August 1986 LIDS-P-1617

COMMUNICATION COMPLEXITY OF CONVEX OPTIMIZATION*

John N. Tsitsiklis
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139
U.S.A.

Zhi-Quan Luo
Operations Research Center
Massachusetts Institute of Technology
Cambridge, MA 02139
U.S.A.

ABSTRACT

We consider a situation where each one of two processors has access to a different convex function f_i, $i = 1, 2$, defined on a common bounded domain. The processors are to exchange a number of binary messages, according to some protocol, until they find a point in the domain at which $f_1 + f_2$ is minimized, within some prespecified accuracy ϵ. Our objective is to determine protocols under which the number of exchanged messages is minimized.

I. INTRODUCTION.

Let $\mathcal{F}$ be a set of convex functions defined on the n–dimensional bounded domain $[0,1]^n$. (Typically, $\mathcal{F}$ will be defined by imposing certain smoothness conditions on its elements.) Given any $\epsilon > 0$, and $f \in \mathcal{F}$, let $I(f;\epsilon)$ be the set of all $x \in [0,1]^n$ such that $f(x) \leq f(y) + \epsilon$, $\forall y \in [0,1]^n$.

Let there be two processors, denoted by P_1 and P_2. Each processor is given a function $f_i \in \mathcal{F}$. Then they start exchanging binary messages, according to some protocol π, until processor P_1 determines an element of $I(f_1 + f_2; \epsilon)$. Let $C(f_1, f_2; \epsilon, \pi)$ be the total number of messages that are exchanged; this is a function of the particular protocol being employed and we are looking for an optimal one. More precisely, let

$$C(\mathcal{F}; \epsilon, \pi) = \sup_{f_1, f_2 \in \mathcal{F}} C(f_1, f_2; \epsilon, \pi) \tag{1.1}$$

* Research supported by the Army Research Office under Contract DAAAG–29–84–K–0005 and by the Office of Naval Research under contract N00014–84–K–0519 (NR 649–003). This paper will be presented at the 25th IEEE Conference on Decision and Control, Athens, Greece, December 1986.

be the communication requirement (in the worst case) of the particular protocol and let

$$C(\mathcal{F};\epsilon) = \inf_{\pi\in\Pi(\epsilon)} C(\mathcal{F};\epsilon,\pi) \tag{1.2}$$

be the communication requirement under an optimal protocol, where $\Pi(\epsilon)$ is the class of all protocols which work properly, for a particular choice of ϵ. The quantity $C(\mathcal{F};\epsilon)$ may be called the *ϵ-communication complexity* of the above defined problem of distributed, approximate, convex optimization.

For the above definition to be precise, we need to be specific regarding the notion of a protocol; that is, we have to specify the set $\Pi(\epsilon)$ of admissible protocols and this is what we do next. A protocol π consists of

a) A termination time T;

b) A collection of functions $M_{i,t} : \mathcal{F} \times \{0,1\}^t \mapsto \{0,1\}$, $i = 1,2$, $t = 0,1,2,...,T-1$;

c) A final function $Q : \mathcal{F} \times \{0,1\}^T \mapsto [0,1]^n$.

A protocol corresponds to the following sequence of events. Each processor P_i, receives its "input" f_i and then, at each time t, transmits to the other processor P_j a binary message $m_i(t)$ determined by

$$m_i(t) = M_{i,t}(f_i, m_j(0), ..., m_j(t-1)).$$

Thus the message transmitted by a processor is only a function of the function f_i known by it, together with all messages it has received in the past. At time T the exchange of messages ceases and processor P_1 picks a point in $[0,1]^n$ according to

$$x = Q(f_1, m_2(0), ..., m_2(T-1)). \tag{1.3}$$

The number $C(f_1, f_2; \epsilon, \pi)$ of messages transmitted under this protocol is simply $2T$. We define $\Pi(\epsilon)$ as the set of all protocols with the property that the point x generated by (1.3) belongs to $I(f_1 + f_2; \epsilon)$, for every $f_1, f_2 \in \mathcal{F}$.

A couple of remarks on our definition of protocols are in order.

(i) We have constrained each processor to transmit exactly one binary message at each stage. This may be wasteful if, for example, a better protocol may be found in which P_1 first sends many messages and then P_2 transmit its own messages. Nevertheless, the waste that results can be at most a factor of two. Since, in this paper, we study only orders of magnitude, this issue is unimportant.

