acyclic (rooted) graphs, briefly, DAGS, called Diamond DAGS. Because these graphs are not trees, the material in Appendix II does not depend on this one, so it can be skipped without loss of continuity.

On Diamond DAG's

We consider a special class of directed acyclic rooted graphs, as follows. Let a set of input connection nodes *I* be given. Suppose that two of these nodes, called the left end node a and the right end node b, are specified, that another node r, the root, is also specified.

If there are n nodes in *I*, then the graph consisting of nodes shown in Figure AI-27 is called a complete 2,2-DARG (directed acyclic rooted graph) on n input connection nodes.

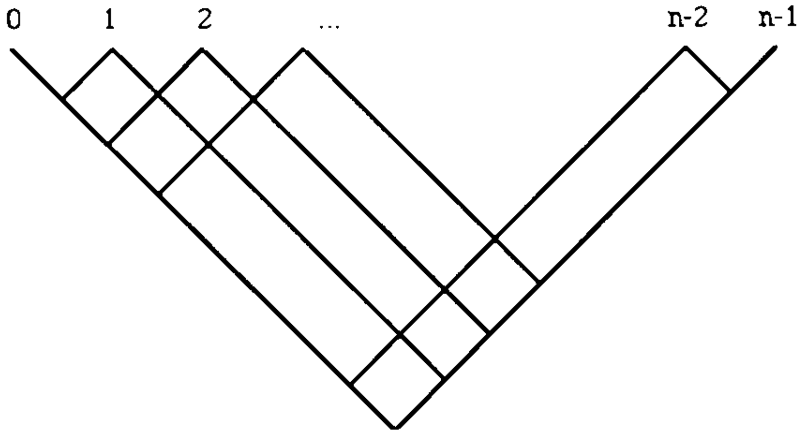


FIGURE AI-27

Every interior node has in degree 2 and out degree 2. The nodes along the lines connecting the left and right end nodes, respectively, to the root have in degree 2 and out degree 1.

Let (G,ψ_I) be a complete 2,2-DARG on n+1 input connections. We can associate to each input connection in *I* two integers i and j, satisfy the conditions $i + j = n$, $0 \leq i, j \leq n$. We can regard such a pair as the label of the node to which it is associated.

The first integer i of the pair (i,j) is the "distance" of the node from the left end node and the second, j, is its "distance" from the right end node, each intervening node counting as 1 unit of distance.

The node labelled (i+1, j-1) is the immediate right hand neighbor of (i,j), and (i-1, j+1) is its left hand nieghbor.

A subset of input connection nodes, $I \supset \mathrm{I}$, is contiguous in a graph G (contiguous in G), or an interval, if, whenever it contains two nodes, it contains all the nodes of I that lie between them. More formally, $I \supset \mathrm{I}$ is contiguous in G, or an interval, if

(i,j) $\in$ I and (i',j') $\in$ I, with i < i'and j > j', and if (i",j") $\in$ I
with i < j" < i' and j' < j" < j, then (i",j")$\in$ I.

(Here (i,j) $\in$ I means "the node labeled (i,j) is an element of I".)

Suppose I is an interval in I. Node (i,j) is its left end point if

(i,j) $\in$ I and (i-1,j+1) I.

Similarly (i',j') is the right end point of I, if

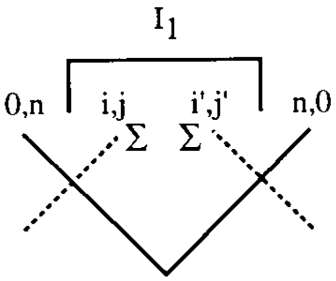
(i',j') $\in$ I and (i+1,j-1) I.

Lemma. Let G be a 2,2-DARG with input connection set I. Suppose $I \supset \mathrm{I}_1$ is an interval, and $\mathrm{I}_2 = I \setminus \mathrm{I}_1$ is its complement in I. There are 8 cuts that determine a component G_1 with the property that

$\mathrm{G}_1 \cap I = \mathrm{I}_1$.

These 8 cuts are shown in the following table.

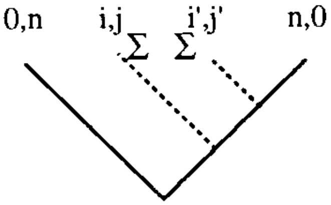
Table Name	Formula for size of cut
symmetric endcut	$i + j'$



1)

FIGURE AI-28

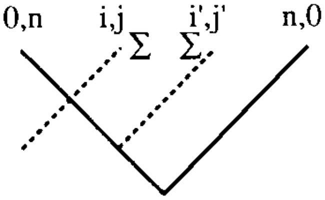
right 2-cut	$j + j' + 1 = (j+1+j')$
	(provided $i \neq 0$, $j' \neq 0$)



2)

FIGURE AI-29

left 2-cut	$i + i' + 1$
	(provided $i \neq 0 \neq j'$)



3)

FIGURE AI-30

center cut $j + 1 - j' + i' + 1 - i$

$= j - j' + i' - i + 2$

$= (j-i) - (j'-i') + 2$

(provided $i \neq 0 \neq j'$)

