DATA PARALELIZATION FOR SOLVING BIMATRIX GAMES

HÂNCU Boris
CATARANCIUC Emil
Republic of Moldova, Chișinău
ecataranciuc@gmail.com
boris.hancu@gmail.com

Articolul prezintă un studiu teoretic şi practic al modalităţilor de determinare a soluţiilor în jocurile bimatriceale împărţite în blocuri de submatrici utilizând algoritmul 2D-ciclic de divizare şi distribuire a matricelor. Teoremele demonstrate reprezintă baza pe care soluţia jocului bimatriceal poate fi construită folosind soluţiile subjocurilor generate de algoritmul de distribuire a matricei ciclice 2D.

***Keywords and phrases:**: non cooperative game, bimatrix subgames, Nash equilibrium, parallel algorithms, distributed memory clusters, 2D block-cyclic partitioned algorithm*

It is known that the basic parallel strategy consists of three main steps: the **first step** is to partition the input into several partitions of almost equal sizes, and so the parallelization at the level of data and operations is actually achieved; the **second step** is to solve recursively the subproblem defined by each partition of the input, and so, the distribution and solution of subproblems on a parallel computing system is actually performed; the **third step** is to combine or merge the solutions of the different subproblems into a solution for the overall problem. The success of such a strategy depends on whether or not we can perform the fist and third steps efficiently [1].

In this presentation we will analyze the ways to build parallel algorithms, and especialy, data parallelization, for a class of noncooperative games, bimamtrix games.

Matrix games. Algorithms for determining equilibrium profiles in pure strategies. We consider the bimatrix game in the following strategic form $\Gamma = \langle I, J, A, B \rangle$, where $I = \{1,2,..,n\}$ is the line index set (the set of strategies of the player 1), $J = \{1,2,..,m\}$ is the column index set (the set of strategies of the player 2) and $A = \| a_{ij} \|_{i\in I}^{j\in J}$, $B = \| b_{ij} \|_{i\in I}^{j\in J}$ are the payoff matrices of player 1 and player 2, respectively. All players know exactly the payoff matrices and the sets of strategies. So, the game is incomplete and has imperfect information. Players intent to maximize their payoffs. The matrices A and B are called *global matrices*. We denote by $NE[\Gamma]$ the set of all equilibrium profiles in the game Γ..Thus, Nash equilibrium profile is the pair of indices (i^*, j^*), for which the following system of inequalities is verified $(i^*, j^*) \in NE[\Gamma] \Leftrightarrow \begin{cases} a_{i^* j^*} \geq a_{ij^*} & \forall i \in I, \\ b_{i^* j^*} \geq b_{i^* j} & \forall j \in J. \end{cases}$

The definition of Nash equilibrium profiles "in the language of inequalities" is difficult to apply in practice. Therefore, we will present the following equivalent definitions, which are already "in the language of optimization problems" and "in the language of point-to-set applications."It is obvious that the inequality system is equivalent to the following equality system: $(i^*, j^*) \in NE[\Gamma] \Leftrightarrow \begin{cases} a_{i^* j^*} = \max\limits_{i\in I} a_{ij^*}, \\ b_{i^* j^*} = \max\limits_{j\in J} b_{i^* j}. \end{cases}$ We will present below a equivalent definition but "in the language of best-response type point-to-set applications." We will build the following point-to-set aplication $Br_1 : J \to 2^I$ such that for any fixed column $j \in J$, $Br_1(j) = Arg \max\limits_{i\in I} a_{ij}$ is determined. Similarly $Br_2 : I \to 2^J$ such that for any fixed column $i \in I$, $Br_2(i) = Arg \max\limits_{j\in J} b_{ij}$ is determined. Denoted by $GrBr_1 = \{(i, j) : i \in Br_1(j), \forall j \in J\} \equiv \left\{(i, j) : i = \arg\max\limits_{i\in I} a_{ij}, \forall j \in J\right\}$ and $GrBr_2 = \{(i, j) : j \in Br_2(i), \forall i \in I\} \equiv \left\{(i, j) : j = \arg\max\limits_{j\in J} b_{ij}, \forall i \in I\right\}$ the graphs of Br_1 and Br_2. Then we obtain the following definition of Nash equilibrium profiles: $\left(i^{*}, j^{*}\right) \in NE[\Gamma] \Leftrightarrow \left(i^{*}, j^{*}\right) \in GrBr_1 \bigcap GrBr_2$. These definition from a practical point of view it is easier to apply and so the following algorithm can be formulated to determine the Nash equilibrium profiles in pure strategies.

