

AD-A147 174

ON THE COMPLEXITY OF DISTRIBUTED DECISION PROBLEMS(U)  
MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR  
INFORMATION AND DECISION SYSTEMS J TSITSIKLIS ET AL.

1/1

UNCLASSIFIED

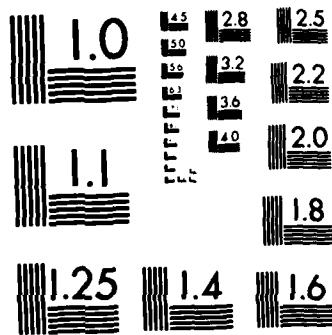
AUG 84 LIDS-P-1398 N00014-77-C-0532

F/G 12/1

NL



END  
FORM  
NO. 1  
MAY 68

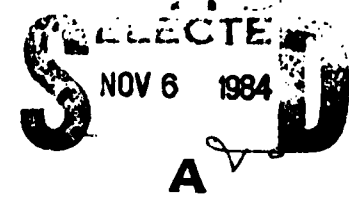


MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963 A

## ON THE COMPLEXITY OF DISTRIBUTED DECISION PROBLEMS

by

John Tsitsiklis and Michael Athans  
 Laboratory for Information and Decision Systems  
 Massachusetts Institute of Technology  
 Cambridge, Massachusetts 02139



## ABSTRACT

We study the computational complexity of finite versions of the simplest and fundamental problems of distributed decision making and we show that, apart from a few exceptions, such problems are hard (NP-complete, or worse). Some of the problems studied are the well-known team decision problem, the distributed hypothesis testing problem, as well as the problem of designing a communications protocol that guarantees the attainment of a prespecified goal with as little communications as possible. These results indicate the inherent difficulty of distributed decision making, even for very simple problems, with trivial centralized counterparts and suggest that optimality may be an elusive goal of distributed systems.

set of processors obtain (possibly conflicting) observations on the state of the environment. Each processor has to make a decision, based on his own observation. However, for each state of the environment, only certain decisions accomplish the desired goal. The question "are there any communications necessary?" may be then reformulated as "can the goal be accomplished, with certainty, without any communications?" We show that this problem is, in general, a hard one.

We then impose some more structure on the problem, by assuming that the observations of different processors are related in a particular way. The main issue that we address is "how much structure is required so that the problem is an easy one?" and we try to determine the boundary between easy and hard problems.

In Section 3 we formulate a few problems which are related to the basic problem of Section 2 and discuss their complexity.

In Section 4 we study a particular (more structured) decentralized problem - the problem of decentralized hypothesis testing - on which there has been some interest recently, and characterize its difficulty.

Suppose that it has been found that communications are necessary. The next question of interest is "what is the least amount of communications needed?" This problem (Section 5) is essentially the problem of designing an optimal communications protocol; it is again a hard one and we discuss some related issues.

In Section 6 we present our conclusions and discuss the conceptual significance of our results. These conclusions may be summarized by saying that:

- Even the simplest (exact) problems of decentralized decision making are hard.
- Allowing some redundancy in communications, may greatly facilitate the (off-line) problem of designing a decentralized system.
- Practical communications protocols should not be expected to be optimal, as far as minimization of the amount of communications is concerned.

Some of the results of this paper appear in [Papadimitriou and Tsitsiklis, 1983] and (almost) all proofs may be found in [Tsitsiklis, 1983].

## 2. A Problem of Silent Coordination

In this section we formulate and study the problem whether a set of processors with different information may accomplish a given goal -with certainty- without any communications.

Let  $(1, \dots, N)$  be a set of processors. Each processor, say processor  $i$ , obtains an observation  $y_i$  which comes from a finite set  $Y_i$  of possible observations. Then, processor  $i$  makes a decision  $u_i$  which belongs to a finite set  $U_i$  of possible decisions, according to a rule.

## 1. Introduction and Motivation

In this paper we formulate and study certain simple decentralized problems. Our goal is to formulate problems which reflect the inherent difficulties of decentralization; that is, any difficulty in this class of problems is distinct from the difficulty of corresponding centralized problems. This is accomplished by formulating decentralized problems whose centralized counterparts are either trivial or vacuous.

One of our goals is to determine a boundary between "easy" and "hard" decentralized problems. Our results will indicate that the set of "easy" problems is relatively small.

All problems to be studied are imbedded in a discrete framework; the criteria we use for deciding whether a problem is difficult or not come from complexity theory [Garey and Johnson, 1979; Papadimitriou and Steiglitz, 1982]: following the tradition of complexity theory, problems that may be solved by a polynomial algorithm are considered easy; NP-complete, or worse, problems are considered hard. However, an NP-completeness result does not close a subject, but is rather a result which can guide research: further research should focus on special cases of the problem or on approximate versions of the original problem.