4)

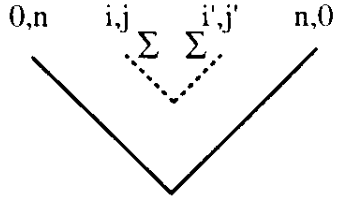


FIGURE AI-31

$i' + 1$ if $i' \neq n$

0 otherwise

5a)

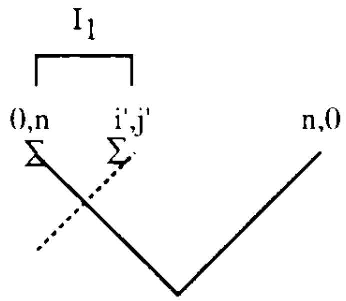


FIGURE AI-32

j' $j' \neq n$

0 otherwise

Suppose the interval I_1 has endpoints i,j and i',j' with i ≠ 0 and j' ≠ 0. Then there are four possible cuts, namely 1), 2), 3) and 4) in the preceeding table. These may be compared as follows.

Size of Cut 1		Size of Cut 2
$i + j'$	$<$	$j + 1 + j'$
i	$<$	$j + 1$

Since $i = n - j$

$n - j$	$<$	$j + 1$

$$\frac{n-1}{2} < j$$

Therefore, size of cut 1 is less than size of cut 2 if and only if $j > \frac{n-1}{2}$. Comparing the remaining cuts, we see that:

Size of Cut 1		Size of Cut 3
$i + j'$	$<$	$i + i' + 1$
j'	$<$	$i' + 1$
$n - i'$	$<$	$i' + 1$

$$\frac{n-1}{2} < i'$$

Size of 5a		Size of 5b
i' + 1	<	'

$$i' + \frac{n-1}{2}$$

If j' = 0 we have

Size of 6a		Size of 6b
i	<	j + 1

$$i < \frac{n-1}{2}$$

Suppose G, ψ_I is a complete 2,2-DARG on n + 1 nodes of *I*, labelled as above. We define a regular partition of *I* into maximal monochrome intervals as follows.

Let I_1 contain the node labelled 0,n. Suppose $\psi_I(0,n) = x$. Then I_1 contains i, n - i if and only if $\psi_I(i,n-i) = \psi_I(i-1,n-i+1) = x$. I.e., I_1 is the largest interval that contains 0,n and all nodes of the same color as 0,n.

Suppose i,j is the last node in I_1, (i.e., if i',j' $\in I_1$ then i' $\leq$ i). Let I_2 be the largest interval that contains i + 1, j - 1, whose color $\psi_I(i+1,j-1) = y$, where $y \neq x$, and such that all of its nodes have the color y. Continue in this fashion until I_P is reached, where I_P contains the node n,0 and is the largest monochrome interval to do so.

For brevity we refer to P so defined as a monochrome partition of *I*, or an M-partition of *I*.

