DIMACS Technical Report 94-16
April 15, 1994

A SUBLINEAR-TIME RANDOMIZED APPROXIMATION ALGORITHM FOR MATRIX GAMES*

by

Michael D. Grigoriadis[1] and Leonid G. Khachiyan[1]
Department of Computer Science
Rutgers University
New Brunswick, NJ 08903

* Research supported by the National Science Foundation under grant CCR-9208539.
[1] Permanent member of DIMACS.

DIMACS is a cooperative project of Rutgers University, Princeton University, AT&T Bell Laboratories, and Bellcore.

DIMACS is an NSF Science and Technology Center funded under Contract STC-88-09648; and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

This paper presents a parallel randomized algorithm which computes a pair of ε-optimal strategies for a given (m,n)-matrix game $A = [a_{ij}] \in [-1,1]$ in $O(\varepsilon^{-2}\log^2(n+m))$ expected time on an $(n+m)/\log(n+m)$-processor EREW PRAM. This algorithm is a natural extension of the classical method of fictitious play by Brown and Robinson. For any fixed accuracy $\varepsilon > 0$, the expected sequential running time of the suggested algorithm is $O((n+m)\log(n+m))$, which is sublinear in mn, the number of input elements of A. On the other hand, simple arguments are given to show that for $\varepsilon < 1/2$, any sequential deterministic algorithm for computing a pair of ε-optimal strategies of an (m,n)-matrix game A with ± 1 elements examines $\Omega(mn)$ of its elements. In particular, for $m = n$ the randomized algorithm achieves an almost quadratic expected speedup relative to any deterministic method.

Key words. approximation algorithms, complexity, linear programming, matrix games, parallel algorithms, randomized algorithms

1. Introduction. Let $A = [a_{ij}]$ be an (m,n)-*matrix game* of *value*

$$\begin{aligned} v^* &= \min\{\ v \mid Ax \le ve,\ e^T x = 1,\ x \ge 0\ \} \\ &= \max\{\ v \mid A^T y \ge ve,\ e^T y = 1,\ y \ge 0\ \}, \end{aligned} \tag{1.1}$$

which is always achieved for a pair of *optimal strategies* $x^* \in \mathbb{R}^n$ and $y^* \in \mathbb{R}^m$. We assume that the elements of the payoff matrix A lie in some fixed interval, say, in $[-1,1]$. This normalization assumption entails no loss of generality.

A pair of strategies x, y is ε-*optimal* for A if

$$\begin{aligned} &Ax \le (v^* + \varepsilon)e,\ e^T x = 1,\ x \ge 0, \\ &A^T y \ge (v^* - \varepsilon)e,\ e^T y = 1,\ y \ge 0, \end{aligned} \tag{1.2}$$

where e is the vector of all ones. In [3] we proposed a deterministic approximation algorithm to compute such an ε-optimal solution in $O(\varepsilon^{-2}\log^2(n+m))$ time on an mn-processor EREW PRAM. That algorithm is an extension of the exponential-potential reduction method for convex block-angular optimization described in [4].

In this paper we present a different approximation algorithm for matrix games, which is a randomized variant of the classical iterative method, called *fictitious play*, by Brown [1] and Robinson [6]. Specifically, we show that a pair of ε-optimal strategies for a given game $A \in [-1,1]^{mn}$ can be computed in $O(\varepsilon^{-2}\log^2(n+m))$ expected time on an EREW PRAM with only $(n+m)/\log(n+m)$ processors. For any fixed $\varepsilon > 0$, the sequential variant of our algorithm runs in $O((n+m)\log(n+m))$ expected time, which is sublinear in the number of elements of A if, say, m and n are of the same order. We also show that for $\varepsilon < 1/2$, any deterministic algorithm for computing a pair of ε-optimal strategies of an (m,n)-matrix game with ± 1 elements must examine almost half of the mn elements of A.

