Communication Complexity of Convex Optimization*

JOHN N. TSITSIKLIS AND ZHI-QUAN LUO

Laboratory for Information and Decision Systems and the Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

We consider a situation where each 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. © 1987 Academic Press, Inc.

1. 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 $\varepsilon > 0$, and $f \in \mathcal{F}$, let $I(f; \varepsilon)$ be the set of all $x \in [0, 1]^n$ such that $f(x) \leq f(y) + \varepsilon$, $\forall y \in [0, 1]^n$.

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 $I(f_1 + f_2; \varepsilon)$. Let $C(f_1, f_2; \varepsilon, \pi)$ 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

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

* 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). A preliminary version of this paper was presented at the 25th IEEE Conference on Decision and Control, Athens, Greece, Dec. 1986.

0885-064X/87 $3.00

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

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

be the communication requirement under an optimal protocol, where $\Pi(\varepsilon)$ is the class of all protocols which work properly, for a particular choice of ε. The quantity $C(\mathcal{F}; \varepsilon)$ 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 $\Pi(\varepsilon)$ 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 $M_{i,t} : \mathcal{F} \times \{0, 1\}^t \mapsto \{0, 1\}$, $i = 1, 2$, $t = 0, 1, 2, \ldots, T - 1$;

(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

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

Thus the message transmitted by a processor depends only on the function f_i known by it, together will 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

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

The number $C(f_1, f_2; \varepsilon, \pi)$ of messages transmitted under this protocol is simply $2T$. We define $\Pi(\varepsilon)$ as the set of all protocols with the property that the point x generated by (1.3) belongs to $I(f_1 + f_2; \varepsilon)$, 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 tissue 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 because we are interested in a worst case analysis.

Related Research

The study of communication complexity was initiated by Abelson (1980) and Yao (1979). Abelson 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.

Yao (1979) 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 (Papadimitriou and Sipser, 1982; Papadimitriou and Tsitsiklis, 1982; Aho *et al.,* 1983; Pang and El Gamal, 1986; Mehlhorn and Schmidt, 1982; Ullman, 1984). 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 trade-offs on the achievable performance of special-purpose VLSI chips for computing certain functions (Ullman, 1984).

Finally, communication complexity has been also studied for models of asynchronous distributed computation, in which messages may reach their destination after an arbitrary delay (Awerbuch and Gallager, 1985).

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 nonlinear optimization, and numerical integration of differential equations (Nemirovsky and Yudin, 1983; Traub and Woźniakowski, 1980) ("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 $C(\mathcal{F}; \varepsilon) \geq O(n \log(1/\varepsilon))$. In Section III we show that the naive distributed version of ellipsoid-type algorithms leads to protocols with $O(n^2 \log(1/\varepsilon)(\log n +$

$\log(1/\varepsilon)$) 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 $O(\log(1/\varepsilon))$ communication requirements for the case $n = 1$ (Section IV) and with $O(n \log n(\log n + \log(1/\varepsilon))$ 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 $C(\mathcal{F}; \varepsilon)$

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* $C(\mathcal{F}; \varepsilon) \leq C(\mathcal{G}; \varepsilon)$.

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 $f(x) = \frac{1}{4}\max_{i=1,\ldots,n} |x - x_i^*|$, where $|x_i^*| \leq 1, \forall i$.

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

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

Proof. (i) Consider a protocol $\pi \in \Pi(\varepsilon)$ 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 Eq. (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 $\varepsilon^{1/2}$, in Euclidean distance, from every point in $[0, 1]^n$. Now, one needs at least $(An/\varepsilon^{1/2})^{Bn}$ Euclidean balls of radius $\varepsilon^{1/2}$ to cover $[0, 1]^n$, where A and B are absolute constants. (This follows by simply taking into account the volume of a ball in n-dimensional space.) Therefore, the cardinality of S is at least $(An/\varepsilon^{1/2})^{Bn}$. 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, $T \geq O(n(\log n + \log(1/\varepsilon))$, which proves the first part.

(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 $O((1/\varepsilon)^n)$ such balls are necessary and sufficient. ∎

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.