The c-cut K with components $G_1,\ldots,G_q$ is associated with an M-partition

Example 2

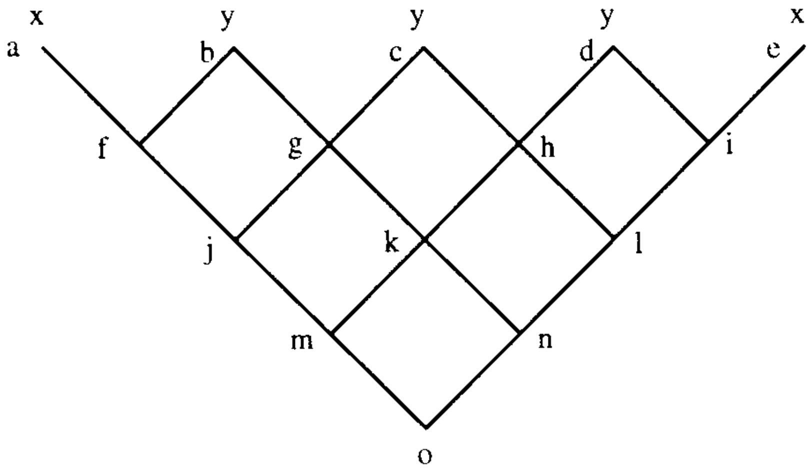


FIGURE AI-37

I = {a,b,c,d,e}

K_1 = {(a,f),(e,i)} K_1 = 2

K_2 = {(b,f),(g,j),(k,m),(k,n),(h,l),(d,i)} K_2 = 6

K_3 = {(a,f),(d,i),(h,l),(h,n),(m,o)} $|K_3|$ = 5

Example 3

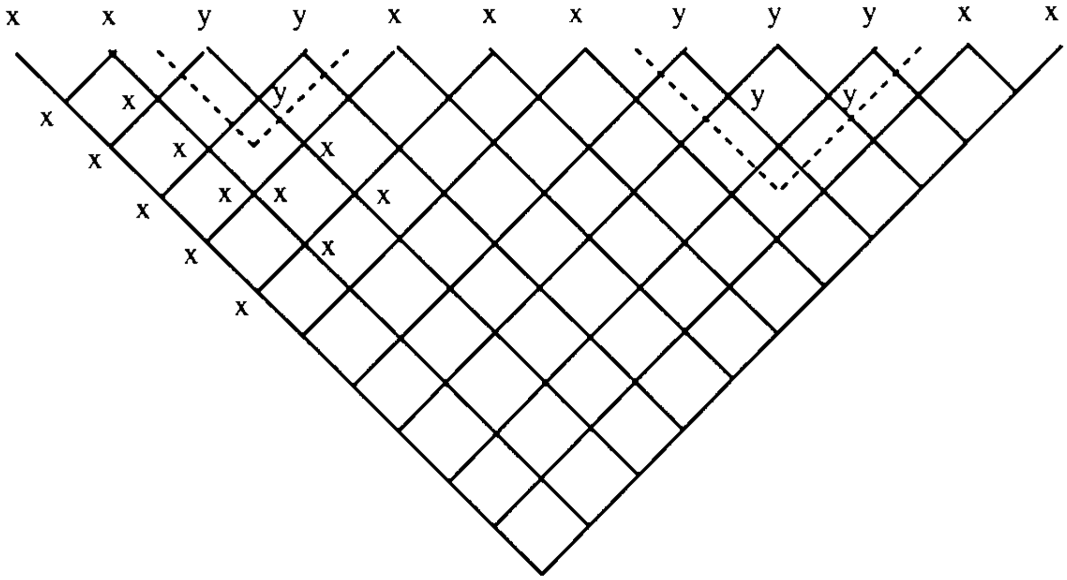


FIGURE AI-38

A sharper bound is as follows. If i is odd, let $|I_i| = \alpha_j$, where $i = 2j - 1$, $j = 1,2,\ldots,s_1$ and $i \leq p$; and if i is even, let $|I_i| = \beta_j$ $i = 2j$, $j = 1,2,\ldots,s_2$ and $i \leq p$.

Consider two cases: 1) p is even.

2) p is odd.

If p is even, then $\psi_I(I_1) \neq \psi_I(I_p)$;

if p is odd, then $\psi_I(I_1) = \psi_I(Ip)$.

Among the c-cuts associated with an M-partition P, the following two are of interest.

K_1: isolate each of the odd intervals $I_3, I_5, \ldots$, (except those colored x), by a "center cut," (cut 4 in the table) and isolate I_1 by a left end cut, (cut 5a in the table).

K_2: isolate each even interval $I_2, I_4, \ldots, I_{p-1}$ by a center cut and I_p by a right end cut.

The size of K_1 is given by $|K_1| = 2\ (\alpha_2 + \ldots + \alpha_{s_1} - 1) + \alpha_1 + \alpha_{s_1}$ provided $\alpha_1 < n + 1$.

The size of K_2 is given by $|K_2| = 2\,(\beta_1 + \ldots + \beta_{s_1}) + p$.

So the smaller of these two numbers is an upper bound for the size of a minimal c-cut. This bound is easily calculated, and once calculated enables us to eliminate a priori possible c-cuts associated with P.

If p is odd, then the M-partition P has the form

$$\underline{|\mathrm{I}_1|}\underline{I_2|}\cdots\underline{|I_p|}$$

where $\psi_I(I_1) = x$, $\psi_I(I_p) = y$, i.e. for $i = 1,\ldots,p$, $\psi_I(I_i) = \begin{cases} x & \text{if } i \text{ is odd} \\ y & \text{if } i \text{ is even} \end{cases}$

While if p is even, $\psi_I(I_1) = x = \psi_I(I_p)$. If p is odd, the odd intervals are marked in the figure is immediately seen to exceed the bound given by $|K_1|$ or $|K_2|$ in that example. This rules out about half the a priori possible c-cuts associated with P.

In the examples given above, this screening leads to the following

Example 1. $I_1 = \{a\}, I_2 = \{b\}, I_3 = \{c\}, I_4 = \{d\}, I_5 = \{e\}$. In each case the size of I_i is 1. Therefore

$$|K_1| = a_1 + 2a_2 + a_3 = 1 + 2 + 1 = 4$$

$$|K_2| = 2b_1 + 2b_2 = 2\bullet 1 + 2\bullet 1 = 4.$$

All the c-cuts shown in this example are minimal.

Example 2. $I_1 = \{a\}, I_2 = \{b,c,d\}, I_3 = \{e\}$

$$|K_1| = a_1 + a_2 = |I_1| + |I_3| = 2$$

$$|K_2| = 2b_1 = 2|I_2| = 6.$$

Therefore $|K_1| = 2$ is an upper bound for minimal c-cuts. It is the minimum in this example.

Example 3. $I_1 = \{a,b\}, I_2 = \{c,d\}, I_3\{e,f,g\}, I_4 = \{h,i,j\}, I_5 = \{k,l\}$

$$|K_1| = a_1 + 2a_2 + a_3 = 2 + 6 + 2 = 10$$

$$|K_2| = 2b_1 + 2b_2 = 4 + 6 = 10.$$

Therefore, the upper bound is 10.

given (T,ψ). I.e., $\hat{\lambda}(T,\psi)$ is a <u>minimal assignment</u> if $\tau_{\hat{\lambda}} \leq \tau_{\lambda}$, for all schedules λ of (T,ψ).