Brown's original algorithm [1] is as follows:

$$\begin{aligned} &X(0) = 0 \in \mathbb{R}^n,\ \ Y(0) = 0 \in \mathbb{R}^m.\ \text{For}\ \ t = 0,1,\ldots,\ \text{repeat:} \\ &X(t+1) = X(t) + e_{i(t)},\ \ e_{i(t)} = \operatorname{argmin}\{\, Y(t)Ae_i \ \mid i = 1,\ldots,n\ \}, \\ &Y(t+1) = Y(t) + e_{j(t)},\ \ e_{j(t)} = \operatorname{argmax}\{e_j^T AX(t) \mid j = 1,\ldots,m\}, \end{aligned} \tag{1.3}$$

where e_i and e_j denote the ith and jth coordinate vectors in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively.

Its convergence for an arbitrary matrix game A,

$$
\begin{aligned}
&A\frac{X(t)}{t} \le \big(v^* + \varepsilon(t)\big)e, \quad e^T\frac{X(t)}{t} = 1, \quad \frac{X(t)}{t} \ge 0\\
&A^T\frac{Y(t)}{t} \ge \big(v^* - \varepsilon(t)\big)e, \quad e^T\frac{Y(t)}{t} = 1, \quad \frac{Y(t)}{t} \ge 0,\\
&\varepsilon(t) \to 0 \quad \text{for} \quad t \to \infty,
\end{aligned}
$$

was shown by Robinson [6]. The only known upper bound on the convergence rate of this method is $\varepsilon(t) \le 2^{n+m}\, t^{-1/(n+m-2)}$, due to Shapiro [7]. A lower bound on the convergence rate of a wider class of methods was given by Khachiyan [5]. It has long been suspected that Shapiro's bound is much too pessimistic and that the number of fictitious plays (iterations) actually grows quadratically in $1/\varepsilon$.

In the tth iteration of the Brown-Robinson method, the minimizing player computes the index $i(t)$ of the smallest component of the vector $V(t) = Y(t)A \in \mathbb{R}^n$ (see (1.3), second line) and then increases the $i(t)$th component of $X(t)$ by one. That is, the minimizing player makes a move in the direction of the pure strategy $e_{i(t)} \in \mathbb{R}^n$ that minimizes his pay-off against the maximizing player's current strategy $Y(t)$. At the same time, the maximizing player selects the pure strategy $e_{j(t)} \in \mathbb{R}^m$ in order to maximize his immediate reward $e_j^T U(t)$, where $U(t) = AX(t) \in \mathbb{R}^m$. For symmetric games ($A = -A^T$) the iterative procedure (1.3) produces $X(t) = Y(t)$ and $V(t) = -U(t)$ for all t.

Our algorithm is a simple randomized variant of this deterministic method in which indices $i(t) \in \{1, \ldots, n\}$ or $j(t) \in \{1, \ldots, m\}$ are selected with probability proportional to $\exp\{-\varepsilon V_i(t)/2\}$ or $\exp\{+\varepsilon U_j(t)/2\}$ for the minimizing and maximizing player, respectively. For symmetric games, the probability of selecting an index $i(t)$ is given by the Gibbs distribution $p_i(t) = \exp\{\varepsilon U_i(t)/2\} \;/\; \sum_{j=1}^n \exp\{\varepsilon U_j(t)/2\}$. This distribution tends to concentrate around the pure strategies the Brown-Robinson method would select and avoids its discontinuous selection rule.

In Sections 2 and 3 we shall analyze the performance of this randomized algorithm. In the last section we shall present a simple information-theoretic lower bound for deterministic approximation algorithms for matrix games.

2. Parallel randomized algorithm for symmetric games. We first consider *symmetric* matrix games for which $A = -A^T$. In this case $m = n$, the value of the game is $v^* = 0$, the sets of optimal strategies of both players coincide, and (1.2) simplifies to the problem of computing an x such that

$$
Ax \le \varepsilon e, \quad x \in S = \{x \in \mathbb{R}^n \mid e^T x = 1, \;\; x \ge 0\}. \tag{2.1}
$$

The algorithm below finds an ε-optimal strategy x for A in time $O(\varepsilon^{-2}\log^2 n)$ on an $n/\log n$-processor EREW PRAM with probability at least $1/2$. We shall assume without loss of generality that $\varepsilon \in (0,1]$ and $n \geq 8$.

ALGORITHM $\mathcal{A}$:

[1] *Initialize:* $X \leftarrow U \leftarrow 0 \in \mathbb{R}^n$, $p \leftarrow e/n$, $t \leftarrow 0$.