Algorithm 1

1. For any fixed column $j \in J$, the set $Br_1(j) = Arg \max_{i \in I} a_{ij}$ is determined. Under algorithmic aspect it can be as follows: for any column j of the matrix A all maximum elements of this column are highlighted.
2. For any fixed row $i \in I$, the set $Br_2(i) = Arg \max_{j \in J} b_{ij}$ is determined. Under algorithmic aspect it can be as follows: for any row i of the matrix B all maximum elements on this row are highlighted.
3. The graph $GrBr_1$ of the multilevel application Br_1 from step 1) and also the graph $GrBr_2$ of the multilevel application Br_2 from step 2) are built. The equilibrium profiles are all the profiles belonging to the intersection of the two given graphs: $NE = GrBr_1 \cap GrBr_2$. From an algorithmic point of view it can be done as follows: we look for all highlighted elements in the matrices A and B and the indices of the elements whose positions coincide both in matrix A and in matrix B will be the equilibrium profiles.

Generation of sets of bimatrix subgames as a result of the division into blocks of global matrices. Suppose that matrices A and B are divided into submatrices (at the moment we do not analyze the case of distribution of submatrices). We can consider the following cases: a) when both matrix A and matrix B are divided into submatrices using one and the same algorithm, b) when both matrix A and matrix B are divided into submatrices using different algorithms. Thus we can obtain a series of pairs of submatrices of the same size $\left\{\left(A_r, B_r\right)\right\}_{r=\overline{1,p}}$ where $A_r = \left\| a^r_{i_r j_r} \right\|^{j_r \in J_r}_{i_r \in I_r}$ and $B_r = \left\| b^r_{i_r j_r} \right\|^{j_r \in J_r}_{i_r \in I_r}$. We mention that we will analyze only the case when the submatrices, obtained as a result of the algorithms for dividing the matrices *A* and *B*, will have the same dimension, and therefore bimatrix subgames can be constructed. Here the index *r* actually means "processor" which, as will be seen below, will obtain these submatrices. These submatrices will generate a series of games which are actually sub-games of the original game $\Gamma_r = \langle I_r, J_r, A_r, B_r \rangle$. We denote by $NE[\Gamma_r]$ the set of Nash equilibrium profiles in the problem Γ_r.

We introduce the following applications which determine the correspondence between the "local indices" of the elements of the local matrices A_r, B_r and the "global indices" of the elements of the global matrices A and B, specifically $\varphi_r : I_r \to I$, $\psi_r : J_r \to J$. These functions are in fact determined by the algorithm of dividing the matrices A and B into submatrices. Obviously, these functions must verify the following conditions:

$$\forall i \in I, \exists r \textit{ and } i_r \in I_r \text{ that } i = \varphi_r(i_r), \qquad (1)$$

$$\forall j \in J, \exists\, r \textit{ and } j_r \in J_r \text{ that } j = \psi_r(j_r). \qquad (2)$$

According to (1)-(2) the following condition is verified: for any strategy profile in the bimatrix games Γ, namely $(i, j) \in I \times J$, there exist a strategy profile (i_r, j_r) in the bimatrix games Γ_r, so that $i = \varphi_r(i_r)$ and $j = \psi_r(j_r)$. As a result we analyze only those divisions and distributions of the global matrices in local matrices for which there exist the applications φ_r and ψ_r such that the conditions (1)-(2) are satisfied.

We formulate the following problems.

Problem 1. (**existence of solutions**) What are the conditions (necessary, sufficient, necessary and sufficient) for the equilibrium profiles in the game Γ_r. to be the equilibrium situation in the game Γ. Here we can highlight the following two aspects:

1. which $(i_r^*, j_r^*) \in NE[\Gamma_r]$ there will also be $(\varphi_r(i_r^*), \psi_r(j_r^*)) \in NE[\Gamma]$;

2. what properties must the algorithm of dividing the matrices into submatrices possess in order to if $(i_r^*, j_r^*) \in NE[\Gamma_r]$ then $(\varphi_r(i_r^*), \psi_r(j_r^*)) \in NE[\Gamma]$.

About an matrix division algorithm for which any solution of the subgames is also the solution of the initial game. We will first analyze those algorithms for constructing subgames based on the division of matrices for which the solutions of the subgames will be the solutions of the initial game. We will introduce the following definition.