Let $\mathcal{F}_{\mathrm{SC},M,L}$ ("strongly convex functions") be the set of all continuously differentiable convex functions f with the properties

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

$$\|f'(x)\| \leq MLn^{1/2}, \qquad \forall x \in [0, 1]^n. \tag{2.2}$$

(Note that (2.1) implies that $M \geq 1$.) Also, let $\mathcal{F}_L$ be the set of convex functions which are bounded by $\frac{1}{4}$ and satisfy

$$|f(x) - f(y)| \leq \tfrac{1}{4} \max_i |x_i - y_i|, \qquad \forall x, y.$$

PROPOSITION 2.3. (i) $C(\mathcal{F}_{\mathrm{SC},M,L}; \varepsilon) \geq O(n(\log n + \log(1/\varepsilon))$.

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

Proof. Part (ii) follows from Proposition 2.2 and Lemma 2.1, because $\mathcal{F}_M \subset \mathcal{F}_L$. For part (i), we note that $\mathcal{F}_Q \subset \mathcal{F}_{\mathrm{SC},M,2}$ and Lemma 2.1 proves the result for $\mathcal{F}_{\mathrm{SC},M,2}$. The result for general L follows because any $f \in \mathcal{F}_{\mathrm{SC},M,L}$ can be scaled so that it belongs to $\mathcal{F}_{\mathrm{SC},M,2}$. ■

III. NAIVE UPPER BOUNDS

We consider here a straightforward distributed version of the method of the centers of gravity (MCG), which has been shown by Nemirovsky and Yudin (1983) to be an optimal algorithm in the single-processor case, for functions in $\mathcal{F}_L$, 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 (Papadimitriou and Steiglitz, 1982). We start by describing the uniprocessor version of this method and then analyze the communication requirements of a distributed implementation.

The MCG Algorithm (Nemirovsky and Yudin, 1983, *p.* 62)

Let $f \in \mathcal{F}_L$ 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 kth 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) $z_k \geq f(x_k) - (\varepsilon/2)$.

(Note that if the term $\varepsilon/2$ were absent in condition (ii), we would have $z_k =$

$f(x_k)$ and $y_k = f'(x_k)$, if x_k is an interior point. The presence of the $\varepsilon/2$ term implies that these relations need to hold only approximately.)

Let $a_k = \min_{j \leq k}\{z_j\}$ and let $G_k = \{x \in G_{k-1} : \langle y_k, x - x_k \rangle + z_k \leq a_k\}$. The algorithm terminates when the Lebesgue volume of G_k becomes smaller than $(\varepsilon/2)^n$ and returns a point x_j associated with the smallest value of z_j encountered so far.

The following facts are quoted from Nemirovsky and Yudin (1983).

(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 $n(\log(2/\varepsilon)/\log(1/\alpha)) = O(n \log(1/\varepsilon))$ stages are sufficient.

(b) The result x_j of the algorithm satisfies $f(x_j) \leq \inf_{x \in [0,1]^n} f(x) + \varepsilon V(f)$, where $V(f) = \sup_{x \in [0,1]^n} f(x) - \inf_{x \in [0,1]^n} f(x)$.

Note 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 $I(f; \varepsilon)$.

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 kth stage, both processors know the current convex set G_{k-1} and are therefore able to compute its center of gravity x_k. Processor P_i evaluates $f_i(x_k)$ and transmits the binary representation of a message $b(i, k)$ satisfying $b(i, k) \in [f_i(x_k) - (\varepsilon/4), f_i(x_k) - (\varepsilon/8)]$. Clearly, $b(i, k)$ may be chosen so that its binary representation has at most $O(\log(1/\varepsilon))$ bits. Also, each processor evaluates the gradient $g_{i,k}$ of its function f_i, at x_k (with components $g_{i,k,j}$, $j = 1, \ldots, n$) and transmits the binary representation of messages $c(i, k, j)$ satisfying $|g_{i,k,j} - c(i, k, j)| \leq \varepsilon/(16n)$. Clearly the $c(i, k, j)$'s may be chosen so that they can be all transmitted using $O(n \log(n/\varepsilon)) = O(n \log n + n \log(1/\varepsilon))$ 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. $C(\mathcal{F}_L; \varepsilon) \leq O(n^2 \log(1/\varepsilon)(\log n + \log(1/\varepsilon))$. *In particular, the above-described distributed version of the MCG algorithm stays within this bound.*

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 kth 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 the gradient 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 gradient of f_i. Finally, we require that the same number of bits of $f_i(x_k)$ and of each component of the gradient of f_i get transmitted.

The above restrictions turn out to be quite severe.