The main issue of interest in decentralized systems may be loosely phrased as "who should communicate to whom, what, how often etc." From a purely logical point of view, the first question that has to be raised is "are there any communication necessary?" Any further questions deserve to be studied only if we come to the conclusion that communications are indeed necessary.

The subject of Section 2 is to characterize the inherent difficulty of the problem of deciding whether any communications are necessary, for a given situation. We adopt the following approach: a decentralized system exists in order to accomplish a certain goal which is externally specified and well-known. A

\* Research supported by ONR under contract ONR/N00014-77-C-0332 (NR-041-519).

\* One way of viewing NP-complete problems, is to say that they are effectively equivalent to the Traveling Salesman problem, which is well-known to be algorithmically hard.

This document has been approved  
 for public release and sale; its  
 distribution is unlimited.

84

10 30 002

FILE COPY

VAD-A147 174

$$u_i = \partial_i(y_i), \quad (2.1)$$

where  $\partial_i$  is some function from  $Y_i$  into  $U_i$ . The M-tuple  $(y_1, \dots, y_M)$  is the total information available; so it may be viewed as the "state of the environment." For each state of the environment, we assume that only certain M-tuples  $(u_1, \dots, u_M)$  of decision accomplish a given, externally specified, goal. More precisely, for each  $(y_1, \dots, y_M) \in Y_1 \times \dots \times Y_M$  we are given a set  $S(y_1, \dots, y_M) \subset U_1 \times \dots \times U_M$  of satisficing decisions.

(So,  $S$  may be viewed as a function from

$$Y_1 \times Y_2 \times \dots \times Y_M \text{ into } 2^{U_1 \times \dots \times U_M}.$$

The problem to be studied, which we call "distributed satisficing problem" (after the term introduced by H. Simon [1980]) may be described formally as follows:

Distributed Satisficing (DS): Given finite sets  $Y_1, \dots,$

$Y_M, U_1, \dots, U_M$  and a function  $S: Y_1 \times \dots \times Y_M \rightarrow$

$2^{U_1 \times \dots \times U_M}$ ,

are there functions  $\partial_i: Y_i \rightarrow U_i, i=1,$

$2, \dots, M$ , such that

$$(\partial_1(y_1), \dots, \partial_M(y_M)) \in S(y_1, \dots, y_M), \quad \forall (y_1, \dots, y_M) \in Y_1 \times \dots \times Y_M \quad (2.2)$$

Remarks:

1. We are assuming that the function  $S$  is "easily computable;" for example, it may be given in the form of a table.

2. The centralized counterpart of DS would be to allow the decision  $u_i$  of each agent depend on the entire set  $(y_1, \dots, y_M)$  of observations; so,  $\partial_i$  would be a function from  $Y_1 \times \dots \times Y_M$  into  $U_i$ . (This corresponds to a situation in which all processors share the same information.)

Clearly, then, there exist satisfactory (satisficing) functions  $\partial_i: Y_1 \times \dots \times Y_M \rightarrow U_i$ , if and only if  $S(y_1, \dots, y_M) \neq \emptyset, \forall (y_1, \dots, y_M) \in Y_1 \times \dots \times Y_M$ .

Since  $S$  is an "easily computable" set as a function of its arguments, we can see that the centralized counterpart of DS is a trivial problem. So, any difficulty inherent in DS is only caused by the fact that information is decentralized.

3. A "solution" for the problem DS cannot be a closed-form formula which gives an answer 0(no) or 1(yes). Rather, it has to be an algorithm, a sequence of instructions, which starts with the data of the problem  $(Y_1, \dots, Y_M, U_1, \dots, U_M, S)$  and eventually provides the correct answer. Accordingly, the difficulty of the problem DS may be characterized by determining the place held by DS in the complexity hierarchy. For definitions related to computational complexity and the methods typically used, the reader is referred to [Garey and Johnson, 1979; Papadimitriou and Steiglitz, 1982].

4. If, for some  $i$ , the set  $U_i$  is a singleton, processor  $i$  has no choice, regarding his decision and, consequently, the problem is equivalent to a problem in which processor  $i$  is absent. Hence, without loss of generality, we only need to study instances of DS in which  $|U_i| \geq 2, \forall i$ .

5. We believe that the problem DS captures the essence of coordinated decision making with decentralized information and without communications (silent coordination).

Some initial results on DS are given by the

following:

Theorem 2.1:

a) The problem DS with two processors ( $M=2$ ) and restricted to instances for which the cardinality of the decision sets is 2 ( $|U_i|=2, i=1,2$ ) may be solved in polynomial time.

b) The problem DS with two processors ( $M=2$ ) is NP-complete, even if we restrict to instances for which  $|U_1|=2, |U_2|=3$ .

c) The problem DS with three (or more) processors ( $M \geq 3$ ) is NP-complete, even if we restrict to instances for which  $|U_i|=2, \forall i$ .