<u>Necessary Conditions for a Minimal Schedule</u>

If an assignment λ of the nodes of the colored network (T,ψ) has gaps, then it cannot be a minimal assignment.

Proof of the proposition that a minimal assignment must be free of gaps is immediate; if λ has a gap, subtract 1 from each value of λ that follows the first gap and repeat for each subsequent gap in numerical order.

Let (T,ψ) be given.

Let

$$\bar{N}^{r} \equiv \bar{N}^{r}_{T,\psi} = \{ j \in V(T) \mid \text{j is not the root of T and } \psi(j) = r\}$$

i.e. $\bar{N}^{r}$ is the set of nodes of T, other than the root, that are colored r by the coloring function, ψ, and let

$$\bar{N}^{b} \equiv \bar{N}^{b}_{T,\psi} \equiv \{\text{nodes j of T} \mid \text{j is not the root of T and } \psi(j) = b\}.$$

The color x is called the <u>majority color</u> of the colored tree (T,ψ) if

card $(\bar{N}^{x}) \geqq$ card $(\bar{N}^{y})$ where $x \in \{r,b\}$ $y \in \{r,b\}$ and $y \neq x$.

(In the case of equality either colored is a majority color.)

Suppose for definiteness that b is the majority color (r is the minority color).

, <u>Definition</u>: A schedule λ of the colored tree (T,ψ) has the <u>matching property</u> (<u>Property M</u>) if for every $n \in \bar{N}^{r}$, there exists $n' \in \bar{N}^{b}$ such that $\lambda(n) = \lambda(n')$. I.e., λ

Nodes are labeled by two integers as follows. The first integer indicates the tier of the tree T on which the node is located, starting with the tier containing the leaves, or input nodes, of T.*) The second integer, separated by a point or large dot from the first, indicates the rank of the node on that tier, numbering them from the left. Thus, for example, in the tree shown in figure AII-1 the nodes are numbered as shown.

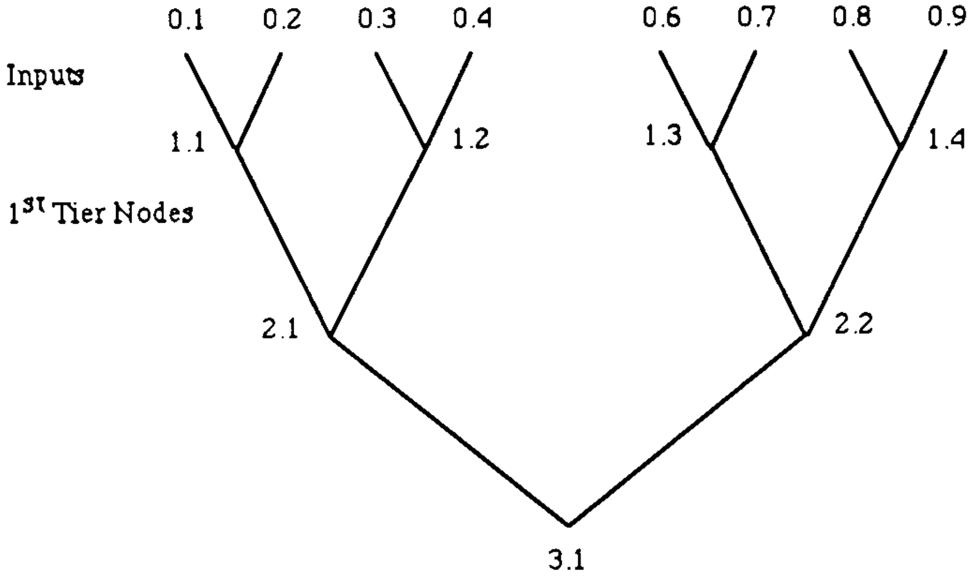


FIGURE AII-1

Inputs are denoted 0.1, 0.2, etc., following the same ordering scheme. The lexicographic ordering of nodes, i.e. i.j < i'.j' if i < i' or i = i' and j < j', is a complete ordering of the nodes of T. For ease of reading we shall usually write i.j. as $i \bullet j$.

A regular tree with L tiers has 2^{L-1} nodes on the first tier 2^{L-2} on the second, 2^{L-k} on the kth tier; $1 \leq k \leq L-1$, and 2^L inputs on the 0th tier nodes.

