STUDIA UNIVERSITATIS MOLDAVIAE, 2021, nr.2(142)
Seria "Ştiinţe exacte şi economice" ISSN 1857-2073 ISSN online 2345-1033 p.26-39

CZU:519.832

DOI: http://doi.org/10.5281/zenodo.5094574

PARALLEL ALGORITHM TO SOLVING 2D BLOCK-CYCLIC PARTITIONED BIMATRIX GAMES

Boris HÂNCU, Emil CATARANCIUC
Moldova State University

The article presents a theoretical and practical study of the ways of determining solutions in bimatrix games divided into blocks of submatrices using 2D block-cyclic matrix dividing and distribution algorithm. The proved theorems represent the foundation on which the bimatrix game solution can be built using the sub-games solutions generated by the 2D-cyclic matrix distribution algorithm.

Keywords: non cooperative game, Nash equilibrium, parallel algorithms, distributed memory clusters.

ALGORITM PARALEL PENTRU REZOLVAREA JOCURILOR BIMATRICEALE PARTIŢIONATE CICLIC ÎN BLOCURI 2D

Articolul prezintă un studiu teoretic şi practic al modalităţilor de determinare a soluţiilor în jocurile bimatriceale împărţite în blocuri de submatrice 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.

Cuvinte-cheie: *joc noncooperatist, echilibru Nash, algoritmi paraleli, clustere cu memorie distribuite.*

1 Introduction

Contemporary decision-making problems are very complex and require the processing of a very large volume of data. Thus, for the mathematical modelling of these processes, it is necessary to take into account the big data problems. Big data is huge amount of data which is beyond the processing capacity to manage and analyse the data in a specific time interval. The data is too big to be stored and processed by a single machine. In many large-scale solutions, data is divided into partitions that can be managed and accessed separately. In order to solve such problems in real time, parallel algorithms are built, and then implemented on various types of parallel computing systems. For parallel data processing we must use the ways of dividing, partitioning (sharing) and distributing data. Data partitioning is a technique for physically dividing the data during the loading of the Master Data. Using this method we are going to split the data into smaller pieces according to the rules set by the user. In a distributed-memory parallel systems, load balance is the essential motivation for distributing the data over a collection of processes according to the block-cyclic decomposition scheme [1]. As noted in [2] the two-dimensional block-cyclic distribution has been suggested as a possible general purpose basic decomposition for parallel dense linear algebra software libraries. In [3] there are presented important properties of the two-dimensional block-cyclic data distribution that are the basis of efficient algorithms for address generation, fast indexing techniques and communication scheduling. At the basis of data level parallelisation for BLAS it is the 2D block-cyclic distribution algorithm [4]. For solving systems of simultaneous linear equations, least-squares solutions of linear systems of equations, eigenvalue problems and singular value problems it is used in the same way the 2D block-cyclic distribution algorithm [5]. The ScaLAPACK (or Scalable LAPACK) library includes a subset of LAPACK routines redesigned for distributed memory MIMD parallel computers assumes that matrices are laid out in a two-dimensional block cyclic decomposition[6-7]. The problem of partitioning a matrix

into a set of submatrices has received increased attention in the last few years [8]. This operation is indeed crucial when considering dense linear algebra kernels and other applications with similar communication patterns on heterogeneous platforms.

At the moment, in the opinion of the authors, there are no algorithms that use the partitioning of matrices based on the 2D block-cyclic distribution algorithm for solving bimatrix games with very large matrices. In this article the authors make an attempt to complete this gap.

In section 2 of the article, the way of dividing and distributing the matrices on the one-dimensional linear array of processes, based on the 2D block-cyclic partitioned algorithm, is analysed in detail. There are demonstrated some properties of this algorithm that later are used to determine solutions in bimatrix games.

In sections 3 the following problem is studied and solved: if we divide and distribute the matrices of the bimatrix game using the 2D block-cyclic partitioned algorithm, and using a parallel algorithm is determined the equilibrium profiles in the subgames generated by the distribution algorithm, then which of these equilibrium profiles can be considered equilibrium profiles in the initial problem.