[2] Repeat:

[3] *Iteration count:* $t \leftarrow t+1$.

[4] *Random direction:* pick a random $k \in \{1,\ldots,n\}$ with probability p_k.

[5] *X-update:* $X_k \leftarrow X_k + 1$.

[6] *U-update:* For $i = 1,\ldots,n$ do in parallel $U_i \leftarrow U_i + a_{ik}$.

[7] *p-update:* For $i = 1,\ldots,n$ do in parallel

$$p_i \leftarrow p_i \exp\left\{\frac{\varepsilon}{2}a_{ik}\right\} \Bigg/ \sum_{j=1}^{n} p_j \exp\left\{\frac{\varepsilon}{2}a_{jk}\right\}.$$

[8] *Stopping criterion:* If $U/t \leq \varepsilon e$, report $x = X/t$ and halt. □

THEOREM 1.

i) Algorithm $\mathcal{A}$ is correct: the stopping criterion in line 8 guarantees the ε-optimality of the output vector x.

ii) With probability $\geq 1/2$ Algorithm $\mathcal{A}$ halts in $t^* = 4\varepsilon^{-2}\ln n$ iterations.

Proof. Lines 5 and 6 of Algorithm $\mathcal{A}$ maintain the invariants $X(t)/t \in S = \{x \in \mathbb{R}^n \mid e^T x = 1,\ x \geq 0\}$ and $U(t) = AX(t)$, where $X(t)$, $U(t)$ denote the the n-vectors X, U at the end of iteration $t \geq 1$. This immediately implies part (i) of the theorem.

It is also easy to see from lines 6 and 7 of Algorithm $\mathcal{A}$ that

$$p_i(t) = P_i(t) \Bigg/ \sum_{j=1}^{n} P_j(t)\ , \qquad \text{for } t = 1,2,\ldots,$$

where

$$P_i(t) = \exp\{\varepsilon U_i(t)/2\}, \qquad i = 1,\ldots,n. \tag{2.2}$$

Now, define the potential function

$$\Phi(t) = \sum_{i=1}^{n} P_i(t). \tag{2.3}$$

If the algorithm selects an index $k \in \{1, \ldots, n\}$ in line 4 at iteration t, then

$$\begin{aligned}\Phi(t+1) &= \sum_{i=1}^{n} P_i(t+1) = \sum_{i=1}^{n} \exp\{\varepsilon U_i(t+1)/2\} \\ &= \sum_{i=1}^{n} \exp\{\varepsilon U_i(t)/2 + \varepsilon\, a_{ik}/2\} = \sum_{i=1}^{n} P_i(t) \exp\{\varepsilon\, a_{ik}/2\} \\ &= \Phi(t) \sum_{i=1}^{n} p_i(t) \exp\{\varepsilon\, a_{ik}/2\}.\end{aligned}$$

And since such a k is selected with probability $p_k(t)$, we conclude that the expected value of $\Phi(t+1)$ is given by

$$E[\Phi(t+1)] = \Phi(t) \sum_{i,k=1}^{n} p_i(t)\, p_k(t) \exp\{\varepsilon\, a_{ik}/2\}. \tag{2.4}$$

By our assumption that $a_{ik} \in [-1, 1]$, we have

$$\exp\{\varepsilon\, a_{ik}/2\} \leq 1 + \frac{\varepsilon}{2} a_{ik} + \frac{\varepsilon^2}{6}$$

for all $\varepsilon \in (0, 1]$. Since A is skew-symmetric,

$$\sum_{i,k=1}^{n} p_i(t)\, p_k(t)\, a_{ik} = 0.$$

We also have that

$$\sum_{i,k=1}^{n} p_i(t)\, p_k(t) = \Big(\sum_{i=1}^{n} p_i(t)\Big)^2 = 1$$

for any $p(t) \in S$. Hence,

$$E[\Phi(t+1)] \leq E[\Phi(t)]\,\big(1 + \varepsilon^2/6\big),$$

which readily implies

$$E[\Phi(t)] \leq \Phi(0)\big(1 + \varepsilon^2/6\big)^t = n\big(1 + \varepsilon^2/6\big)^t \leq n \exp\{t\varepsilon^2/6\}.$$