Theorem 2.1 states that the problem DS is, in general, a hard combinatorial problem, except for the special case in which there are only two processors and each one has to make a binary decision. It should be noted that the difficulty is not caused by an attempt to optimize with respect to a cost function, because no cost function has been introduced. In game theoretic language, we are faced with a "game of kind," rather than a "game of degree."

We will now consider some special cases (which reflect the structure of typical practical problems) and examine their computational complexity, trying to determine the dividing line between easy and hard problems. From now on we restrict our attention to the case in which there are only two processors. Clearly, if a problem with two processors is hard, the corresponding problem with three or more processors cannot be easier.

We have formulated above the problem DS so that all pairs  $(y_1, y_2) \in Y_1 \times Y_2$  are likely to occur. So, the information of different processors is completely unrelated; their coupling is caused only by the structure of the satisficing sets  $S(y_1, y_2)$ . In most practical situations, however, information is not completely unstructured: when processor 1 observes  $y_1$ , he is often able to make certain inferences about the value of the observation  $y_2$  of the other processor and exclude certain values. We now formalize these ideas:

Definition: An Information Structure  $I$  is a subset of  $Y_1 \times Y_2$ . We say that an information structure  $I$  has

degree  $(D_1, D_2)$  ( $D_1, D_2$  are positive integers) if

(i) For each  $y_1 \in Y_1$  there exist at most  $D_1$  distinct elements of  $Y_2$  such that  $(y_1, y_2) \in I$ .

(ii) For each  $y_2 \in Y_2$  there exist at most  $D_2$  distinct elements of  $Y_1$  such that  $(y_1, y_2) \in I$ .

(iii)  $D_1, D_2$  are the smallest integers satisfying (i),

(ii). An information structure  $I$  is called classical if  $D_1=D_2=1$ ; nested if  $D_1=1$  or  $D_2=1$ .

We now interpret this definition: The information structure  $I$  is the set of pairs  $(y_1, y_2)$  of observations that may occur together. If  $I$  has degree  $(D_1, D_2)$

processor 1 may use his own observation to decide which elements of  $Y_2$  may have been observed by processor 2.

In particular, he may exclude all elements except for  $D_1$  of them. The situation faced by processor 2 is symmetrical.

If  $D_1=1$  and processor 1 observes  $y_1$ , there is only one possible value for  $y_2$ . So, processor 1 knows the observation of processor 2. (The converse is true when  $D_2=1$ ). This is called a nested information structure because the information of one processor contains the information of the other.

When  $D_1=D_2=1$ , each processor knows the observation of the other; so, their information is essentially

shared.

Since pairs  $(y_1, y_2)$  not in  $I$  cannot occur, there is no meaning in requiring the processors to make compatible decisions if  $(y_1, y_2)$  were to be observed. This leads to the following version of the problem DS:

**DSI:** Given finite sets  $Y_1, Y_2, U_1, U_2$ ,  $I \subset Y_1 \times Y_2$  and a function  $S: I \rightarrow 2^{U_1 \times U_2}$ , are there functions  $\partial_i: Y_i \rightarrow U_i$ ,  $i=1,2$ , such that

$$(\partial_1(y_1), \partial_2(y_2)) \in S(y_1, y_2), \forall (y_1, y_2) \in I? \quad (2.3)$$

Note that any instance of DSI is equivalent to an instance of DS in which  $S(y_1, y_2) = U_1 \times U_2$ ,  $\forall (y_1, y_2) \in I$ .

That is, no compatibility restrictions are placed on the decisions of the two processors, for those  $(y_1, y_2)$  that cannot occur.

We now proceed to the main result of this Section:

**Theorem 3.2.2:**

a) The problem DSI restricted to instances satisfying any of the following:

- (i) One or more of  $|U_1|, |U_2|, D_1, D_2$  is equal to 1.
- (ii)  $|U_1| = |U_2| = 2$ .
- (iii)  $D_1 = D_2 = 2$ .
- (iv)  $D_1 = |U_2| = 2$ , (or  $D_1 = |U_1| = 2$ )

may be solved in polynomial time.

b) The problem DSI is NP-complete even if we restrict to instances for which

$$|U_1| = D_1 = 3, |U_2| = D_2 = 2$$

The result concerning the case  $D_1=1$  or  $D_2=1$  is not surprising. It is well-known that nested information structures may be exploited to solve otherwise difficult decentralized problems. But except for the case  $D_1=D_2=2$  (which is sort of a boundary) the absence of nestedness makes decentralized problems computationally hard. Our result gives a precise meaning to the statement that non-nested information structures are much more difficult to handle than nested ones.