(ii) We have assumed that the termination time T is the same for all $f_1, f_2 \in \mathcal{F}$, even though for certain "easy" functions the desired result may have been obtained earlier. Again, this is of no concern, given that we are interested in a worst case analysis.

Related Research:

The study of communication complexity was initiated in [1–2]. Reference [1] deals with problems of continuous variables, in which an exact result is sought, and allows the messages to be real–valued, subject to a constraint that they are smooth functions of the input. This is a different type of problem from ours, because we are interested in an approximate result and we are assuming binary messages.

Reference [2] deals with combinatorial problems, in which messages are binary and an exact result is obtained after finitely many stages. This reference has been followed by a substantial amount of research which developed the theory further and also evaluated the communication complexity of selected combinatorial problems [3–8]. The main application of this research has been in VLSI, where communication complexity constrains the amount of information that has to flow from one side of a chip to the other; this in turn determines certain tradeoffs on the achievable performance of special purpose VLSI chips for computing certain functions [8].

Finally, communication complexity has been also studied for models of asynchronous distributed computation, in which messages may reach their destination after an arbitrary delay [9].

The communication complexity of the approximate solution of problems of continuous variables has not been studied before, to the best of our knowledge. However, there exists a large amount of theory on the information requirements for solving (approximately) certain problems such as non-linear optimization, and numerical integration of differential equations [11–12] ("information based complexity"). Here one raises questions such as: how many gradient evaluations are required for an algorithm to find a point which minimizes a convex function within some prespecified accuracy ϵ? We can see that, in this type of research, information flows one–way: from a "memory unit" (which knows the function being minimized) to the processor and this is what makes it different from ours.

Outline:

In Section II we establish straightforward lower bounds such as $C(\mathcal{F};\epsilon) \geq O(n\log(1/\epsilon))$. In Section III we show that the naive distributed version of ellipsoid–type algorithms leads to protocols with $O(n^2\log(1/\epsilon)(\log n + \log(1/\epsilon))$ communication requirements and we show that this upper bound cannot be improved substantially within a restricted class of protocols. In Sections IV and V we partially close the gap between the above mentioned upper and lower bounds by presenting protocols with $O(\log(1/\epsilon))$ communication requirements for the case $n = 1$ (Section IV) and with $O(n(\log n + \log(1/\epsilon))$ communication requirements for the case of general n (Section V), under certain regularity assumptions on the elements of $\mathcal{F}$. In Section VI, we provide some discussion of possible extensions and questions which remain open.

II. LOWER BOUNDS ON $C(\mathcal{F};\epsilon)$.

Before we prove any lower bounds we start with a fairly trivial Lemma whose proof is omitted.

Lemma 2.1: If $\mathcal{F} \subset \mathcal{G}$ then $C(\mathcal{F};\epsilon) \leq C(\mathcal{G};\epsilon)$.

Let $\mathcal{F}_Q$ be the set of quadratic functions of the form $f(x) = ||x - x^*||^2$, with $x^* \in [0,1]^n$ and where $||\cdot||$ is the Euclidean norm. Also, let $\mathcal{F}_M$ be the set of functions of the form $f(x) = \max_{i=1,\ldots,n} |x - x_i^*|$, where $|x_i^*| \leq 1$, $\forall i$.

Proposition 2.2: (i) $C(\mathcal{F}_Q;\epsilon) \geq O(n(\log n + \log(1/\epsilon)))$;

(ii) $C(\mathcal{F}_M;\epsilon) \geq O(n\log(1/\epsilon))$.

Proof: (i) Consider a protocol $\pi \in \Pi(\epsilon)$ with termination time T and let us study its operation for the special case where $f_1 = 0$. Let S be the range of the function Q corresponding to that protocol (see equation (1.3)), when $f_1 = 0$. Given that the minimum of f_2 may be anywhere in $[0,1]^n$, S must contain points which come within $\epsilon^{1/2}$, in Euclidean distance, from every point in $[0,1]^n$. Now, one needs at least $(\frac{An}{\epsilon^{1/2}})^{Bn}$ Euclidean balls of radius $\epsilon^{1/2}$ to cover $[0,1]^n$, where A and B are absolute constants. (This follows by simply taking into acount the volume of a ball in n–dimensional space.) Therefore, the cardinality of S is at least $(An/\epsilon^{1/2})^{Bn}$. Given that the cardinality of the range of a function is no larger than the cardinality of its domain, it follows that the cardinality of S is no larger than 2^T. Therefore, $T \geq O(n(\log n + \log(1/\epsilon))$, which proves part (i).