Specifically, after $t^* = 4\varepsilon^{-2} \ln n$ iterations we have $E[\Phi(t^*)] \leq n^{5/3}$. By the Markov inequality, the latter implies that $\Phi(t^*) \leq 2n^{5/3}$ with probability at least $1/2$. And since $n \geq 8$ by assumption,

$$\Phi(t^*) \leq n^2 \quad \text{with probability } \geq 1/2.$$

On the other hand, if $\Phi(t^*) \le n^2$, then by the definition (2.3) of $\Phi(t)$ we have $P_i(t^*) \le n^2$ for all i. By (2.2) this is equivalent to

$$\frac{\varepsilon}{2} U_i(t^*) \le 2 \ln n, \quad i = 1, \ldots, n,$$

or equivalent to

$$\frac{U(t^*)}{t^*} \le \frac{4 \ln n}{t^* \varepsilon}\, e = \varepsilon\, e,$$

which means that the stopping criterion (line 8) is satisfied at the t^*th iteration of Algorithm $\mathcal{A}$ with probability at least $1/2$. □

COROLLARY 1. *An ε-optimal strategy for a symmetric game $A \in [-1,1]^{n^2}$ can be computed in $O(\varepsilon^{-2} \log n)$ expected iterations of Algorithm $\mathcal{A}$.*

Proof. Run Algorithm $\mathcal{A}$ repeatedly, for at most $t^* = 4\varepsilon^{-2} \ln n$ iterations each time, until the stopping criterion is satisfied. □

Each iteration of Algorithm $\mathcal{A}$ can be implemented to run in $O(\log n)$ time on an $n/\log n$-processor EREW PRAM. In particular, line 4 of the algorithm can be implemented by performing a parallel prefix computation to evaluate $S_i = p_1 + \ldots + p_i$, $i = 0, 1, \ldots, n$, followed by the computation of $k \in \{1, \ldots, n\}$ such that $S_{k-1} < \xi \le S_k$ for a randomly and uniformly generated number $\xi \in [0,1]$. This fact, along with Corollary 1 justifies the following result.

THEOREM 2. *An ε-optimal solution of a given symmetric $n \times n$ matrix game $A = [a_{ij}] \in [-1,1]$ can be computed in $O(\varepsilon^{-2} \log^2 n)$ expected time on an $n/\log n$-processor* EREW PRAM.

3. The general case. We shall now extend the complexity bounds of the previous section to the case of general nonsymmetric matrix games by using the following standard reduction (see e.g. Dantzig [2].)

Given an (m,n)-matrix game A, consider the $(m+n+1)$-order symmetric game

$$B = \begin{pmatrix} 0 & A & -e \\ -A^T & 0 & e \\ e^T & -e^T & 0 \end{pmatrix}, \tag{3.1}$$

whose value is zero. Suppose that an ε-optimal solution $z = (y, x, u)$ is known for B, where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, and $u \in \mathbb{R}^1$. By the definition (2.1) of the ε-optimality for symmetric games, such a solution of B must satisfy the inequalities:

$$Ax \le (u + \varepsilon)e \tag{3.2}$$

$$A^T y \geq (u - \varepsilon)e \tag{3.3}$$

$$e^T y \leq e^T x + \varepsilon \tag{3.4}$$

$$e^T x + e^T y + u = 1, \quad x, y, u \geq 0. \tag{3.5}$$

The following proposition shows that a pair of $O(\varepsilon)$-optimal strategies for A can easily be derived from an ε-optimal solution for B [3]. For its proof, which we give here for the sake of completeness, we make the technical assumption that the elements of A lie in the interval $[1/2, 1]$ rather than in $[-1, 1]$. If this is not the case, we can always replace ε by $\varepsilon/4$, and replace the original matrix $A \in [-1, 1]^{mn}$ by $A/4 + 3E/4$, where E is the (m, n)-matrix of all ones, before constructing B by (3.1).