Theorem 3.2.2 shows that even if  $D_1, D_2$  are held constant, the problem DSI is, in general, NP-complete. There is, however, a special case of DSI, with  $D_1, D_2$  constant, for which an efficient algorithm of the dynamic programming type is possible:

**Theorem 3.2.3:** Let  $Y_1 = \{1, \dots, m\}$ ,  $Y_2 = \{1, \dots, n\}$  and suppose that  $|i-j| \leq D$ ,  $\forall (i,j) \in I$ . Then, if  $D$  is held constant, DSI may be solved in polynomial time.

**Remark:** In fact, the conclusion of Theorem 3.2.3 remains true if we assume  $m=n$  and we replace the condition  $|i-j| \leq D$  by the weaker condition  $|i-j| \pmod n \leq D$ . The proof consists of a small modification of the preceding one.

The condition  $|i-j| \leq D$ ,  $\forall (i,j) \in I$  is fairly natural in certain applications. For example, suppose that the observations  $y_1$  and  $y_2$  are noisy measurements of an unknown variable  $x$  ( $y_i = x + w_i$ ) where the noises  $w_i$  are bounded:  $|w_i| \leq D/2$ .

The condition  $|i-j| \pmod n \leq D$  may also arise if the observations  $y_1, y_2$  are noisy measurements of some unknown angle:  $y_i = \theta + w_i$

**3. Related Problems**

In this Section we define and discuss briefly a few more combinatorial problems relevant to decentralized decision making. All of them will be seen to be harder than problem DS of the last section (i.e. they contain DS as a special case) and are, therefore NP-hard (that is, NP-complete, or worse).

The best known static decentralized problem is the team decision problem [Marschak and Badner, 1972] which admits an elegant solution under linear quadratic assumptions. Its discrete version is the following:

**TDP (Team Decision Problem):** Given finite sets  $Y_1, Y_2, U_1, U_2$ , a probability mass function  $p: Y_1 \times Y_2 \rightarrow Q$ , and a cost function  $c: Y_1 \times Y_2 \times U_1 \times U_2 \rightarrow N$ , find decision rules  $\partial_i: Y_i \rightarrow U_i$ ,  $i=1,2$  which minimize the expected cost

$$J(\partial_1, \partial_2) = \sum_{y_1 \in Y_1} \sum_{y_2 \in Y_2} c(y_1, y_2, \partial_1(y_1), \partial_2(y_2)) P(y_1, y_2)$$

Let  $S(y_1, y_2) = \{(u_1, u_2) \in U_1 \times U_2 : c(y_1, y_2, u_1, u_2) = 0\}$ . If we solve TDP, we have effectively answered the question whether there exist  $\partial_1, \partial_2$  such that  $J(\partial_1, \partial_2) = 0$ . This is equivalent to the question whether there exist satisficing decision rules (with the satisficing sets  $S(y_1, y_2)$  defined as above). Therefore, TDP is harder than DS:

**Proposition 3.1:** The discrete team decision problem is NP-hard, even if the range of the cost function  $c$  is  $\{0,1\}$ .

Instead of trying to "satisfice" for every pair of observations  $(y_1, y_2) \in Y_1 \times Y_2$ , it may be more appropriate to impose a probability mass function on  $Y_1 \times Y_2$  and try to maximize the probability of satisficing. This leads to the next problem:

**MPS (Maximize Probability of Satisficing):** Given finite sets  $Y_1, Y_2, U_1, U_2$ , a probability mass function

$p: Y_1 \times Y_2 \rightarrow Q$  and a function  $S: Y_1 \times Y_2 \rightarrow 2^{U_1 \times U_2}$ , find decision rules  $\partial_i: Y_i \rightarrow U_i$ ,  $i=1,2$ , which maximize the probability of satisficing  $J(\partial_1, \partial_2) =$

$$\Pr((\partial_1(y_1), \partial_2(y_2)) \in S(y_1, y_2)).$$

We now take a slightly different point of view. Suppose that communications are allowed, so that the processors may always make satisficing decisions by communicating (assuming that  $S(y_1, y_2) \neq \emptyset$ ,

$\forall (y_1, y_2) \in Y_1 \times Y_2$ ). Suppose, however, that communications are very expensive, so that we are interested in a scheme which guarantees satisficing with a minimum amount of communications. We will assume that if one of the processors initiates a communication, all their information will be exchange at unity cost. (For a more refined way of counting the amount of communications, see Section 3.5.)

**MPC (Minimize Probability of Communications):** Given finite sets  $Y_1, Y_2, U_1, U_2$  a probability mass function

$p: Y_1 \times Y_2 \rightarrow Q$  and a function  $S: Y_1 \times Y_2 \rightarrow 2^{U_1 \times U_2}$ , find decision rules  $j_i: Y_i \rightarrow U_i \cup \{C\}$ ,  $i=1,2$ , which minimize the probability  $\Pr(\partial_1(y_1) = C \text{ or } \partial_2(y_2) = C)$  of communicating subject to the constraint

If  $\{\partial_1(y_1) \neq C\}$  and  $\{\partial_2(y_2) \neq C\}$  then  $(\partial_1(y_1), \partial_2(y_2)) \in S(y_1, y_2)$ .