(ii) The proof is almost identical to that of part (i) and is therefore omitted. The only difference is that now $[0,1]^n$ is covered by balls in the supremum norm and $O((1/\epsilon)^n)$ such balls are necessary and sufficient. •

Using Lemma 2.1, we conclude that $C(\mathcal{F}_{SC,M,L};\epsilon) \geq O(n(\log n + \log(1/\epsilon))$, where $\mathcal{F}_{SC,M,L}$ is the set of all continuously differentiable convex functions f with the property

$$L||x - y||^2 \leq \langle f'(x) - f'(y)|x - y\rangle \leq ML||x - y||^2, \tag{2.1}$$

and such that the norm of their gradient is bounded by $n^{1/2}$.

We also conclude that $C(\mathcal{F}_L;\epsilon) \geq O(n\log(1/\epsilon)$, where $\mathcal{F}_L$ is the set of differentiable convex functions which are bounded by $1/2$ and satisfy

$$|f(x) - f(y)| \leq \frac{1}{2}\max_i |x_i - y_i|, \quad \forall x, y.$$

In the proof of Proposition 2.2 we made use of the assumption that the final result is always obtained by processor P_1. Nevertheless, at the cost of minor complications of the proof, the same lower bound may be obtained even if we enlarge the class of allowed protocols so that the processor who computes the final result is not prespecified.

III. NAIVE UPPER BOUNDS.

We consider here a straightforward distributed version of the method of the centers of gravity (MCG), which has been shown in [11] to be an optimal algorithm for the single–processor case, in the sense that it requires a minimal number of gradient evaluations. This method may be

viewed as a generalization of the well–known ellipsoid algorithm for linear programming [10]. We start by describing the uniprocessor version of this method and then analyze the communication requirements of a distributed implementation.

The MCG Algorithm:[11, p. 62] Let f be a convex function to be minimized with accuracy ϵ. Let $G_0 = [0,1]^n$ and let x_0 be its center of gravity. At the beginning of the k-th stage of the computation, we assume that we are given a convex set $G_{k-1} \subset [0,1]^n$ and its center of gravity x_k. Let z_k be a scalar and let y_k be a vector in R^n with the following properties:
(i) $z_k + \langle y_k, x - x_k \rangle \leq f(x), \forall x \in [0,1]^n$;
(ii) $z_k \geq f(x_k) - (\epsilon/2)$.
(Notice that if the term $\epsilon/2$ was absent in condition 2, then we would have $z_k = f(x_k)$ and y_k would be a subgradient of f at x_k. The presence of the $\epsilon/2$ term implies that these relations only need to hold approximately.)
Let $a_k = \min_{j \leq k}\{z_j\}$ and let $G_k = \{x \in G_{k-1} : \langle y_k, x - x_k \rangle + z_k \geq a_k\}$. The algorithm terminates when the Lebesgue volume of G_k becomes smaller than $(\epsilon/2)^n$ and returns a point x_j which has the smallest value of z_j encountered so far.

The following facts are quoted from [11]:
(a) The volume of G_k is no larger than α^k, where α is an absolute constant, smaller than one and independent of the dimension n. Thus a total of $n\frac{\log(2/\epsilon)}{\log(1/\alpha)} = O(n\log(1/\epsilon))$ stages are sufficient.
(b) The result x_j of the algorithm satisfies $f(x_j) \leq \inf_{x \in [0,1]^n} f(x) + \epsilon V(f)$, where $V(f) = \sup_{x \in [0,1]^n} f(x) - \inf_{x \in [0,1]^n} f(x)$.

Notice that $V(f) \leq 1$, for $f = f_1 + f_2$, $f_1, f_2 \in \mathcal{F}_L$ so that the algorithm indeed produces a result belonging to $I(f;\epsilon)$.