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

Proof. Using an argument similar to Lemma 2.1, it is sufficient to prove the result under the restriction that $f_1 = 0$ and under the restriction that f_2 be differentiable and bounded, together with every component of its derivative, by $\varepsilon^{1/2}$. Using the results of Nemirovsky and Yudin (1983), 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 $An \log(1/\varepsilon^{1/2})$ different points. Note that the $O(\log(\varepsilon^{1/2}))$ 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 $O(n \log(1/\varepsilon^{1/2}))$ bits have to be transmitted. This leads to a total communication requirement of $O(n^2 \log^2(1/\varepsilon^{1/2})) = O(n^2 \log^2(1/\varepsilon))$ bits, which proves the result. ■

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

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. We only present the protocol under the assumption that each f_i is differentiable. The argument is the same in the nondifferentiable case, except that each f_i' is to be interpreted as a subgradient.

PROPOSITION 4.1. *If $n = 1$ then $C(\mathcal{F}_L; \varepsilon) \leq O(\log(1/\varepsilon))$.*

Proof. The protocol consists of consecutive stages. At the beginning of the kth 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 $[c_k, d_k]$. (Note 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 kth stage, the processors do the following: processor P_i transmits a message $m_{i,k} = 0$ if $(-1)^{i-1}f_i'((a_k + b_k)/2) \le (c_k + d_k)/2$; otherwise it transmits $m_{i,k} = 1$.

If $m_{1,k} = 0$ and $m_{2,k} = 1$, then $f_1'((a_k + b_k)/2) + f_2'((a_k + b_k)/2) \le 0$. 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 $x^* \ge (a_k + b_k)/2$, then $f_1'(x^*) \ge f_1'((a_k + b_k)/2) \ge (c_k + d_k)/2$. If $x^* \le (a_k + b_k)/2$, then $f_1'(x^*) \ge -f_2'(x^*) \ge -f_2'((a_k + b_k)/2) \ge (c_k + d_k)/2$. 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 of the four cases, we see that $a_k, \ldots, d_k$ will preserve properties (i), (ii), which were postulated earlier. Furthermore, at each stage, either $b_k - a_k$ or $d_k - c_k$ is halved. Therefore, after at most $k = 2 \log(1/\varepsilon)$ stages, we reach a point where either $b_k - a_k \le \varepsilon$ or $d_k - c_k \le \varepsilon$. If $b_k - a_k \le \varepsilon$, 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 $d_k - c_k \le \varepsilon$, then $|f_1'((a_k + b_k)/2) + f_2'((a_k + b_k)/2)| \le d_k - c_k \le \varepsilon$. It follows that for any $x \in [0, 1]$, we have $f_1(x) + f_2(x) \ge f_1((a_k + b_k)/2) + f_2((a_k + b_k)/2) - |x - (a_k + b_k)/2|\varepsilon$, which shows that $(f_1 + f_2)((a_k + b_k)/2)$ comes within ε of the optimum. ■

V. An Almost Optimal Protocol for Strongly Convex Problems

We consider here the class $\mathcal{F}_{\mathrm{SC},M,L}$ of strongly convex functions which was defined in Section III as the set of continuously differentiable convex functions satisfying (2.1)–(2.2). In this section we show that a suitable distributed version of the gradient projection algorithm comes close to the lower bound of Proposition 2.3, within an $O(\log n)$ factor. In particular, for any fixed dimension n, we have a protocol whose dependence on ε is optimal.

In the protocol to be considered each processor computes the same sequence $\{x_k\}$ of elements of $[0, 1]^n$ according to the iteration

$$x_{k+1} = [x_k - \gamma s_k]_+ ; \qquad x_0 = 0. \tag{5.1}$$

We use the notation $[y]_+$ to denote the projection (with respect to the Euclidean metric) of a vector $y \in \Re^n$ onto the convex set $[0, 1]^n$. Also, γ is a positive scalar stepsize and s_k is an approximation of the gradient of $f_1 + f_2$, evaluated at x_k. In particular, we let $g_k = f_1'(x_k) + f_2'(x_k)$ and we require that s_k satisfy

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

where α is some positive constant, independent of k, belonging to (0, 1). Naturally, we will have to ensure that there is enough communication so that each processor knows s_k at the beginning of the kth stage.

We start by estimating the number of steps required by the above algorithm to come to a small neighborhood of the optimal point. The argument is very similar to the standard proof that the gradient projection algorithm has a linear rate of convergence (Nemirovsky and Yudin, 1983, pp. 258–260) except that we need to take care of the fact that we use s_k instead of the exact gradient g_k. We denote by x^* the unique vector in $[0, 1]^n$ which minimizes $f_1 + f_2$ over that domain. (Uniqueness is a consequence of strict convexity, which follows from strong convexity.)