The proof of the following is trivial:

**Proposition 3.2:** The problems MPS and MPC are NP-hard.

In fact, we also have:

**Proposition 3.3:** The problems TDP (with a zero-one cost function) and MPS are NP-hard, even if  $|U_1| = |U_2| = 2$ .

We could also define dynamic versions of DS or of the team problem, in a straightforward way [Tenney, 1983]. Since dynamic problems cannot be easier than static ones, they are automatically NP-hard.

#### 4. Decentralized Hypothesis Testing

A basic problem in decentralized signal processing, which has attracted a fair amount of attention recently, is the problem of decentralized hypothesis testing [Tenney and Sandell, 1981; Ekchian, 1982; Ekchian and Tenney, 1982; Kushner and Pacut, 1982; Lauer and Sandell, 1983]. A simple version of the problem, involving only two processors and two hypotheses may be described as follows:

Two processors  $S_1$  and  $S_2$  receive observations  $y_1 \in Y_1$ ,  $y_2 \in Y_2$ , respectively, where  $Y_i$  is the set of all possible observations of processor  $i$ . (Figure 1). There are two hypotheses  $H_0$  and  $H_1$  on the state of the environment, with prior probabilities  $p_0$  and  $p_1$  respectively. For each hypothesis  $H_i$ , we are also given the joint probability distribution  $P(y_1, y_2 | H_i)$  of the observations, conditioned on the event that  $H_i$  is true.

Upon receipt of  $y_i$ , processor  $S_i$  evaluates a message  $u_i \in \{0, 1\}$  according to the rule  $u_i = \partial_i(y_i)$ , where  $\partial_i: Y_i \rightarrow \{0, 1\}$ . Then,  $u_1$  and  $u_2$  are transmitted to a

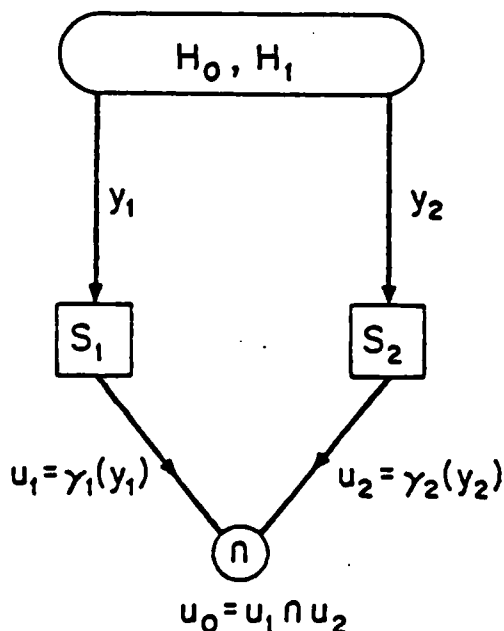


Figure 1: A Scheme for Decentralized Hypothesis Testing.

central processor (fusion center) which evaluates  $u_0 = u_1 \cap u_2$  and declares hypothesis  $H_0$  to be true if  $u_0 = 0$ ,  $H_1$  if  $u_0 = 1$ . (So, we essentially have a voting scheme). The problem is to select the functions  $\partial_1$ ,  $\partial_2$  so as to minimize the probability of accepting the wrong hypothesis. (More general performance criteria may be also considered).

Most available results assume that

$$P(y_1, y_2 | H_i) = P(y_1 | H_i) P(y_2 | H_i), \quad i=1,2, \quad (4.1)$$

which states that the observations of the two processors are independent, when conditioned on either hypothesis. In particular, it has been shown [Tenney and Sandell, 1981] that the optimal decision rules  $\partial_i$  are given in terms of thresholds for the likelihood ratios  $\frac{p_0 P(H_0 | y_i)}{p_1 P(H_1 | y_i)}$ . The optimal thresholds for

the two sensors are coupled through a system of equations which gives necessary conditions of optimality. (These equations are precisely the person-by-person optimality conditions). Few analytical results are available when the conditional independence assumption is removed [Lauer and Sandell, 1983]. The approach of this section is aimed at explaining this status of affairs, by focusing on discrete (and finite) versions of the problem.

We first have:

**Theorem 4.1:** If  $Y_1, Y_2$  are finite sets and (4.1) holds, then optimal choices for  $\partial_1, \partial_2$  may be found in polynomial time.

So, under the conditional independence assumption, decentralized hypothesis testing is a computationally easy problem. Unfortunately, this is not the case when the independence assumption is relaxed. Our main result (Theorem 4.2) states that (with  $Y_1, Y_2$  finite

sets), decentralized hypothesis testing is a hard combinatorial problem (NP-hard). This is true even if we restrict to the special case where perfect detection (zero probability of error) is possible for the corresponding centralized hypothesis testing problem.