We now consider a distributed implementation of this algorithm. The distributed protocol will consist of stages corresponding to the stages of the MCG algorithm. At the beginning of the k-th stage, both processors know the current convex set G_{k-1} and are therefore able to compute its center of gravity x_k. The i-th processor evaluates $f_i(x_k)$ and transmits the binary representation of a message $b(i,k)$ satisfying $b(i,k) \in [f_i(x_k) - (\epsilon/4), f_i(x_k) - (\epsilon/8)]$. Clearly, $b(i,k)$ may be chosen so that its binary representation has at most $O(\log(1/\epsilon))$ bits. Also each processor evaluates a subgradient $g_{i,k}$ of its function f_i, at x_k (with components $g_{i,k,j}$, $j = 1, ..., n$) and transmits the binary representation of messages $c(i,k,j)$ satisfying $|g_{i,k,j} - c(i,k,j)| \leq \epsilon/(16n)$. Clearly the $c(i,k,j)$'s may be chosen so that they can be all transmitted using $O(n\log(n/\epsilon)) = O(n\log n + n\log(1/\epsilon))$ bits.

Next, each processor lets $z_k = b(1,k) + b(2,k)$ and lets y_k be the vector with components $c(1,k,j) + c(2,k,j)$. It then follows by some simple algebra that z_k and y_k satisfy the specifications of the MCG algorithm. Finally, each processor determines G_k, and its center of gravity x_{k+1} and the algorithm proceeds to its next stage.

We now combine our estimates of the number of stages of the MCG algorithm and of the

communication requirements per stage to conclude the following.

Proposition 3.1: $C(\mathcal{F}_L;\epsilon) \leq O(n^2 \log(1/\epsilon)(\log n + \log(1/\epsilon))$. This bound is attained by the distributed version of the MCG algorithm.

The upper bound of Proposition 3.1 is quite far from the lower bound of Proposition 2.2. We show next that within a certain class of protocols this upper bound cannot be substantially improved.

We consider protocols which consist of stages. At the k-th stage there is a current point $x_k \in [0,1]^n$ known by both processors. Then, the processors transmit to each other approximate values of f_i and of a subgradient of f_i, all evaluated at x_k. Using the values of these messages, together with any past common information, they determine the next point x_{k+1}, according to some commonly known rule, and so on. We place one additional restriction: when a processor transmits an approximate value of $f_i(x_k)$ it does so by transmitting a sequence of bits of the binary representation of $f_i(x_k)$ starting from the most significant one and continuing with consecutive less significant bits. (So, for example, a processor is not allowed to transmit the first and the third most significant bits of $f_i(x_k)$, without transmitting the second most significant bit.) The same assumption is made concerning the components of the subgradient of f_i. Finally, we require that the same number of bits of $f_i(x_k)$ and of each component of the subgradient of f_i gets transmitted.

The above restrictions turn out to be quite severe.

Proposition 3.2: There exists a constant A such that for any protocol $\pi \in \Pi(\epsilon)$ satisfying the above restrictions, there exist $f_1, f_2 \in \mathcal{F}_L$ such that $C(f_1, f_2; \epsilon, \pi) \geq An^2 \log^2(1/\epsilon)$. This is true, even if we restrict f_1 to be equal to the identically zero function.

Proof: Using Lemma 2.1, it is sufficient to prove the result under the restriction that $f_1 = 0$ and under the restriction that f_2 is differentiable and bounded, together with every component of its derivative by $\epsilon^{1/2}$. Using the results of [11], for processor P_1 to determine a point which is optimal within ϵ, it must acquire nontrivial information on the values and the derivatives of f_2 for at least $An\log(1/\epsilon^{1/2})$ different points. Notice that the $O(\log(\epsilon^{1/2}))$ most significant bits of f_2 and each component of its derivative, evaluated at any point, are always zero. Thus, for processor P_1 to obtain nontrivial information at a certain point at least $O(n\log(1/\epsilon^{1/2}))$ bits have to be transmitted. This leads to a total communication requirement of $O(n^2 \log^2(1/\epsilon^{1/2})) = O(n^2 \log^2(1/\epsilon))$ bits, which proves the result. •

If we relax the requirement that the same number of bits is transmitted for each component of the subgradient, at each stage, then the same proof yields the lower bound $C(f_1, f_2; \epsilon, \pi) \geq An\log^2(1/\epsilon)$.

IV. AN OPTIMAL ALGORITHM FOR THE ONE–DIMENSIONAL CASE.

We prove here a result which closes the gap between upper and lower bounds for the one–dimensional case. The proof consists of the construction of an optimal protocol.

Proposition 4.1: If $n = 1$ then $C(\mathcal{F}_L;\epsilon) \leq O(\log(1/\epsilon))$.