Proposition 5.1. *If $f \in \mathcal{F}_{\mathrm{SC},M,L}$, if x_k, s_k satisfy* (5.1)–(5.2), *if the stepsize γ is small enough, and if α is sufficiently close to* 1, *then there exist A, B, $C > 0$, depending only on M, L, such that*

$$\text{(i)}\ \ f(x_k) - f(x^*) \leq An\alpha^{2k}, \tag{5.3}$$

$$\text{(ii)}\ \ \|x_k - x^*\|^2 \leq Bn\alpha^{2k}, \tag{5.4}$$

$$\text{(iii)}\ \ \|x_{k+1} - x_k\| \leq Cn^{1/2}\alpha^k. \tag{5.5}$$

Proof. We will prove the result with the following choices of constants: we let $\gamma = 1/(ML)$, $B = 2A/L$, and $C = 2B^{1/2}$. The constants A and α will be fixed later.

We state without proof the following properties of functions in $\mathcal{F}_{\mathrm{SC},M,L}$ (Nemirovsky and Yudin, 1983, pp. 254–255):

$$\text{(i)}\ \ \|f'(x) - f'(y)\| \leq ML\|x - y\|. \tag{5.6}$$

$$\text{(ii) } f(x + y) \geq f(x) + \langle f'(x)|y\rangle + (L/2)\|y\|^2. \tag{5.7}$$

$$\text{(iii) } f(x + y) \leq f(x) + \langle f'(x)|y\rangle + (LM/2)\|y\|^2. \tag{5.8}$$

We will be also using the inequality

$$\langle f'(x^*)|y - x^*\rangle \geq 0, \qquad \forall y \in [0, 1]^n, \tag{5.9}$$

which is a necessary and sufficient condition for optimality of x^*.

We continue with the main part of the proof, which proceeds by induction on k. We first show that part (i) holds for $k = 0$. Using the convexity of f, we have

$$f(x^*) \geq f(x_0) + \langle f'(x_0)|x^* - x_0\rangle \geq f(x_0) - \|f'(x_0)\|\cdot\|x^* - x_0\|.$$

Using (2.2), we see that $\|f'(x_0)\|$ is bounded by $MLn^{1/2}$; also, $\|x^* - x_0\| = \|x^*\|$ is bounded by $n^{1/2}$. It follows that $f(x_0) - f(x^*) \leq MLn \leq An$, as long as A is chosen larger than ML.

Suppose now that (5.3) is valid for some nonnegative integer k. Using (5.7) and then (5.9) we obtain

$$\begin{aligned} f(x_k) &\geq f(x^*) + \langle f'(x^*)|x_k - x^*\rangle + \frac{L}{2}\|x_k - x^*\|^2 \\ &\geq f(x^*) + \frac{L}{2}\|x_k - x^*\|^2. \end{aligned} \tag{5.10}$$

We now use (5.10) and the induction hypothesis to obtain

$$\|x_k - x^*\|^2 \leq \frac{2}{L}[f(x_k) - f(x^*)] \leq \frac{2}{L}An\alpha^{2k} = Bn\alpha^{2k}. \tag{5.11}$$

We have therefore shown that (5.4) is also valid for that particular k. We then use (5.4) and the triangle inequality to obtain

$$\|x_{k+1} - x_k\| \leq \|x_{k+1} - x^*\| + \|x_k - x^*\| \leq 2B^{1/2}n^{1/2}\alpha^k, \tag{5.12}$$

which proves (5.5) for that same value of k.

We now prove (5.3) for $k + 1$, which will complete the induction. Using the definition of the projection, x_{k+1} minimizes $\|y - x_k + \gamma s_k\|^2$ over $y \in [0, 1]^n$, which is equivalent to minimizing

$$f(x_k) + \langle s_k|y - x_k\rangle + \frac{1}{2\gamma}\|y - x_k\|^2 \tag{5.13}$$

over all $y \in [0, 1]^n$. Let us use the notation $J_k(y)$ to denote the expression (5.13) as a function of y. Let $z = x_k + (1/M)(x^* - x_k)$. Note that $z \in [0, 1]^n$ because x_k, x^* belong to $[0, 1]^n$. Thus, by the minimizing property of x_{k+1} we have