Although this is in some sense a negative result, it is useful because it indicates the direction in which future research on this subject should proceed:

Instead of trying to find efficient exact algorithms, research should focus on approximate algorithms, or exact algorithms for problems with more structure than that assumed here. Moreover, our result implies that any necessary conditions for optimality to be developed are likely to be deficient in one of two respects:

- Either there will be a very large number of decision rules satisfying these conditions.
- Or, it will be hard to find decision rules satisfying these conditions.

In particular, optimal decision rules are not given in terms of thresholds on likelihood ratios.

Of course, there remains the question whether efficient approximate algorithms exist for the general decentralized hypothesis testing problem, or whether we must again restrict to special cases of the problem.

We now present formally the problem to be analyzed.

**DHT:** (Decentralized Hypothesis Testing, Restricted to Instances for which Perfect Centralized Detection is Possible).

We are given finite sets  $Y_1, Y_2$ ; a rational number  $k$ ; a rational probability mass function  $p: Y_1 \times Y_2 \rightarrow Q \cap [0, 1]$ ; a partition

\* Such an assumption is reasonable in problems of detection of a known signal in independent noise, but is typically violated in problems of detection of an unknown signal.

$\{A_0, A_1\}$  of  $Y_1 \times Y_2$ . \* Do there exist  $\partial_1: Y_1 \rightarrow \{0,1\}$ ,  $\partial_2: Y_2 \rightarrow \{0,1\}$  such that  $J(\partial_1, \partial_2) \leq k$ , where

$$J(\partial_1, \partial_2) = \sum_{(y_1, y_2) \in A_0} P(y_1, y_2) \partial_1(y_1) \partial_2(y_2) + \sum_{(y_1, y_2) \in A_1} P(y_1, y_2) [1 - \partial_1(y_1) \partial_2(y_2)]? \quad (4.2)$$

**Remarks:** 1. If we let  $k=0$ , then DHT is a special case of problem DS (Section 2), with  $|U_1|=|U_2|=2$ , and is polynomially solvable, according to Theorem 3.2.1. In general DHT is a special case of MPS and TDP (Section 3.3) with  $|U_1|=|U_2|=2$ . Consequently, Theorem 4.2

below proves Proposition 3.3.

2) Clearly, the optimization problem (Minimize  $J(\partial_1, \partial_2)$ , with respect to  $\partial_1, \partial_2$ ) cannot be easier than DHT.

Since DHT will be shown to be NP-complete, it follows that the above optimization problem is NP-hard.

3) In DHT, as defined above, we are only considering instances for which perfect centralized detection is possible: Think of  $H_0$  as being the hypothesis that

$(y_1, y_2) \in A_0$ , and  $H_1$  as being the hypothesis that  $(y_1, y_2) \in A_1$ . Certainly, if a processor knows both  $y_1, y_2$ , the true hypothesis may be found with certainty.

For the decentralized problem, the cost function  $J(\partial_1, \partial_2)$  is easily seen to be the probability of error.

4) The result to be obtained below remains valid if the fusion center uses different rules for combining the messages it receives (e.g.  $u_0 = (u_1 \vee u_2)$ ), or if we leave the combining rule unspecified and try to find an optimal combining rule.

**Theorem 4.2:** DHT is NP-complete.

## 5. On Designing Communications Protocols

Suppose that we are given an instance of the distributed satisficing problem (DS) and that it was concluded that unless the processors communicate, satisficing cannot be guaranteed for all possible observations. Assuming that communications are allowed (but are costly), we have to consider the problem of designing a communications protocol: what should each processor communicate to the other, and at what order? Moreover, since communications are costly, we are interested in a protocol which minimizes the total number of binary messages (bits) that have to be communicated. (The word "bits" above does not have the information theoretic meaning.)

Before proceeding, we must make more precise the notion of a communication protocol and of the number of bits than guarantee satisficing.

Given an instance  $D = (Y_1, Y_2, U_1, U_2, I, S)$  of the problem DS1 we will say that:

There is a protocol which guarantees satisficing with 0 bits of communications, if  $D$  is a YES instance of the problem DS1. (That is, if there exist satisficing decision rules, involving no communications.)

We then proceed inductively:

There is a protocol which guarantees satisficing with  $K$  bits of communications (KE  $N$ ), if for some  $i \in \{1, 2\}$  (say,  $i=1$ ) there is a function  $m: Y_1 \rightarrow \{0,1\}$ , such that for each of the instances  $D^i = (Y_1 \cap m^{-1}(0), Y_2, U_1, U_2, I \cap ((Y_1 \cap m^{-1}(0)) \times Y_2), S)$  and

$D^i = (Y_1 \cap m^{-1}(1), Y_2, U_1, U_2, I \cap ((Y_1 \cap m^{-1}(1)) \times Y_2), S)$  there is a protocol which guarantees satisficing with not more than  $K-1$  bits of communications. (Here  $m^{-1}(i) = \{y_1 \in Y_1 : m(y_1) = i\}$ .)