2 Process grid and block-cyclic distribution of matrices

The basic parallel strategy consists of three main steps. The **first step** is to partition the input into several partitions of almost equal sizes. The **second step** is to solve recursively the subproblem defined by each partition of the input. Note that these subproblems can be solved concurrently in the parallel system. 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 [9]. To realise the first step of the parallel strategy, that is to realise data parallelisation, we use the **two-dimensional block-cyclic data layout scheme** [10].

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 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.

In ScaLAPACK, and thus in BLACS, each process grid is enclosed in a context. Similarly, a context is associated with every global matrix in ScaLAPACK. The use of a context provides the ability to have separate "universes" of message passing. This means that a process grid can safely communicate even if other process grids are also communicating. Thus, a context is a powerful mechanism for avoiding unintentional nondeterminism in message passing and provides support for the design of safe, modular software libraries. In MPI this concept is referred to as a communicator.

A context partitions the communication space. A message sent from one context cannot be received in another context. The use of separate communication contexts by distinct libraries (or distinct library routine invocations) insulates the internal communication of a specific library routine from external communication that may be going on within the user's program. In most respects, we can use the terms process grid and context interchangeably. So, context allows us to do the following:

- create arbitrary groups of processes,

- create an indeterminate number of overlapping and/or disjoint process grids,
- isolate the process grids so that they do not interfere with each other.

The choice of an appropriate data distribution heavily depends on the characteristics of flow of the computation in the algorithm. 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. Procedure steps of the **2D block-cyclic matrix dividing and distribution** (**2DBCMD&D**) algorithm are the following:

- Divide up the global array into blocks with m_A rows and n_A columns.
- From now on, think of the global array as composed only of these blocks.
- Distribute first row of array blocks across the first row of the processor grid in order. If you run out processor grid columns cycle back to first column.
- Repeat for the second row of array blocks, with the second row of the processor grid.
- Continue for remaining rows of array blocks.
- If you run out of processor grid rows, cycle back to the first processor row and repeat.

According to the two dimensional block cyclic data distribution scheme, 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 the program fragment in C ++ with the use of BLACS functions for implementation of the **2DBCMD&D** algorithm.

Block which contains Cblacs declarations

```
extern"C"{
void Cblacs_pinfo(int*,int*);
void Cblacs_get(int,int,int*);
void Cblacs_gridinit(int*,const
char*,int,int);
void Cblacs_gridinfo(int,int*,
int*,int*,int*);
void Cblacs_gridexit(int);
void Cblacs_exit(int);
int numroc_(int*,int*,int*,
int*,int*);
int indxl2g_(int*,int*,int*,
int*,int*); }
```

The main program must contain the following blocks:

- building the communication environment:

```
Cblacs_pinfo(&iam,&nprocs);
Cblacs_get(-1,0,&ictxt);
Cblacs_gridinit(&ictxt,"Row",
nprow,npcol);
Cblacs_gridinfo(ictxt,&nprow,
&npcol,&myrow,&mycol);
```

- generating the sub-matrices, here A, B are the local matrices, AA, BB are the global matrices:

```
int m_loc=numroc_(&m,&mb,&myrow,
&rsrc,&nprow);
int k_loc=numroc_(&k,&nb,&mycol,
&rsrc,&npcol);
double *A=(double*)malloc
(m_loc*k_loc*sizeof(double));
double *B=(double*)malloc
(m_loc*k_loc*sizeof(double));
for(iloc=0;iloc<m_loc;iloc++)
for(jloc=0;jloc<k_loc;jloc++){
int fortidl=iloc+1;
int fortjdl=jloc+1;
i=indxl2g_(&fortidl,&mb,
&myrow,&ZERO,&nprow)-1;
j=indxl2g_(&fortjdl,&nb,
&mycol,&ZERO,&npcol)-1;
A[jloc*m_loc+iloc]=*(AA+m*j+i);
B[jloc*m_loc+iloc]=*(BB+m*j+i); }
```

We present below some examples of the distributed matrix based on **2DBCMD&D** algorithm. If we have a global 6×5 matrix $A = \|a_{ij}\|_{i=\overline{1,6}}^{j=\overline{1,5}}$ 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).