Definition 1 *If for any equilibrium profile* $\left(i_r^*, j_r^*\right) \in NE[\Gamma_r]$ *in the game* Γ_r *exists* $\varphi_r : I_r \to I$, $\psi_r : J_r \to J$ *that is* $\left(\varphi_r(i_r^*), \psi_r(j_r^*)\right) \in NE[\Gamma]$ *then we will say that the algorithm of dividing the matrices into blocks of submatrices is perfect and will be called "**p**erfect **m**atrix **d**ividing and **d**istribution **a**lgorithm" (PMDDA)*

To realise the data parallelisation, we use the **two-dimensional block-cyclic data layout scheme** [2]. The P processes of an abstract parallel computer are often represented as a one-dimensional linear array of processes labelled $0,1,\ldots,P$. It is often more convenient to map this one-dimensional array of processes into a two-dimensional rectangular grid, or process grid by using row-major order (the numbering of the processes increases sequentially across each row) or by using column-major order (the numbering of the processes proceeds down each column of the process grid). This grid will have $l_{\max}$ process rows (line) and $c_{\max}$ process columns, where $l_{\max} + c_{\max} = P$. The process can now be referenced by its row and column coordinates, (l,c), within the grid $L \times C$ where $L = \{1,..,l,...,l_{\max}\}$ is a set of row numbers and $C = \{1,..,c,...,c_{\max}\}$ is a set of column numbers. These groupings of processes are of particular interest to the programmer, since distributed data decomposition of a matrix tends to follow this process mapping. Viewing the rows/columns of the process grid as essentially autonomous subsystems provides the programmer with additional levels of parallelism.

For dense matrix computations we assume the data to be distributed according to the two-dimensional block-cyclic data layout scheme. The block-cyclic data layout has been selected for the dense algorithms implemented in DMM parallel systems principally because of its **scalability, load balance** and **efficient** use of computation routines (data locality). The block-partitioned computations are processed in consecutive order just like a conventional serial algorithm.

According to the **2D** **b**lock-**c**yclic **m**atrix **d**ividing and **d**istribution (**2DBCMD&D**), an m by n dense matrix is first decomposed into m_A by n_A blocks starting at its upper left corner. These blocks are then uniformly distributed in each dimension of the Process Grid. Thus, every process owns a collection of blocks, which are locally and contiguously stored in a two-dimensional column major array. We present below some examples of the distributed matrix based on **2DBCMD&D** algorithm. If we have a global 6×5 matrix $A=\left\| a_{ij} \right\|_{i=\overline{1,6}}^{j=\overline{1,5}}$ then for 2-D process grid $L\times C = 2\times 2$ and block dimension 2×2 we obtain the division and distribution on the process grid represented below. Here $A_{(l,c)}$ represents the submatrix which will be distributed later to the process (l,c).

$$A_{(0,0)} = \begin{pmatrix} a_{11} & a_{12} & a_{15} \\ a_{21} & a_{22} & a_{25} \\ a_{51} & a_{52} & a_{55} \\ a_{61} & a_{62} & a_{65} \end{pmatrix}, A_{(0,1)} = \begin{pmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \\ a_{53} & a_{54} \\ a_{63} & a_{64} \end{pmatrix},$$

$$A_{(1,0)} = \begin{pmatrix} a_{31} & a_{32} & a_{35} \\ a_{41} & a_{42} & a_{45} \end{pmatrix}, A_{(1,1)} = \begin{pmatrix} a_{33} & a_{34} \\ a_{43} & a_{44} \end{pmatrix}.$$

Let $I_{(l,c)}$ (respectively $J_{(l,c)}$) denotes the rows (respectively columns) of the local matrices. Denote by $A_{(c,l)} = \left\| a_{i_{(l,c)} j_{(l,c)}} \right\|_{i_{(l,c)}=\overline{1,\left|I_{(l,c)}\right|}}^{j_{(l,c)}=\overline{1,\left|J_{(l,c)}\right|}}$ and $B_{(c,l)} = \left\| b_{i_{(l,c)} j_{(l,c)}} \right\|_{i_{(l,c)}=\overline{1,\left|I_{(l,c)}\right|}}^{j_{(l,c)}=\overline{1,\left|J_{(l,c)}\right|}}$ submatrices formed from global matrices A and B that are distributed to the $(c,l)\in L\times C$ process grid. We call $i_{(l,c)}$ and $j_{(l,c)}$ local indices of the local elements of the local matrices $A_{(c,l)}$, $B_{(c,l)}$. According (1)-(2) for any strategy profile $(i,j)\in I\times J$, there exist a process $(l,c)\in L\times C$ and the strategy profile $(i_{(l,c)}, j_{(l,c)})$ that $i=\varphi_{(l,c)}\left(i_{(l,c)}\right)$ and $j=\psi_{(l,c)}(j_{(l,c)})$.