The envisaged sequence of events behind this definition is the following: Each processor observes his measurement  $y_i \in Y_i$ ,  $i=1,2$ . Then, one of the processors, say processor 1, transmits a message  $m(y_1)$ , with a single bit to the other processor. From that point on, it has become common knowledge that  $y_1 \in Y_1 \cap m^{-1}(y_1)$ ;

therefore, the remaining elements of  $Y_1$  may be ignored.

We can now state formally the problem of interest:

**MBS (Minimum bits to satisfice):** Given an instance  $D$  of DS1 and KE  $N$ , is there a protocol which guarantees satisficing with not more than  $K$  bits of communications?

By definition, MBS with  $K=0$  is identical to the problem DS1. Moreover, MBS with  $K$  arbitrary cannot be easier than MBS with  $K=0$  (which is a special case). Therefore, MBS is, in general NP-hard. Differently said, problems involving communications are at least as hard as problems involving no communications.

We have seen in Section 2 that when  $|U_1|=|U_2|=2$ , DS1 may be solved in polynomial time. Therefore, MBS with  $K=0$ ,  $|U_1|=2$ ,  $|U_2|=2$  is polynomially solvable.

However, for arbitrary  $K$ , this is no longer true:

**Theorem 5.1:** MBS is NP-complete, even if  $|U_1|=|U_2| = \{0,1\}$  and even if we restrict to instances for which, for any  $(y_1, y_2) \in I$ , either  $S(y_1, y_2) = \{(0,0)\}$  or  $S(y_1, y_2) = \{(1,1)\}$ .

The above theorem proves a conjecture of A. Yao [Yao, 1979]. The proof was mainly constructed by C. Papadimitriou and may be found in [Papadimitriou and Tsitsiklis, 1982].

We should point out that the special case referred to in Theorem 5.1 concerns the problem of distributed function evaluation: we are given a Boolean function  $f: Y_1 \times Y_2 \rightarrow \{0,1\}$  and we require that both agents (processors) eventually determine the value of the function (given the observation -input  $(y_1, y_2)$ ), by exchanging a minimum number of bits. In our formalism,  $S(y_1, y_2) = \{(0,0)\}$  if  $f(y_1, y_2) = 0$  and  $S(y_1, y_2) = \{(1,1)\}$  if  $f(y_1, y_2) = 1$ .

In Section 2 we had investigated the complexity of DS1 by restricting to instances for which the set  $I$  had constant degree  $(D_1, D_2)$ . This may be done, in principle, for MBS, as well, but no results are available, except for the simple case in which  $D_1 = D_2 = 2$ .

In fact, when  $D_1 = D_2 = 2$  each processor may transmit his information to the other agent by communicating a single binary message and, for this reason, we have:

**Proposition 5.1:** MBS restricted to instances for which  $D_1 = D_2 = 2$  may be solved in polynomial time. Moreover, an optimal protocol requires transmission of at most two binary messages, one from each processor.

When  $(D_1, D_2)$  is larger than  $(2,2)$ , there is not much we can say about optimal protocols. However, it is easy to verify that there exist fairly simple non-optimal protocols (which may be calculated in polynomial time) which involve relatively small amounts of communication. This is because:

**Proposition 5.2:** Suppose that  $I$  has degree  $(D_1, D_2)$  and that  $S(y_1, y_2) \neq \emptyset$ ,  $\forall (y_1, y_2) \in I$ . Then information may be centralized (and therefore satisficing is guaranteed) by means of a protocol requiring communication of at

\* That is  $A_0 \cup A_1 = Y_1 \times Y_2$  and  $A_0 \cap A_1 = \emptyset$ .

most  $\lfloor \log_2(D_1 D_2) \rfloor$  binary messages by each processor. Moreover, such a protocol may be constructed in time  $O((|Y_1| \cdot |Y_2|)(|Y_1| + |Y_2|))$ . (Here  $\lfloor x \rfloor$ ,  $x \in \mathbb{R}$ , stands for the smallest integer larger than  $x$ .)

**Remark 1:** It might be tempting to guess that processor 1 (respectively 2) needs to communicate only  $\lfloor \log_2 D_2 \rfloor$  (respectively  $\lfloor \log_2 D_1 \rfloor$ ) bits, but this is not true, as can be seen from fairly simple examples.

## 6. Conclusions

We summarize here the main conclusions of this paper.