Proof: The protocol consists of consecutive stages. At the beginning of the k-th stage, both processors have knowledge of four numbers a_k, b_k, c_k and d_k with the following properties:
(i) The interval $[a_k, b_k]$ contains a point x^* which minimizes $f_1 + f_2$.
(ii) The derivative of f_1 at any minimizer of f_1+f_2 and the derivative of f_1 and of $-f_2$ at $(a_k+b_k)/2$ belong to the interval $[c_k, d_k]$. (Notice that the derivative of each f_i has to be constant on the set of minimizers of $f_1 + f_2$.)

At the first stage of the algorithm we start with $a_1 = 0$, $b_1 = 1$, $c_1 = -1$ and $d_1 = 1$. At the k-th stage, the processors do the following: processor P_i transmits a message $m_{i,k} = 0$ if $(-1)^{i-1} f_i'((a_k + b_k)/2) \leq (c_k + d_k)/2$; otherwise it transmits $m_{i,k} = 1$.

If $m_{1,k} = 0$ and $m_{2,k} = 1$, then $f_1'((a_k + b_k)/2) + f_2'((a_k + b_k)/2) \leq 0$. We may then let $a_{k+1} = (a_k + b_k)/2$ and leave b_k, c_k, d_k unchanged. Similarly, if $m_{1,k} = 1$ and $m_{2,k} = 0$, we let $b_{k+1} = (a_k + b_k)/2$ and leave a_k, c_k, d_k unchanged.

We now consider the case $m_{1,k} = m_{2,k} = 1$. Let x^* be a minimizer of $f_1 + f_2$ belonging to $[a_k, b_k]$. If $x^* \geq (a_k + b_k)/2$, then $f_1'(x^*) \geq f_1'((a_k + b_k)/2) \geq (c_k + d_k)/2$. If $x^* \leq (a_k + b_k)/2$, then $f_1'(x^*) \geq -f_2'(x^*) \geq -f_2'((a_k + b_k)/2) \geq (c_k + d_k)/2$. In either case, we may let $c_{k+1} = (c_k + d_k)/2$ and leave a_k, b_k, d_k unchanged. Finally, if $m_{1,k} = m_{2,k} = 0$, a similar argument shows that we may let $d_{k+1} = (c_k + d_k)/2$ and leave a_k, b_k, c_k unchanged.

For each one of the four cases, we see that $a_k, ..., d_k$ will preserve properties (i), (ii) that were postulated earlier. Furthermore, at each stage, either $b_k - a_k$ or $d_k - c_k$ is halved. Therefore, after at most $k = 2\log(1/\epsilon)$ stages, we reach a point where either $b_k - a_k \leq \epsilon$ or $d_k - c_k \leq \epsilon$. If $b_k - a_k \leq \epsilon$, then there exists a minimizer which is within ϵ of a_k; given that the derivative of $f_1 + f_2$ is bounded by one, it follows that $f_1(a_k) + f_2(a_k)$ comes within ϵ of the optimum, as desired. Alternatively, if $d_k - c_k \leq \epsilon$, then $|f_1'((a_k + b_k)/2) + f_2'((a_k + b_k)/2)| \leq d_k - c_k \leq \epsilon$. It follows that for any $x \in [0, 1]$, we have $f_1(x) + f_2(x) \geq f_1((a_k + b_k)/2) + f_2((a_k + b_k)/2) - |x - (a_k + b_k)/2|\epsilon$, which shows that $(f_1 + f_2)((a_k + b_k)/2)$ comes within ϵ of the optimum. •

V. AN OPTIMAL PROTOCOL FOR STRONGLY CONVEX PROBLEMS.

We consider here the class $\mathcal{F}_{SC,M,L}$ of strongly convex problems, defined in Section II. However, we prefer to deal with unconstrained optimization and for this reason, we limit ourseleves to to the subset $\mathcal{F}_U$ of $\mathcal{F}_{SC,M,L}$ which contains only functions for which there exists a (necessarily unique) $x^* \in [0, 1]^n$ at which $f'(x^*) = 0$. This assumption is not unrealistic; in practical unconstrained optimization problems one is able to obtain a priori bounds on the region in which the optimum is lying. Then, by rescaling, this region may be assumed to be $[0, 1]^n$. We show below that an appropriate inexact implementation of the classical gradient algorithm turns out to lead to an optimal protocol, as far as the dependence on n and ϵ is concerned.