PROPOSITION 1. *For sufficiently small ε (say, $\varepsilon < 0.1$), the inequalities (3.2)–(3.5) imply that $e^T x$ and $e^T y$ are strictly positive, and that*

$$\bar{x} = x/e^T x, \quad \bar{y} = y/e^T y \tag{3.6}$$

is a pair of $O(\varepsilon)$-optimal strategies for A.

Proof. Let v^* be the value of A as defined by (1.2). Then, by (3.2)–(3.3),

$$v^* e^T x \leq u + \varepsilon, \tag{3.7}$$

$$u - \varepsilon \leq v^* e^T y, \tag{3.8}$$

which give $v^* e^T x \leq v^* e^T y + 2\varepsilon$. Since $v^* \geq 1/2$ from the assumption $a_{ij} \in [1/2, 1]$, the latter inequality implies that

$$e^T x \leq e^T y + 4\varepsilon, \tag{3.9}$$

and since $v^* \leq 1$, (3.8) implies

$$u \leq e^T y + \varepsilon. \tag{3.10}$$

Summing (3.9), (3.10), and the trivial inequality $e^T y \leq e^T y$, equation (3.5) yields:

$$e^T y \geq (1 - 5\varepsilon)/3. \tag{3.11}$$

But by (3.4) and (3.11),

$$e^T x \geq (1 - 8\varepsilon)/3. \tag{3.12}$$

Thus, for a sufficiently small ε, both $e^T x$ and $e^T y$ are strictly positive.

To show the second part of the claim, first consider:

$$\begin{aligned}\max\{\, A_1\bar{x}, \cdots, A_m\bar{x}\,\} &\le \frac{u+\varepsilon}{e^T x} = \frac{u-\varepsilon}{e^T x} + \frac{2\varepsilon}{e^T x} \\ &\le v^* \frac{e^T y}{e^T x} + \frac{2\varepsilon}{e^T x} \\ &\le v^* \left(1 + \frac{\varepsilon}{e^T x}\right) + \frac{2\varepsilon}{e^T x} = v^* + (2+v^*)\,\frac{\varepsilon}{e^T x} \\ &\le v^* + \frac{3\varepsilon}{e^T x} \le v^* + \frac{9\varepsilon}{1-8\varepsilon} = v^* + O(\varepsilon),\end{aligned}$$

where A_i denotes the ith row of A. The above inequalities are obtained from (3.2), (3.8), (3.4), the fact that $v^* \le 1$, and (3.12), respectively. An analogous argument, which uses (3.3), (3.7), (3.4), $v^* \le 1$, and (3.11), yields:

$$\min\{\, \bar{y}^T A^1, \cdots, \bar{y}^T A^n\,\} \ge v^* - \frac{9\varepsilon}{1-5\varepsilon} = v^* - O(\varepsilon),$$

where A^j denotes the jth column of A. □

Theorem 2 and Proposition 1 give the main result of our paper:

THEOREM 3. *There is a parallel randomized algorithm that computes a pair of ε-optimal strategies for an (m,n)-matrix game $A \in [-1,1]^{mn}$ in $O(\varepsilon^{-2}\log^2(m+n))$ expected time on an $(m+n)/\log(m+n)$-processor* EREW PRAM.

Even though we implicitly stated Theorems 2 and 3 in the unit-cost model of computation, it is easy to show that the accuracy required by Algorithm $\mathcal{A}$ can be bounded by $O(\log(m+n))$ bits for any fixed ε.

We also mention in passing that for a fixed ε, any matrix game $A \in [-1,1]^{mn}$ has a pair of ε-optimal strategies that uses only $O((m+n)\log(m+n))$ pure strategies. This follows from the iteration bound of Algorithm $\mathcal{A}$ above, but also follows from that of the deterministic algorithm previously reported in [3].

4. A lower bound for deterministic methods. Since Algorithm $\mathcal{A}$ runs in $O(\varepsilon^{-2}\log(m+n))$ expected iterations, inspecting at most one column or row of A at each iteration, the expected number of elements of A it examines can be bounded by $N = O(\varepsilon^{-2}(m+n)\ \log(m+n))$. If ε is fixed, N is almost linear in $m+n$. This should be compared to the following fact.