It's easy to prove that for **2DBCMD&D** algorithm the $\varphi_{(l,c)}$ and $\psi_{(l,c)}$ functions, also verify the following conditions.

Proposition 1 *For the 2DBCMD&D algorithm the* $\varphi_{(l,c)} : I_{(l,c)} \to I$, $\psi_{(l,c)} : J_{(l,c)} \to J$ *functions verify the following conditions:*

a) *for all fixed* $l = \overline{1, l_{\max}}$, *we have* $\varphi_{(l,\hat{c})}\left(i_{(l,\hat{c})}\right) = \varphi_{(l,\tilde{c})}\left(i_{(l,\tilde{c})}\right)$ *for all* $\hat{c} = \overline{1, c_{\max}}$, $\tilde{c} = \overline{1, c_{\max}}$ $\hat{c} \neq \tilde{c}$ *and* $i_{(l,\hat{c})} = i_{(l,\tilde{c})}$.

b) *for all fixed* $c = \overline{1, c_{\max}}$, *we have* $\psi_{(\bar{l},c)}\left(j_{(\bar{l},c)}\right) = \psi_{(\tilde{l},c)}\left(j_{(\tilde{l},c)}\right)$ *for all* $\bar{l} = \overline{1, l_{\max}}, \tilde{l} = \overline{1, l_{\max}}$, $\bar{l} \neq \tilde{l}$ *and* $j_{(\bar{l},c)} = j_{(\tilde{l},c)}$.

Condition a) means the following: processes which are located on the same line of the process grid (for example line l) in its submatrices contain elements of the same line of the global matrix. Respectively, condition b) means the following: the processes which are located on the same column of the process grid (for example column c) in its submatrices, contain elements of the same column of the global matrix. This property is used to construct the equilibrium profiles in the bimatrix games.

Nash equilibrium profiles for bimatrix games with block-cyclic distributed matrices. We denote by $NE\lfloor\Gamma_{(c,l)}\rfloor$ or $NE\lfloor\left(A_{(l,c)}, B_{(l,c)}\right)\rfloor$ the set of all Nash equilibrium profiles of the bimatrix game (subgame) $\Gamma_{(c,l)} = \left\langle I_{(c,l)}, J_{(c,l)}, A_{(c,l)}, B_{(c,l)} \right\rangle$. Based on the above mentioned, and namely according to basic parallel strategies, we can proceed to distribution on a parallel computing system the subproblems, which in our case consist in: determining the sets $NE\lfloor\Gamma_{(c,l)}\rfloor$ for any calculation process $(l,c) \in L \times C$. Based on definition of the Nash equilibrium profiles, any process $(l,c) \in L \times C$ of a parallel computing system with the distributed memory, simultaneously and independently determines the equilibrium profiles,

$\left(i^*_{(l,c)}, j^*_{(l,c)}\right) \in NE\left[\left(A_{(l,c)}B_{(l,c)}\right)\right]$ for each subgame $\Gamma_{(c,l)} = \left\langle I_{(c,l)}, J_{(c,l)}, A_{(c,l)}, B_{(c,l)} \right\rangle$ based on the following Algorithm 1.

Here we are going to analyse the following problem**:** *if using the 2DBCMD&D algorithm, and process with the coordinates* (l,c) *determined* $\left(i^*_{(l,c)}, j^*_{(l,c)}\right) \in NE\left[\left(A_{(l,c)}, B_{(l,c)}\right)\right]$, *then which conditions should be checked so that* $\left(\varphi_{(l,c)}\left(i^*_{(l,c)}\right), \psi_{(l,c)}\left(j^*_{(l,c)}\right)\right) \in NE[\Gamma]$.

Using Proposition 1 for the **2DBCMD&D** algorithm we can easily prove the following

Proposition 2 *Let* $\left(i^*, j^*\right) \in NE[\Gamma]$ *and there are a process* (l,c), *applications* $\varphi_{(l,c)} : I_{(l,c)} \to I$, $\psi_{(l,c)} : J_{(l,c)} \to J$ *for which (1)-(2) is verified and* $\left(i^*_{(l,c)}, j^*_{(l,c)}\right) \in NE\left[\left(A_{(l,c)}, B_{(l,c)}\right)\right]$. *Then* $i^* = \varphi_{(l,c)}\left(i^*_{(l,c)}\right)$ *and* $j^* = \psi_{(l,c)}\left(j^*_{(l,c)}\right)$.