*) We refer to the levels of the unassigned tree T as "tiers", starting the numbering from the level of input nodes, to distinguish them from the levels assigned to nodes by an assignment function.

$$\lambda_\sigma\colon\ A_\sigma \rightarrow \{0, q_\sigma^* + 1, q_\sigma^* + 2, \ldots\}$$

assigns nodes of A_σ (eligible at stage σ) to levels, $(\lambda(i \bullet j) \neq 0)$ or carries them over to stage $\sigma + 1$, $(\lambda_\sigma(i \bullet j) = 0)$.

$$A_{\sigma+1'} = \{i \bullet j \in N \mid \text{(i) } i \bullet j \in A_\sigma \text{ and } \lambda_\sigma(i \bullet j) = 0, \text{ or, (ii) } i \bullet j \notin A_\sigma \text{ and } \exists \sigma' \leq \sigma \text{ such that } \lambda_{\sigma'}(i' \bullet j') \neq 0, \text{ and } \lambda_\sigma(i'' \bullet j'') \neq 0, \text{ where } i' \bullet j' \text{ and } i'' \bullet j'' \text{ are parent nodes of } i \bullet j\},$$

and

$$\lambda_{\sigma+1}\colon\ A_{\sigma+1'} \rightarrow \{0, q_\sigma^* + 1, q_\sigma^* + 2, \ldots\}.$$

It remains to specify the stage schedule functions λ_σ, $\sigma = 1,2,\ldots.$ This is done by a sequence of steps within each stage.

The first stage schedule function λ_1 is defined as follows. Let

$$A_1^{rb} = \{(1 \bullet j, 1 \bullet j') \in A_1 \times A_1 \mid 1 \bullet j \text{ and } 1 \bullet j' \text{ form an (r,b)-p-pair}\}$$
$$= \{(1 \bullet j, 1 \bullet j' \in A_1 \times A_1) \text{ (i) } \exists\, 2 \bullet k \in N \text{ such that } (1 \bullet j, 2 \bullet k) \text{ is an arc of T, and } (1 \bullet j', 2 \bullet k) \text{ is an arc of T and (ii) } \psi(1 \bullet j) = x, \psi(1 \bullet j') = y \text{ and } x \neq y.\}$$

First the nodes of A_1^{rb} are assigned levels. The pairs in A_1^{rb} are considered in (lexicographic) order, according to the rank of the lower ranked member of the pair.

We may write

We may write

$$A_1^r = \{1\bullet u_1, 1\bullet u_2, \ldots, 1\bullet u_{q_1^1}\}$$

and

$$A_1^b = \{1\bullet v_1, 1\bullet v_2, \ldots, 1\bullet v_{q_1^2}\}$$

where

$$u_1 < u_2 < \ldots < u_{q_1^1}$$

and

$$v_1 < v_2 < \ldots < v_{q_1^2}$$

[Note that if b is the majority color in (T,ψ) then $q_1^2 \geq q_1^1$. However, this observation itself is not used in the process defining the schedule functions.]

We assign nodes to levels in pairs in order. Thus,

$$\lambda_1(1\bullet u_j) = \lambda_1(1\bullet v_j) = q_1 + j$$

for

$$\begin{array}{ll} 0 \leq j \leq \min\{q_1^1, q_1^2\}, & \text{if } \min\{q_1^1, q_1^2\} > 0 \\ j = 0 & \text{if } \min\{q_1^1, q_1^2\} = 0, \end{array}$$

Next, as in stage 1, define

$$A_2^r = \{i \bullet j \in A_2 \mid \psi(i \bullet j) = r \text{ and if } i' \bullet j' \in A_2 (i \bullet j, i' \bullet j') \text{ are a}$$
$$\text{p-pair, then } \psi(i' \bullet j') \neq b\}.$$

I.e., A_2^r consists of all nodes in A_2 that are colored r, but are not part of an (r,b)-p-pair. Similarly, define A_2^b . We may write

$$A_2^r = \{i_1 \bullet x_1, i_2 \bullet x_2 ... x_{q_2^1} \bullet x_{q_2^2}\}$$

and

$$A_2^b = \{k_1 \bullet y_1, k_2 \bullet y_2 ... k_{q_2^2} \bullet y_{q_2^2}\}.$$

Note that the nodes in A_2^r and A_2^b can be in tiers 1 or 2, but that only one of these sets can contain nodes from tier 1.

Then, define

$$\lambda_2(i_j \bullet x_j) = \lambda_2(k_j \bullet y_j) = q_1^* + q_2 + j$$

for

$$1 \leq j \leq \min(q_2^1, q_2^2), \text{ if } \min\{q_2^1, q_2^2\} > 0$$
$$\text{and} \quad j = 0 \qquad \text{if } \min\{q_2^1, q_2^2\} = 0.$$

At stage σ, we form A_σ^{rb} , A_σ^r and A_σ^b , from A_σ , where and $\quad j = 0, \qquad\qquad$ if $\min\{q^1_\sigma, q^2_\sigma\} = 0.$

Let

$$q^*_\sigma = q^*_{\sigma-1} + q_\sigma + \min\{g^1_\sigma, q^2_\sigma\}.$$

Finally, $\lambda_\sigma(i \bullet j) = 0$ if $i \bullet j \in A_\sigma$, and is not one of the nodes assigned as above.

There exists a first stage, call it σ^*, at which $A_\sigma = A^b_\sigma$, because b is the majority color and the process assigns equal numbers of r and b nodes at each step. Then the nodes of A_{σ^*} are assigned in order using the rule

$$\lambda_{\sigma^*}(x_j) = q^*_{\sigma^*-1} + j$$

if x_j is the node of $j^{\underline{th}}$ rank in A_{σ^*}, where

$$1 \le j \le q^2_{\sigma^*},$$

where

$$q^2_{\sigma^*} = \mathrm{card}(A^b_{\sigma^*}).$$

If ψ is a minimal coloring then $A^b_\sigma = A_\sigma$ for all $\sigma \ge \sigma^*$.