	$c = 0$	$c = 1$
$l = 0$	$A_{(0,0)} = \begin{pmatrix} a_{11} & a_{12} & a_{15} \\ a_{21} & a_{22} & a_{25} \\ a_{51} & a_{52} & a_{55} \\ a_{61} & a_{62} & a_{65} \end{pmatrix}$	$A_{(0,1)} = \begin{pmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \\ a_{53} & a_{54} \\ a_{63} & a_{64} \end{pmatrix}$
$l = 1$	$A_{(1,0)} = \begin{pmatrix} a_{31} & a_{32} & a_{35} \\ a_{41} & a_{42} & a_{45} \end{pmatrix}$	$A_{(1,1)} = \begin{pmatrix} a_{33} & a_{34} \\ a_{43} & a_{44} \end{pmatrix}$

Table 1

If we have a global 6×5 matrix $A = \|a_{ij}\|_{i=\overline{1,6}}^{j=\overline{1,5}}$ then for 2-D process grid $L \times C = 2 \times 2$ and block dimension 2×5 we obtain the division and distribution on the process grid represented below.

	$c = 0$	$c = 1$
$l = 0$	$A_{(0,0)} = \begin{pmatrix} a_{11} & a_{12} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{24} & a_{25} \\ a_{51} & a_{52} & a_{54} & a_{55} \\ a_{61} & a_{62} & a_{64} & a_{65} \end{pmatrix}$	*non*
$l = 1$	$A_{(1,0)} = \begin{pmatrix} a_{31} & a_{32} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{44} & a_{45} \end{pmatrix}$	*non*

Table 2

And finally, if we have a global 6×5 matrix $A = \|a_{ij}\|_{i=\overline{1,6}}^{j=\overline{1,5}}$ then for 2-D process grid $L \times C = 2 \times 2$ and block dimension 6×2 we obtain the division and distribution on the process grid represented below.

	$c = 0$	$c = 1$
$l = 0$	$A_{(0,0)} = \begin{pmatrix} a_{11} & a_{12} & a_{15} \\ a_{21} & a_{22} & a_{25} \\ a_{31} & a_{32} & a_{35} \\ a_{41} & a_{42} & a_{45} \\ a_{51} & a_{52} & a_{55} \\ a_{61} & a_{62} & a_{65} \end{pmatrix}$	$A_{(0,1)} = \begin{pmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \\ a_{33} & a_{34} \\ a_{43} & a_{44} \\ a_{53} & a_{54} \\ a_{63} & a_{64} \end{pmatrix}$
$l = 1$	*non*	*non*

Table 3

Let $I_{(l,c)}$ (respectively $J_{(l,c)}$) denotes the rows (respectively columns) of the local matrices. For example, if the matrices are distributed as shown in Table 1 then $I_{(0,0)} = \{1,2,3,4\}$ (respectively columns $J_{(0,0)} = \{1,2,3\}$) when the lines of the global matrix are $\{1,2,5,6\}$(respectively columns $\{1,2,5\}$). Denote by $A_{(c,l)} = \left\|a_{i_{(l,c)}j_{(l,c)}}\right\|_{i_{(l,c)}=\overline{1,|I_{(l,c)}|}}^{j_{(l,c)}=\overline{1,|J_{(l,c)}|}}$ and $B_{(c,l)} = \left\|b_{i_{(l,c)}j_{(l,c)}}\right\|_{i_{(l,c)}=\overline{1,|I_{(l,c)}|}}^{j_{(l,c)}=\overline{1,|J_{(l,c)}|}}$ submatrices formed from global matrices A and B, that are distributed to the $(c,l) \in L \times C$ process grid and by $i_{(l,c)} \in I_{(l,c)}$, respectively $j_{(l,c)} \in J_{(l,c)}$, number of the line (column) in the matrices $A_{(c,l)}$ and $B_{(c,l)}$ which was distributed to the process (l,c). We call $i_{(l,c)}$ and $j_{(l,c)}$ local indices of the local elements of the local matrices $A_{(c,l)}$, $B_{(c,l)}$. We introduce the following applications which determine the correspondence between the "local indices" of the elements of the local matrices $A_{(l,c)}$, $B_{(l,c)}$ and the "global indices" of the elements of the global matrices A and B, specifically $\varphi_{(l,c)} : I_{(l,c)} \to I$, $\psi_{(l,c)} : J_{(l,c)} \to J$. Obviously, these functions must verify the following conditions:

$$\begin{aligned} \forall i \in I\ , \exists l = \overline{1,l}_{\max}, \exists c = \overline{1,c}_{\max}, \exists i_{(l,c)} \in I_{(l,c)} \text{ that } i = \varphi_{(l,c)}\left(i_{(l,c)}\right), \\ \forall j \in J\ , \exists l = \overline{1,l}_{\max}, \exists c = \overline{1,c}_{\max}, \exists j_{(l,c)} \in J_{(l,c)} \text{ that } j = \psi_{(l,c)}\left(j_{(l,c)}\right). \end{aligned} \tag{2.1}$$

Moreover, according to (2.1) the following condition is verified: for any strategy profile in the bimatrix games, namely $(i,j) \in I \times J$, there exist a process $(l,c) \in L \times C$ and the strategy profile $(i_{(l,c)}, j_{(l,c)})$ so that $i = \varphi_{(l,c)}\left(i_{(l,c)}\right)$ and $j = \psi_{(l,c)}(j_{(l,c)})$.

As a result, we analyse only those divisions and distributions of the global matrices in local matrices for which there exist the applications $\varphi_{(l,c)}$ and $\psi_{(l,c)}$ such that the conditions (2.1) are satisfied. It's obvious that the matrix division based on the **2DBCMD&D** algorithm verifies the (2.1) properties. These functions are similar to the INDXL2G ones in the BLACS system [10] that computes the global row index or column index of a distributed matrix entry pointed to by the local index.

It's easy to prove that for **2DBCMD&D** algorithm the $\varphi_{(l,c)}$ and $\psi_{(l,c)}$ functions, additionally to the (2.1) conditions also verify the following conditions.

Proposition 2.1 *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 $\varphi_{(l,\widehat{c})}\left(i_{(l,\widehat{c})}\right) = \varphi_{(l,\widetilde{c})}\left(i_{(l,\widetilde{c})}\right)$ for all $\widehat{c} = \overline{1,c}_{\max}, \widetilde{c} = \overline{1,c}_{\max}$, $\widehat{c} \neq \widetilde{c}$ and $i_{(l,\widehat{c})} = i_{(l,\widetilde{c})}$.*

b) *For all fixed $c = \overline{1,c}_{\max}$, we have $\psi_{(\widehat{l},c)}\left(j_{(\widehat{l},c)}\right) = \psi_{(\widetilde{l},c)}\left(j_{(\widetilde{l},c)}\right)$ for all $\widehat{l} = \overline{1,l}_{\max}, \widetilde{l} = \overline{1,l}_{\max}$, $\widehat{l} \neq \widetilde{l}$ and $j_{(\widehat{l},c)} = j_{(\widetilde{l},c)}$.*

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. In other words, for all submatrices from a processing grid line, all the processes of a given line of the submatrices belong to the same line of the global matrix. For example, according to Table 1, for all processes on the $l = 0$ line, all elements from the 3d line of the submatrix belong to the 5th line of the global matrix. It means that for $i_{(0,0)} = i_{(0,1)} = 3$ we have $\varphi_{(0,c)}\left(i_{(0,c)}\right) = 5$ for any $c = 0,1$. 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. For example, according to Table 1, for all processes on the $c = 0$ column, all elements from the $3d$ column of the submatrix belong to the $5th$ column of the global matrix. Meaning that for $j_{(0,0)} = j_{(1,0)} = 3$ we will have $\psi_{(l,0)}\left(j_{(l,0)}\right) = 5$ for any $l = 0,1$. This property is used to construct the equilibrium profiles in the bimatrix games.