The proposition 2 means the following: for any Nash equilibrium profile in the global matrix game there is a subgame generated by the **2DBCMD&D** algorithm, for which this strategy profile is also the equilibrium profile. In the next theorems [3], sufficient conditions are formulated under which a equilibrium profile in the bimatrix subgame, generated by the **2DBCMD&D** algorithm, becomes an equilibrium profile in the initial game with the global matrices.

Theorem 1 *Let's assume thats* $\left(i^*_{(l,c)}, j^*_{(l,c)}\right) \in NE\left[\left(A_{(l,c)}, B_{(l,c)}\right)\right]$ *is determined by the process (l,c.). If for any process on the column* c, *namely* $(\tilde{l}, c)$ *for all* $\tilde{l} \neq l$, *the condition* $j^*_{(\tilde{l},c)} \neq j^*_{(l,c)}$ *is fulfilled, and for any process from the line* l *namely* $(l, \tilde{c})$, *for all* $\tilde{c} \neq c$, *the condition* $i^*_{(l,\tilde{c})} \neq i^*_{(l,c)}$ *is fulfilled, then* $\left(\varphi_{(l,c)}\left(i^*_{(l,c)}\right), \psi_{(l,c)}\left(j^*_{(l,c)}\right)\right) \in NE[\Gamma]$.

This theorem states the following: if in the $A_{(l,c)}$ submatrices of the processes on the column c there are no marked elements which belong to the column $j^*_{(l,c)}$ and, at the same time, in the submatrices $B_{(l,c)}$ of the processes on the line l there are no marked elements which belong to the line $i^*_{(l,c)}$, then the strategy profile $\left(\varphi_{(l,c)}\left(i^*_{(l,c)}\right), \psi_{(l,c)}\left(j^*_{(l,c)}\right)\right)$ is a Nash equilibrium profile in the initial global matrix game.

Let's analyse the case when there are (l,c) processes in the process grid, so that $\left(i^*_{(l,c)}, j^*_{(l,c)}\right) \in NE\left[\left(A_{(l,c)}, B_{(l,c)}\right)\right]$ but $\left(\varphi_{(l,c)}\left(i^*_{(l,c)}\right), \psi_{(l,c)}\left(j^*_{(l,c)}\right)\right) \notin NE[\Gamma]$. In other words, *not every equilibrium profile in the subgame is an equilibrium profile in the global matrix game*.

The following theorem can easily be proved for this case.

Theorem 2 *Supposing for a given* $(l,c) \in L \times C$ *process found strategy profile* $\left(i^*_{(l,c)}, j^*_{(l,c)}\right) \in NE\left[\left(A_{(l,c)}, B_{(l,c)}\right)\right]$. *If for fixed* c *and all* $\tilde{l} \neq l$ *such that* $(\tilde{l}, c) \in L \times C$ *the conditions* $a_{i^*_{(l,c)}, j^*_{(l,c)}} \geq a_{i^*_{(\tilde{l},c)} j^*_{(l,c)}}$ *are fulfilled, where* $i^*_{(\tilde{l},c)} = \arg \max_{i_{(\tilde{l},c)} \in I_{(\tilde{l},c)}} a_{i_{(\tilde{l},c)} j^*_{(l,c)}}$ *and for fixed* l *and all* $\tilde{c} \neq c$ *such that* $(l, \tilde{c}) \in L \times C$ *the conditions* $b_{i^*_{(l,c)}, j^*_{(l,c)}} \geq b_{i^*_{(l,c)} j^*_{(l,\tilde{c})}}$ *are fulfilled where* $j^*_{(l,\tilde{c})} = \arg \max_{j_{(l,\tilde{c})} \in J_{(l,\tilde{c})}} b_{i^*_{(l,c)} j_{(l,\tilde{c})}}$ *then* $\left(\varphi_{(l,c)}\left(i^*_{(l,c)}\right), \psi_{(l,c)}\left(j^*_{(l,c)}\right)\right) \in NE[\Gamma]$.

Finaly the theorem 1and teorem 2 determine the conditions under which the algorithm **2DBCMD&D** becomes the **PMDDA**

References

1. Joseph Jaja, *An Introduction to Parallel Algorithms*, Addison-Wesley Publishing Company, Inc., 1992.
2. "ScaLAPACK -- Scalable Linear Algebra PACKage". [Online]. Available: http://www.netlib.org/scalapack/.
3. Boris Hâncu, Emil Cataranciuc. *Parallel algorithm to solving 2d block-cyclic partitioned bimatrix games*. Studia Universitatis Moldaviae, seria Stiinte exacte si economice, nr. 2(142), Chisinau 2021. p. 26-39.