The scheduling process terminates when $A_\sigma = \emptyset$. Finally, λ: $N \to \{1,2,...\}$ is defined for each x by $\lambda(x) = \lambda_\sigma(x)$ for the (unique) value σ such that $\lambda_\sigma(x) \neq 0$.

node, k_j is the j^{th} integer such that node k_j is an r-node. The proof is by finite induction. We show first that the first r-node, k_1, is matched. Let the first r-node be node i•j. If $i \neq 1$, so that the first r-node is not on tier 1 of T, then $\bar{N}^r = \emptyset$, i.e. there are no r-nodes. Hence, in this case Property M holds vacuously. To see that $\bar{N}^r = \emptyset$ in the case $i \neq 1$, notice that if $i \neq 1$, every node on tier 1 is a b-node. In that case a minimal coloring ψ requires all nodes in N to be b-nodes. Therefore, $\bar{N}^r = \emptyset$.

So we may suppose the first r-node is on tier 1. Let it be the node 1•j for some $1 \leq j \leq n_1$.

If 1•j has no mating node (i.e., if $n_1 = 1$) then T consists of the single node 1•j = 1•1 so that $\bar{n}^r = 1$ and $\bar{n}^b = 0$, contradicting $\bar{n}^r < \bar{n}^b$.

So we may suppose that node 1•j has a mating node; for definiteness let it be 1•(j+1). If the mating node 1•(j+1) were a b-node, the p-pair 1•j and 1•(j+1) would be matched according to the matching algorithm, because both nodes are eligible initially, i.e. 1•j $\in A_1$ and 1•(j+1) $\in A_1$. In this case we have shown that i•j is matched.

Therefore, we may suppose that the p-pair (1•j, 1•(j+1)) form an (r,r)-pair, and, because ψ is a minimal coloring, their descendant, 2•l, a node in tier 2 of T, is an r-node. Suppose that 1•j is unmatched. Then all nodes on tier 2 form either (r,b)-pairs or (r,r)-pairs. To see this note that if there are any other unmatched pairs on tier 1, they are initially eligible (in A_1), and remain eligible in stage 2 of the matching process and hence are also elements of A_2. If any of them is a (b,b)-pair then the first such pair would be matched with the (r,r)-pair (1•j, 1•(j+1)), contradicting the hypothesis that 1•j is not matched. It follows that all unmatched nodes on tier 1 are r-nodes. Hence, since ψ is a minimal coloring the descendants of unmatched nodes on tier 1 are also r-nodes on tier 2.

Next observe that there can be no (b,b)-pairs among descendants on tier 2 of the matched nodes on tier 1. For if there were any such pairs, they would be eligible for matching at stage 2 (i.e. elements of A_2) and, according to the scheduling algorithm, the

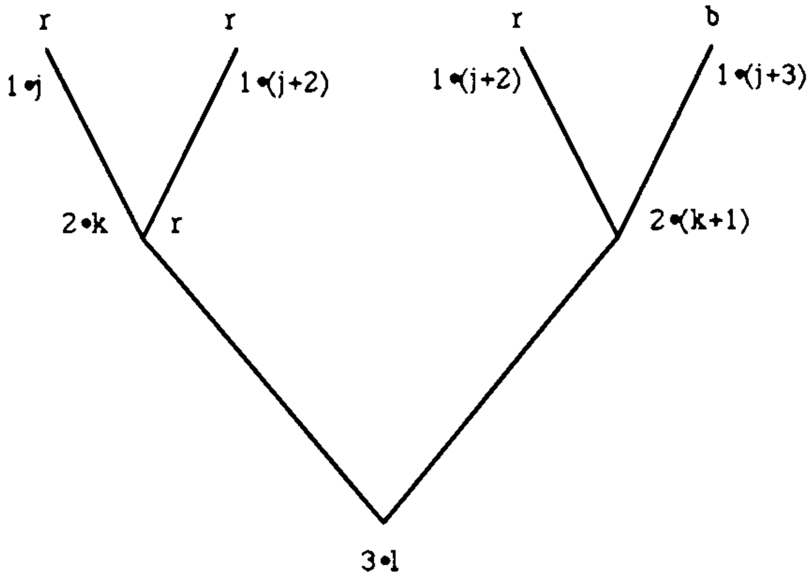


FIGURE AII-3

Node 2•(k+1) is eligible at stage 2, i.e. 2•k+1 ∈ A_2. We have already shown that all nodes on tier 2 form (r,b) or (r,r)-pairs. Hence node i•j, which being unmatched remains eligible at stage 2 (i.e., i•j ∈ A_2), cannot be paired with any node before 2•(k+1) in the ordering of nodes. Moreover node 2•k is not eligible at stage 2, because it is the descendant of a node that is not already matched by stage 2, i.e. 2•k ∉ A_2. Hence 2•k and 2•k+1 are not an eligible (r,b)-pair at stage 2. Therefore the matching algorithm would require that nodes i•j and 2•k+1 be matched, contradicting the hypothesis that 1•j is unmatched. Therefore, it cannot be the case that the parents of node 2•(k+1) are an (r,b)-pair.