$$J_k(x_{k+1}) \le J_k(z). \tag{5.14}$$

Now,

$$\begin{aligned}
f(x_{k+1}) &\le f(x_k) + \langle g_k | x_{k+1} - x_k \rangle + \frac{LM}{2} \|x_{k+1} - x_k\|^2 \\
&\le f(x_k) + \langle s_k | x_{k+1} - x_k \rangle + \frac{LM}{2} \|x_{k+1} - x_k\|^2 + \|s_k - g_k\| \cdot \|x_{k+1} - x_k\| \\
&\le J_k(x_{k+1}) + (n^{1/2}\alpha^k)(2B^{1/2}n^{1/2}\alpha^k) \\
&\le J_k(z) + 2B^{1/2}n\alpha^{2k} \\
&= f(x_k) + \left\langle s_k \middle| \frac{1}{M}(x^* - x_k) \right\rangle + \frac{LM}{2}\frac{1}{M^2}\|x^* - x_k\|^2 + 2B^{1/2}n\alpha^{2k} \\
&\le f(x_k) + \left\langle g_k \middle| \frac{1}{M}(x^* - x_k) \right\rangle + \frac{L}{2M}\|x^* - x_k\|^2 + 2B^{1/2}n\alpha^{2k} \\
&\quad + \|s_k - g_k\| \frac{1}{M} \|x^* - x_k\| \\
&\le \left(1 - \frac{1}{M}\right) f(x_k) + \frac{1}{M}\left[f(x_k) + \langle g_k | x^* - x_k \rangle + \frac{L}{2}\|x^* - x_k\|^2 \right] \\
&\quad + 3B^{1/2}n\alpha^{2k} \\
&\le \left(1 - \frac{1}{M}\right) f(x_k) + \frac{1}{M} f(x^*) + 3B^{1/2}n\alpha^{2k}.
\end{aligned}$$

Here, the first inequality followed from (5.8); the second from the Schwarz inequality; the third from (5.2), (5.12), and the definition of $J_k(x_{k+1})$; the fourth from (5.14). In the equality we made use of the definition of z and J_k, and the next step followed from the Schwarz inequality; then, we used the fact $M \ge 1$, (5.2), and (5.11); finally, the last line followed from (5.7). We therefore have, using the induction hypothesis,

$$\begin{aligned}
f(x_{k+1}) - f(x^*) &\le \left(1 - \frac{1}{M}\right)(f(x_k) - f(x^*)) + 3B^{1/2}n\alpha^{2k} \\
&\le \left(1 - \frac{1}{M}\right) An\alpha^{2k} + 3\left(\frac{2A}{L}\right)^{1/2} n\alpha^{2k}.
\end{aligned} \tag{5.15}$$

The induction will be completed if the right-hand side of (5.15) is smaller than $An\alpha^{2(k+1)}$. This is accomplished by taking $\alpha \in (0, 1)$ close enough to 1 so that $1 - 1/M < \alpha^2$ and then choosing A large enough so that the term involving $A^{1/2}$ is negligible in comparison with the first term in the right-hand side of (5.15). This concludes the proof. ■

We now return to the distributed protocol. Since $f_1, f_2 \in \mathcal{F}_{\mathrm{SC},M,L}$, it follows that $f_1 + f_2 \in \mathcal{F}_{\mathrm{SC},M,2L}$. Consequently, Proposition 5.1 applies to $f_1 + f_2$ and shows that after $O(\log(1/\varepsilon) + \log n)$ stages, the algorithm (5.1)–(5.2) reaches a point which is within ε from optimality.

We now indicate how the protocol may be implemented with $O(n \log n)$ bits being communicated at each stage. All we need to do is to make sure that the processors share enough information at each stage to be able to compute a vector s_k satisfying (5.2). This is accomplished by letting each processor know a set of scalars $s_k(i, j)$, $i = 1, 2, j = 1, \ldots, n$, such that $|s_k(i, j) - g_k(i, j)| \leq \alpha^k$, where $g_k(i, j)$ is the jth component of $f_i'(x_k)$. We first consider stage $k = 0$. Using (2.2) we see that $|g_0(i, j)|$ is bounded by $O(n^{1/2})$, for each i, j. Therefore, it is sufficient to transmit $O(\log n)$ bits, to specify each component with accuracy $\alpha^0 = 1$.