PROPOSITION 2. *Any deterministic method for computing the value of a matrix game $A \in \{-1,1\}^{mn}$ to an absolute accuracy less than $1/2$ must examine at least $N = \lfloor m/2 \rfloor n \approx mn/2$ elements of A.*

Note that, given a pair of ε-optimal strategies, the value of A can be readily computed to within the same accuracy ε. Proposition 2 thus implies that unlike the randomized Algorithm $\mathcal{A}$, any deterministic algorithm for computing a pair of ε-optimal strategies with $\varepsilon < 1/2$ must examine about $mn/2$ entries of A. Therefore, such a deterministic algorithm must require $\Omega(mn)$ sequential time or $\Omega\left((mn/(m+n))\log(m+n)\right)$ parallel time on an $(n+m)/\log(m+n)$-processor PRAM.

Proof of Proposition 2. Let m be even for simplicity and consider an arbitrary deterministic algorithm $\mathcal{D}$ for approximating the value of A. Suppose that elements of the payoff matrix are provided to $\mathcal{D}$ by the following rule. Each time $\mathcal{D}$ queries an element, say, the (k,ℓ)th element of the payoff matrix, it is given the value $a_{k\ell} = -1$, unless the number of -1's it has already received in the ℓth column is $m/2$, in which case it is given $a_{k\ell} = +1$.

Suppose that $\mathcal{D}$ examines at most $mn/2$ entries of the payoff matrix. Let A denote the resulting partially defined (m,n) payoff matrix and consider the following two extensions of A to fully-defined payoff matrices.

1. Define the remaining (unexamined) elements of A so that the resulting matrix A' contains an equal number $m/2$ of $+1$'s and -1's in each column. This is clearly possible by the above rule. By (1.1) we have $A'x^* \leq v'e$, where x^* is the optimal strategy for the minimizing player and $v' = v(A')$ is the value of A'. Summing these inequalities, we see that $v' \geq 0$.
2. Define A'' to be the extension of A obtained by setting all of its unexamined elements to -1. Since $\mathcal{D}$ examines at most $mn/2$ entries by our assumption, A'' must contain an entire column of -1's. This implies that $v'' = v(A'') = -1$.

Since $v' = v(A') \geq 0$, $v'' = v(A'') = -1$, and A', A'' coincide with A on all examined elements, $\mathcal{D}$ cannot determine the approximate value of A with accuracy $\varepsilon < 1/2$ unless it examines more than $mn/2$ elements. □

A straightforward strengthening of Proposition 2 is that any deterministic algorithm must examine at least $\lfloor m(1-\varepsilon)\rfloor n \approx mn(1-\varepsilon)$ entries in order to compute the value of an (m,n)-matrix game $A \in \{-1,1\}^{mn}$ to within an absolute accuracy of $0 < \varepsilon < 1/2$.

5. References.

[1] G.W. Brown, "Iterative solution of games by fictitious play," in *Activity Analysis of Production and Allocation*, T.C. Koopmans, Ed., Cowles Commission Monograph 13, John Wiley, New York (1951).

[2] G.B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ (1953).

[3] M.D. Grigoriadis and L.G. Khachiyan, "Approximate solution of matrix games in parallel", in *Advances in Optimization and Parallel Computing*, pp. 129-136, P.M. Pardalos, Ed., Elsevier Science Publishers B.V. (1992). Also available as TR-91-73 from DIMACS, CoRE-Busch, Rutgers Univ., Nerw Brunswick, NJ 08903.

[4] M.D. Grigoriadis and L.G. Khachiyan, "Fast approximation schemes for convex programs with many blocks and coupling constraints", *SIAM J. Optimization* 4 (1994), pp. 86-107.

[5] L.G. Khachiyan, "Convergence rate of the game processes for solving matrix games", *Zh. Vȳchisl. Mat. & Mat. Fiz.*, 17-6 (1977) pp. 1421–1431. English translation in *U.S.S.R. Comput. Math. & Math. Phys.*, 17 (1978), pp. 78–88.

[6] J. Robinson, "An iterative method of solving a game," *Annals of Math.* 54 (1951), pp. 296-301.

[7] H.N. Shapiro, '"Note on a computation method in the theory of games", *Comm. Pure and Applied Math.* 11 (1958), pp. 587-593.

April 15, 1994.