STUDIA UNIVERSITATIS MOLDAVIAE, 2021, nr.2(142)
Seria "Științe exacte și economice" ISSN 1857-2073 ISSN online 2345-1033 p.26-39

3 Nash equilibrium profiles for bimatrix games with block-cyclic distributed matrices

We consider the bimatrix game in the following strategic form

$$\Gamma = \langle I, J, A, B \rangle, \tag{3.1}$$

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

$$(i^*,\ j^*) \in NE[\Gamma] \Leftrightarrow \begin{cases} a_{i^*j^*} \geq a_{ij^*}\ \forall i \in I, \\ b_{i^*j^*} \geq b_{i^*j}\ \forall j \in J. \end{cases}$$

Based on this definition, it is easy to develop the following algorithm for determining the Nash equilibrium profiles in bimatrix games.

Algorithm 3.1

1. For any fixed column $j \in J$, $i^*(j) = Arg\max\limits_{i\in I} a_{ij}$ is determined[1]. 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 $i \in I$, $j^*(i) = Arg\max\limits_{j\in J} b_{ij}$ 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 function graph of the application i^* from step 1) is built: $gr\_i^* = \{(i,j) : i = i^*(j), \forall j \in J\}$ and of the application j^* from step 2) is built as well: $gr\_j^* = \{(i,j) : j = j^*(i), \forall i \in I\}$. The equilibrium profiles are all the profiles belonging to the intersection of the two given function graphs: $NE = gr\_i^* \cap gr\_j^*$. 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.

We assume that global matrices A and B are divided into submatrices and are distributed to the processes from the process grid (communication environment, context, communicator). So we obtain the series of bimatrix subgames in complete and perfect information in the following strategic form:

$$\Gamma_{(c,l)} = \langle I_{(c,l)}, J_{(c,l)}, A_{(c,l)}, B_{(c,l)} \rangle \tag{3.2}$$

Example 3.1 *Consider bimatrix game* $(A, B) = \|(a_{ij}, b_{i,j}\|_{i=\overline{1,6}}^{j=\overline{1,5}}$ *and suppose that the matrices* A *and* B *are distributed according to the two-dimensional block-cyclic data layout scheme. Let's build the array of matrix subgames generated by this method of division and distribution of the matrices.*

Solution. As shown above (see Table 1) according to **2DBCMD&D** algorithm with 2×2 blocks

[1] The notation *Argmax* means that all maximum elements are determined.

and $L \times C = 2 \times 2$ process grid, we get the following submatrices of the array of bimatrix subgames

$$(A_{(0,0)}, B_{(0,0)}) = \begin{pmatrix} (a_{11}, b_{11}) & (a_{12}, b_{12}) & (a_{13}, b_{13}) \\ (a_{21}, b_{21}) & (a_{22}, b_{22}) & (a_{23}, b_{23}) \\ (a_{31}, b_{31}) & (a_{32}, b_{32}) & (a_{33}, b_{33}) \\ (a_{41}, b_{41}) & (a_{42}, b_{42}) & (a_{43}, b_{43}) \end{pmatrix} =$$

$$= \begin{pmatrix} (a_{11}, b_{11}) & (a_{12}, b_{12}) & (a_{15}, b_{15}) \\ (a_{21}, b_{21}) & (a_{22}, b_{22}) & (a_{25}, b_{25}) \\ (a_{51}, b_{51}) & (a_{52}, b_{52}) & (a_{55}, b_{55}) \\ (a_{61}, b_{61}) & (a_{62}, b_{62}) & (a_{65}, b_{65}) \end{pmatrix};$$