So suppose that (nodes 1•(j+2) and 1•(j+3) form a (b,b)-pair. Since 1•j is the first r-node, 1•j+2 and 1•j+3 cannot be matched with any (r,r)-pair before 1•j in the ordering of nodes. But all nodes on tier 1 are eligible at Stage 1.

of T. The eligible nodes at the stage including step s can contain no r-node that is not part of an (r,b)-pair. Otherwise it would be paired with p•q. Furthermore, the eligible nodes can contain no (r,b)-pair, otherwise node p•q would not be schdeuled at step s, but would be carried forward to the new set of eligible nodes at the next stage. Therefore at the stage including step s at which node p•q is assigned, the eligible set can include only b-nodes, and moreover node p•q must be the first of those, otherwise, since only the first such node would be scheduled, p•q would not be scheduled at step s. The eligible set at step s includes all the nodes p•q' on tier p with q' > q, because all nodes on tier p-1 have been assigned before stage s. Hence, these nodes (excluding for the moment the mating node to p•q) must all be b-nodes. Consequently their descendants on tier p + 1 must be b-nodes. Furthermore, the nodes p•q' on tier p with q' < q have been scheduled before stage s, hence their descendants on tier p + 1 are eligible at stage s and hence must all be b-nodes.

Consider next the pair p•q and its mate, either p•(q-1) or p•(q+1). If the mate is a b-node, then with p•q, it forms a (b,b)-pair whose common descendant is a b-node. It would then follow from the minimal coloring property of ψ that all nodes on tier p+1 and above are b-nodes. If the mate is p•(q+1), then, it is the eligible set at stage s and has been shown to be a b-node. The remaining possibility is that the mating node is p•(q-1) and is an r-node, (already matched to a b-node other then p•q). Then one of the following two possibilities obtains.

Therefore the hypothesis that the k^{th} r-node is unmatched is false. This establishes the Lemma.

The proof of Lemma A also establishes the following: The scheduling algorithm assigns matched r and b nodes to levels until all minority nodes (r-nodes) have been assigned; then it assigns the remaining b-nodes, one to a level, until all nodes have been assigned.

<u>Lemma B</u>: Let λ be the schedule defined by applying the scheduling algorithm to the given colored tree (T,ψ) where ψ is a minimal coloring. Then, there exists an integer l^* such that if $1 \leq l^*$, then $\lambda^{-1}(1)$ contains 2 nodes, and if $l > 1^{*}$, $\lambda^{-1}(1)$ contains at most one node.

Lemma B tells us that the schedule λ, defined by the scheduling algorithm has the matching property, M. It is evident that such a schedule has no gaps, since the range of λ is an interval in the integers. Therefore, the schedule λ as defined by the scheduling algorithm attains the lower bound $|\tilde{N}^{b}| + 1$. Therefore it is a minimal schedule, i.e., $\tau_{\lambda} \leq \tau_{\lambda'}$, for all schedules λ' of (T,ψ). This assures us that property M and the "no gaps" property are necessary and, by Theorem 1, sufficient for an assignment λ of (T,ψ) to be minimal. When the coloring function ψ is the unique minimal coloring of T, then the pair (ψ,λ) where λ is a minimal schedule and (ψ,λ) satisfy the parallel constraint, are together efficient in the sense that the pair $(\nu_{\psi}(T) , \tau_{\lambda}(T) \leq (\nu_{\psi'}(T) , \tau_{\lambda'}(T))$ for all colorings ψ' of T and assignments λ' of (T,ψ'), satisfying the parallel constraint. (The inequality is, of course, the usual vectorial inequality.)

or

$$2^0 + 2^1 + 2^2 + ... + 2^{t-1} - 2|\breve{N}^r| = \breve{U}^b + 1.$$

Therefore,

$$2^1 + 2^2 + ... + 2^{t-1} - 2|\breve{N}^r| = \breve{U}^b.$$

Therefore $\breve{U}^b = 2p$, where $p = 1 + 2 + 2^2 + ... + 2^{t-2} - |\breve{N}^r|$, which is an integer; therefore $\breve{U}^b$ is even.

It follows that there is a point in (ν,τ)-space obtained by recoloring half of the "excess" b-nodes to r-nodes, and reassigning them to levels matched with b-nodes. The point so obtained is efficient.

To see this, note that if the (even) number of excess b-nodes is 2p, for some integer p, then this process reduces the length of the network by p levels. This is the minimum length that can be attained by recoloring, because at that point there are equally many r-nodes as b-nodes. Hence recoloring any node increases the number of minority color nodes by one, and decreases the number of majority color nodes by one, and hence increases the lower bound on the length of the network by one.