Suppose now that $k > 0$ and that quantities $s_{k-1}(i, j)$ with the desired properties have been shared at stage $k - 1$. We have $|g_k(i, j) - s_{k-1}(i, j)| \leq |g_k(i, j) - g_{k-1}(i, j)| + |g_{k-1}(i, j) - s_{k-1}(i, j)| \leq LM\|x_k - x_{k-1}\| + n^{1/2}\alpha^{k-1} \leq (LMCn^{1/2} + n^{1/2})\alpha^{k-1}$. (Here we have made use of (5.6), our hypothesis that s_k satisfies (5.2), and part (iii) of Proposition 5.1.) Let us impose the additional requirement that $s_k(i, j)$ be an integer multiple of α^k. This requirement does not prohibit the attainment of our goal, which is to satisfy inequality (5.2). With this requirement, there are at most $\alpha^{-1}(LMCn^{1/2} + 1) + 1$ possible choices for $s_k(i, j)$. Therefore, each processor P_i may choose $s_k(i, j)$ as above and transmit its value to the other processor, while communicating only $O(\log n)$ bits for each component j, thus leading to a total of $O(n \log n)$ communications per stage. We have thus proved the following result.

PROPOSITION 5.2. *For any fixed M, L, we have* $C(\mathcal{F}_{\mathrm{SC},M,L}; \varepsilon) \leq O(n \log n(\log n + \log(1/\varepsilon)))$.

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), as opposed to variations of the conjugate gradient method (Nemirovsky and Yudin, 1983). It remains to be seen whether a suitable approximate version of the conjugate gradient method admits a distributed implementation with low communication requirements as a function of M.

2. 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 $O(n \log(1/\varepsilon))$ points and it is hard to imagine any useful transfer of information concerning the behavior of the function in the vicinity of a point which does not require $O(n)$ messages. On the other hand, it may be possible to reduce the factor $\log^2(1/\varepsilon)$ to just $\log(1/\varepsilon)$ although we do not know how to accomplish this. A related open problem concerns the $O(\log n)$ gap between Propositions 5.2 and 2.3, for the class $\mathcal{F}_{\mathrm{SC},M,L}$.

3. Some directions along which it is likely that the results can be extended concern the case of $K > 2$ processors and the case where the constraints under which the optimization is carried out are not commonly known: for example, we may have a constraint of the form $g_1(x) + g_2(x) \leq 0$, where each g_i is a convex function known by processor P_i.

REFERENCES

ABELSON, H. (1980), Lower bounds on information transfer in distributed computations, *J. Assoc. Comput. Mach.* **27,** No. 2, 384–392.

AHO, A. V., ULLMAN, J. D., AND YANNAKAKIS, M. (1983), On notions of information transfer in VLSI circuits, *in* "Proceedings, 15th STOC, 1983," pp. 133–139.

AWERBUCH, B., AND GALLAGER, R. G. (1985), "Communication Complexity of Distributed Shortest Path Algorithms," Technical Report LIDS-P-1473, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

MEHLHORN, K., AND SCHMIDT, E. M. (1982), Las Vegas is better than Determinism in VLSI and distributed computing, *in* "Proceedings, 14th STOC, 1982," pp. 330–337.

NEMIROVSKY, A. S., AND YUDIN, D. B. (1983), "Problem Complexity and Method Efficiency in Optimization," Wiley, New York.

PANG, K. F., AND EL GAMAL, A. (1986), Communication complexity of computing the Hamming distance, *SIAM J. Comput.* **15,** No. 4, 932–947.

PAPADIMITRIOU, C. H., AND SIPSER, M. (1982), Communication complexity, *in* "Proceedings, 14th STOC, 1982," pp. 196–200.

PAPADIMITRIOU, C. H., AND STEIGLITZ, K. (1982), "Combinatorial Optimization: Algorithms and Complexity," Prentice-Hall, Englewood Cliffs, NJ.

PAPADIMITRIOU, C. H., AND TSITSIKLIS, J. N. (1982), On the complexity of designing distributed protocols, *Inform. and Control* **53,** No. 3, 211–218.

TRAUB, J. F., AND WOŹNIAKOWSKI, H. (1980), "A General Theory of Optimal Algorithms," Academic Press, Orlando, FL.

ULLMAN, J. D. (1984), "Computational Aspects of VLSI," Comput. Sci. Press, Rockville, MD.

YAO, A. C. (1979), Some complexity questions related to distributed computing, *in* "Proceedings, 11th STOC, 1979," pp. 209–213.