Even if a set of processors have complete knowledge of the structure of a distributed decision making problem and the desired goal; even if the corresponding centralized problem is trivial; even if all relevant sets are finite, a satisficing decision rule that involves no on-line communications may be very hard to find, the corresponding problem being, in general, NP-complete. There are many objections to the idea that NP-completeness is an unequivocal measure of the difficulty of a problem, because it is based on a worst case analysis, whereas the average performance of an algorithm might be a more adequate measure; moreover NP-hard optimization problems may have very simple approximate algorithms. However, NP-complete problems are often characterized by the property that any known algorithm is very close to systematic exhaustive search; they do not possess any structure to be exploited.

Concerning the problem DS, and its variations, we may reach the following specific conclusions: No simple algorithm could solve DS. Given that communications would be certainly required for those instances of DS that possess no satisficing decision rules, it would not be a great loss if we allowed the processors to communicate even for some instances of DS for which this would not be necessary. Even if these extra communications -being redundant- do not lead to better decisions, they may greatly facilitate the decision process and -from a practical point of view - remove some load from the computing machines employed.

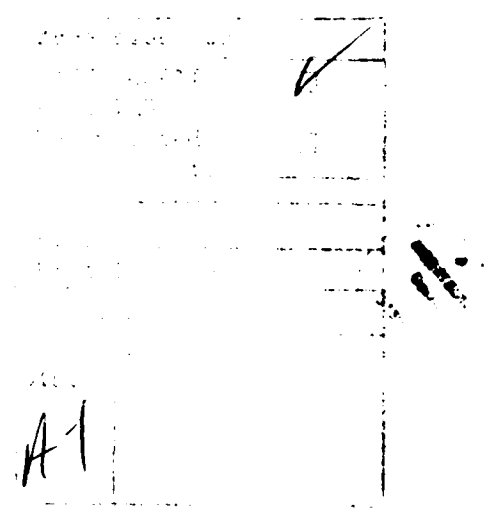
Concerning the problem of distributed hypothesis testing, we have shown that it becomes hard, once a simplifying assumption of conditional independence is removed. This explains why no substantial progress on this problem had followed the work of Tenney and Sandell [1982].

From a more general perspective, we are in a position to say that the basic (and the simplest) problems of decentralized decision making are hard, in a precise mathematical sense. Moreover, their difficulty does not only arise when one is interested in optimality. Difficulties persist even if optimality is replaced by satisficing. As a consequence, further research should focus on special cases and easily solvable problem as well as on approximate versions of the original problems.

In cases where communications are necessary (but costly) there arises naturally the problem of designing a protocol of communications. Unfortunately, if this problem is approached with the intention to minimize the amount of communications that will guarantee the accomplishment of a given goal, we are again led to intractable combinatorial problems. Therefore, practical communications protocols can only be designed on a "good" heuristic or ad-hoc basis, and they should not be expected to be optimal; approximate optimality is probably a more meaningful goal. Again, allowing some redundancy in on-line communications may lead to substantial savings in off-line computations.

## 6. References

- Ekchian, L.K., (1982) "Distributed Detection and Communications Problems," Ph.D. Thesis, Dept. of EECs, MIT, Cambridge, MA.
- Ekchian, L.K., Tenney, R.R., (1982), "Detection Networks," Technical Report LIDS-P-1191, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.
- Garey, M.R., D.S. Johnson, (1979), Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Co., San Francisco.
- Kushner, H.J., A. Pacut, (1982), "A Simulation Study of a Decentralized Detection Problem," IEEE Trans. on Auto. Control, Vol. AC-27, No. 5, pp. 1116-1119.
- Lauer, G.S., N.R. Sandell, Jr., (1983), "Distributed Detection of Signal Waveforms in Additive Gaussian Observation Noise," Technical Paper 160, Alphatech Inc., Burlington, Mass.
- Marschak, J., R. Radner, (1972), The Economic Theory of Teams, Yale University Press, New Haven, 1972.
- Papadimitriou, C.H., K. Steiglitz, (1982), Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall.
- Papadimitriou, C.H., J. Tsitsiklis, (1983), "On the Complexity of Designing Distributed Protocols," to appear in Information and Control.
- Simon, H.A., (1980), The Sciences of the Artificial, MIT Press, Cambridge, MA.
- Tenney, R.R., (1983), "Optimal Decentralized Control of Finite Nondeterministic Systems," Proceedings of the 1983 American Control Conference.
- Tenney, R.R., N.R. Sandell, Jr., (1981), "Detection with Distributed Sensors," IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-17, No. 4, pp. 501-509.
- Tsitsiklis, J.N., (1983), "Problems in Distributed Decision Making," Technical Report, Lab. for Information and Decision Systems, MIT, in preparation.
- Yao, A.C.-C., (1979), "Some Complexity Questions Related to Distributed Computing," Proceedings of the 11th Annual Symposium on Theory of Computing, pp. 209-213.



**END**

**FILMED**

**11-84**

**DTIC**