The protocol produces a sequence $\{x_k\}_{k=0}^T$ of points according to the iteration

$$x_{k+1} = x_k - \gamma s_k; \qquad x_k = 0, \tag{5.1}$$

where s_k is an approximation to g_k, the gradient of $f_1 + f_2$, evaluated at x_k. In particular, we will require that the inequality

$$||s_k - g_k|| \leq n^{1/2}\alpha^k \tag{5.2}$$

holds for all $k \leq T$, with some $\alpha \in (0,1)$. (Throughout this section $||\cdot||$ will stand for the Euclidean norm.) We first estimate the number of iterations required to reach an ϵ–optimal point.

Proposition 5.1: If the stepsize γ is small enough, and if α is sufficiently close to 1, then there exist $A > 0$, $B > 0$, depending only on M, L, such that

(a) $f(x_k) - f^* \leq An\alpha^k$, (5.3)

(b) $||x_{k+1} - x_k|| \leq Bn^{1/2}\alpha^k$. (5.4)

Proof: We state without proof the following properties of functions in $\mathcal{F}_{SC,M,L}$ [11, pp. 254–255]:

(i) $||f'(x) - f'(y)|| \leq ML||x - y||$. (5.5)

(ii) $f(x+y) \geq f(x) + \langle f'(x)|y\rangle + (L/2)||y||^2$. (5.6)

(iii) $f(x+y) \leq f(x) + \langle f'(x)|y\rangle + (LM/2)||y||^2$. (5.7)

Using inequality (5.7) together with (5.1) and (5.2) we obtain

$$f(x_{k+1}) \leq f(x_k) - \gamma\langle s_k, g_k\rangle + \gamma^2\frac{LM}{2}||s_k||^2 \leq$$

$$f(x_k) - \gamma||g_k||^2 + \gamma n^{1/2}\alpha^k + \gamma^2\frac{LM}{2}(||g_k||^2 + n\alpha^{2k}). \tag{5.8}$$

Let x^* be the minimizer of f. Using (2.1) we obtain $||g_k|| \geq L||x_k - x^*||$ and using (5.7) we obtain

$$f(x_k) - f(x^*) \leq \frac{ML}{2}||x_k - x^*||^2 \leq (M/2L)||g_k||^2.$$

Combining this with (5.8) we conclude that

$$f(x_{k+1}) - f(x^*) \leq f(x_k) - f(x^*) - (\gamma 2L/M)(1 - \gamma LM/2)(f(x_k) - f(x^*)) + \gamma n^{1/2}\alpha^k + n\gamma^2(LM/2)\alpha^{2k}.$$

Taking γ small enough so that $\beta = 1 - 2(\gamma L/M)(1 - \gamma LM/2) \in (0,1)$, and assuming that α was chosen larger than β, part (a) is easily proved by induction. (To start the induction, we use the inequality (5.7).) For part (b) we use inequality (5.6) to obtain

$$||x_k - x^*||^2 \leq \frac{2}{L}(f(x_k) - f(x^*))$$

and the result follows. •

As an immediate corollary of Proposition 5.1, we see that after $A(\log(1/\epsilon) + \log n)$ iterations of the approximate gradient algorithm (5.2), we have reached an ϵ–optimal point. Next we show how this algorithm may be implemented in a distributed manner with only $O(n)$ bits being communicated at each stage. All we need to do is to make sure that the processor share at each stage enough information to compute a vector s_k satisfying the bound (5.2). This is accomplished by having each processor know a set of scalars $s_k(i,j)$, $i = 1,2$, $j = 1,...,n$ such that $|s_k(i,j) - g_k(i,j)| \leq n^{1/2}\alpha^k$,