The efficient point corresponding to recoloring of p b-nodes is obtained by proper choice of the set of b-nodes to be recolored from b to r. An example:

This is the only efficient recoloring of 4 b-nodes to r-nodes, when all 1st tier b-nodes have (b,b) inputs.

The following table shows the possibilities for 4-tier regular trees. Such a tree has 15 nodes = $2^0 + 2^1 + 2^2 + 2^3$

$\|N^b\|$	$\|\bar{N}^r\|$	$\|\bar{E}^b\|$	$\|\bar{E}^b\| = \| \|$
			2
14	0	14	7
13	1	12	6
12	2	10	5
* 11	3	8	4
10	4	6	3
9	5	4	2
8	6	2	1
7	7	0	0

FIGURE AII-7

For the Example, the changes can involve 1,2,3 or 4 b-nodes. Consequently, the whole efficient frontier can be generated as follows.

To change 1 b-node efficiently, if there is a 1st tier b-node with an (r,b) input, change it to an r-node. This results in an increase of 1 crosslink (the output connection of such a node must be a b-node if the original coloring was a minimal one). If there is no such b-node, then change a b-node with (b,b) inputs and an r-node as the output connection. This results in an increase of 1 crosslink.

If there is no such node, change a first tier b-node with (b,b) inputs and output connection to a b-node. This results in an increase of 3 crosslinks.

This results in an increase of 1 crosslink. Changing 3 b-nodes that do not form a tree, but have (r,b) inputs results in an increase of 1 crosslink per node or 3 crosslinks.

If it is possible by changing 3 b-nodes to r-nodes to create a subtree whose output connection is an r-node, then do that. E.g.

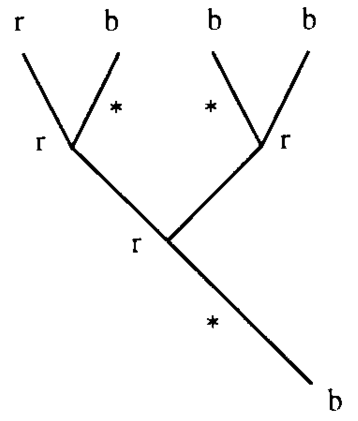


FIGURE AII-10

This results in an increase of 3 crosslinks.

If it is possible by changing 3 b-nodes to r-nodes to create a subtree whose output connection is an r-node, then do that. E.g.

$\Delta\tau$	Δv
-1	+3
-2	+4
-3	+2
-4	+5

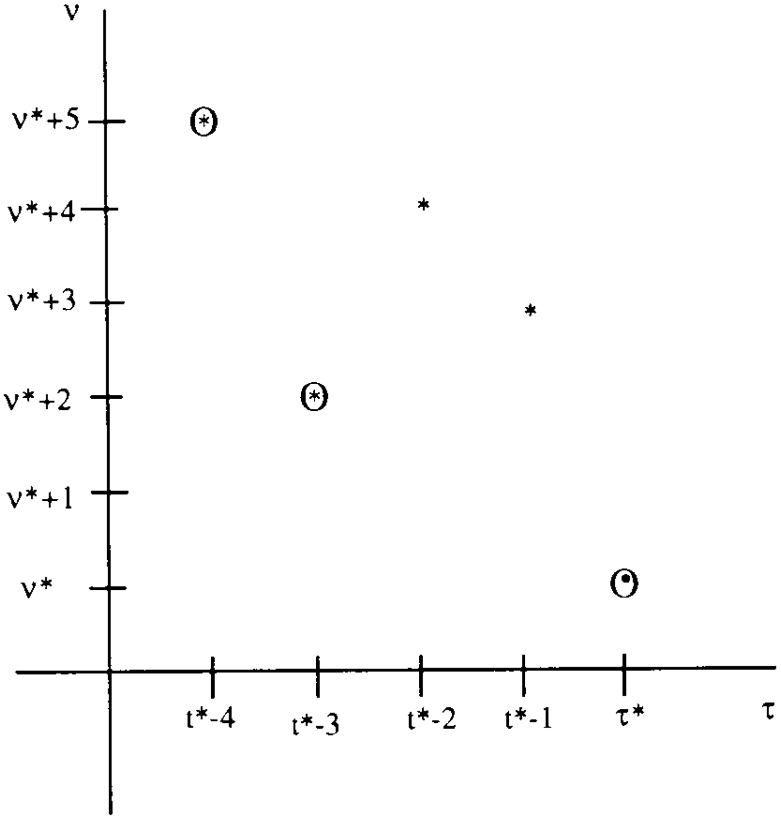


FIGURE AII-12

The point (v^*,τ^*) is the point obtained by applying the scheduling algorithm to the minimally colored tree (T,ψ) where ψ is an efficient minimal coloring.

All points shown in Figure 53 have the property that each coordinate is minimal given the other. However, only the circled points in the graph are efficient. At the 2 uncircled points, v is minimal given τ, and τ is minimal given v, but they are both dominated by the point $(\tau^* - 3, v^* + 2)$. This shows that trading off time for crosslinks one unit at a time does not generate the lower boundary of a convex set, (viewed as consisting of the line segments between integer pairs corresponding to τ^*, τ^*-1, τ^*-2, etc.)