where $g_k(i,j)$ is the j-th component of $f_i'(x_k)$. At stage $k=0$ this is easy: $g_0(i,j)$ is bounded by $n^{1/2}$ and therefore the choice $s_0(i,j)=0$ will do. Suppose that quantities $s_k(i,j)$ with the desired properties have been shared at stage k and let us now consider stage $k+1$. We have $|g_{k+1}(i,j) - s_k(i,j)| \leq |g_{k+1}(i,j)-g_k(i,j)|+|g_k(i,j)-s_k(i,j)| \leq LM||x_{k+1}-x_k||+n^{1/2}\alpha^k \leq (LMB+1)n^{1/2}\alpha^k$, where the last inequality follows from part (b) of Proposition 5.1. We require that $s_{k+1}(i,j)$ be an integer multiple of $n^{1/2}\alpha^{k+1}$. This requirement does not prohibit the satisfaction of the constraint (5.2); furthermore, with this requirement, there are at most $(LMB+1)\alpha^{-1}+1$ possible choices for $s_{k+1}(i,j)$. Therefore, processor P_i may choose $s_{k+1}(i,j)$ as above and transmit its value to the other processor while communicating only a constant number of bits. This has to be done by each processor P_i and for each component j, for a total of $O(n)$ bits of communication per stage. We have thus proved the following result.

Proposition 5.2: $C(\mathcal{F}_U;\epsilon) \leq An(\log n + \log(1/\epsilon))$, where A is a constant depending only on M, L.

Notice that this result attains the lower bound of Section II.

VI. POSSIBLE EXTENSIONS AND OPEN QUESTIONS.

1. The protocol of Section V is likely to be far from optimal concerning the dependence on the parameters M and L. The gradient algorithm tends to be inefficient for poorly conditioned problems (large M), whereas a version of the conjugate gradient method requires an optimal number of iterations [11]. It remains to be seen whether a suitable approximate version of the conjugate gradient method admits a distributed implementation with low communication requirements.

2. In Section V we dealt essentially with unconstrained problems for which the optimal solution may be a priori localized into a bounded region. If we consider true constrained problems, we expect that a similar approximate version of the gradient projection method will have the same communication requirements as the gradient method.

3. For the class $\mathcal{F}_L$, gradient methods do not work and the gap between the lower bound of Section II and the upper bound of Section III remains open. We believe that the factor of n^2 in the upper bound cannot be reduced. The reason is that any conceivable algorithm would need to consider at least $O(n\log(1/\epsilon))$ points and it is hard to imagine of any useful transfer of information concerning the behavior of the function in the vicinity of a point which does not require $O(n)$. On the other hand, it may be possible to reduce the factor $\log^2(1/\epsilon)$ to just $\log(1/\epsilon)$ although we do not know how to accomplish this.

4. Another extension concerns the case of $K>2$ processors minimizing the function $f_1+...+f_K$. The protocols of Propositions 4.1 and 5.2 may be adjusted to obtain protocols with communication $O(K\log K)$, as far as the dependence on K is concerned. It is an open question whether this may be reduced to just $O(K)$.

REFERENCES.

1. Abelson, H., "Lower Bounds on Information Transfer in Distributed Computations", *Journal of the ACM,* 27, 2, 1980, pp. 384–392.

2. Yao, A.C., "Some Complexity Questions Related to Distributed Computing", *Proceedings of the 11th STOC,* 1979, pp. 209–213.

3. Papadimitriou, C.H. and Sipser, M., "Communication Complexity", *Proceedings of the 14th STOC,* 1982, pp. 196–200.

4. Papadimitriou, C.H., and Tsitsiklis, J.N., "On the Complexity of Designing Distributed Protocols", *Information and Control,* 53, 3, 1982, pp. 211–218.

5. Aho, A.V., Ullman, J.D., and Yannakakis, M., "On Notions of Information Transfer in VLSI Circuits", *Proceedings of the 15th STOC,* 1983, pp. 133–139.

6. Pang, K.F., El Gamal, A., "Communication Complexity of Computing the Hamming Distance", 1985 International Symposium on Information Theory, Brighton, England, 1985.

7. Mehlhorn, K., Schmidt, E.M., "Las Vegas is Better than Determinism in VLSI and Distributed Computing", *Proceedings of the 14th STOC,* 1982, pp. 330–337.

8. Ullman, J.D., *Computational Aspects of VLSI,* Computer Science Press, 1984.

9. Awerbuch, B., Gallager, R.G., "Communication Complexity of Distributed Shortest Path Algorithms, Technical Report LIDS–P–1473, Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA, 1985.

10. Papadimitriou, C.H., Steiglitz, K., *Combinatorial Optimization: Algorithms and Complexity,* Prentice Hall, 1982.

11. Nemirovsky, A.S., Yudin, D.B., *Problem Complexity and Method Efficiency in Optimization,* Wiley, 1983.

12. Traub, J.F., Wozniakowski, H., *A General Theory of Optimal Algorithms,* Academic Press, 1980.