$$(A_{(0,1)}, B_{(0,1)}) = \begin{pmatrix} (a_{11}, b_{11}) & (a_{12}, b_{12}) \\ (a_{21}, b_{21}) & (a_{22}, b_{22}) \\ (a_{31}, b_{31}) & (a_{32}, b_{32}) \\ (a_{41}, b_{41}) & (a_{42}, b_{42}) \end{pmatrix} = \begin{pmatrix} (a_{13}, b_{13}) & (a_{14}, b_{14}) \\ (a_{23}, b_{23}) & (a_{24}, b_{24}) \\ (a_{53}, b_{53}) & (a_{54}, b_{54}) \\ (a_{63}, b_{63}) & (a_{64}, b_{64}) \end{pmatrix};$$

$$(A_{(1,0)}, B_{(1,0)}) = \begin{pmatrix} (a_{11}, b_{11}) & (a_{12}, b_{12}) & (a_{13}, b_{13}) \\ (a_{21}, b_{21}) & (a_{22}, b_{22}) & (a_{23}, b_{23}) \end{pmatrix} =$$

$$= \begin{pmatrix} (a_{31}, b_{31}) & (a_{32}, b_{32}) & (a_{35}, b_{35}) \\ (a_{41}, b_{41}) & (a_{42}, b_{42}) & (a_{45}, b_{45}) \end{pmatrix};$$

$$(A_{(1,1)}, B_{(1,1)}) = \begin{pmatrix} (a_{11}, b_{11}) & (a_{12}, b_{12}) \\ (a_{21}, b_{21}) & (a_{22}, b_{22}) \end{pmatrix} = \begin{pmatrix} (a_{33}, b_{33}) & (a_{34}, b_{34}) \\ (a_{43}, b_{43}) & (a_{44}, b_{44}) \end{pmatrix}.$$

So, we obtain the following array of normal forms of bimatrix games:
$\Gamma_{(0,0)} = \langle I_{(0,0)} = \{1,2,3,4\}, J_{(0,0)} = \{1,2,3\}, A_{(0,0)}, B_{(0,0)} \rangle$;
$\Gamma_{(0,1)} = \langle I_{(0,1)} = \{1,2\}, J_{(0,1)} = \{1,2,3,4\}, A_{(0,1)}, B_{(0,1)} \rangle$;
$\Gamma_{(1,0)} = \langle I_{(1,0)} = \{1,2,3\}, J_{(1,0)} = \{1,2\}, A_{(1,0)}, B_{(1,0)} \rangle$;
$\Gamma_{(1,1)} = \langle I_{(1,1)} = \{1,2\}, J_{(1,1)} = \{1,2\}, A_{(1,1)}, B_{(1,1)} \rangle$.

We denote by $NE\left[\Gamma_{(c,l)}\right]$ or $NE\left[\left(A_{(l,c)}B_{(l,c)}\right)\right]$ the set of all Nash equilibrium profiles of the bimatrix game (subgame) $\Gamma_{(c,l)} = \langle I_{(c,l)}, J_{(c,l)}, A_{(c,l)}, B_{(c,l)} \rangle$. 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 $NE\left[\Gamma_{(c,l)}\right]$ 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 $\Gamma_{(c,l)} = \langle I_{(c,l)}, J_{(c,l)}, A_{(c,l)}, B_{(c,l)} \rangle$ based on the following algorithm.

Algorithm 3.2

1. For any column $j_{(l,c)} \in J_{(l,c)}$, $i^*_{(l,c)}(j_{(l,c)}) = Arg \max\limits_{i_{(l,c)} \in I_{(l,c)}} a_{i_{(l,c)} j_{(l,c)}}$ is determined. Under the algorithmic aspect, it can be as follows: for any column $j_{(l,c)}$ of the local matrix $A_{(l,c)}$ all maximal elements on this column are highlighted.

2. For any row $i_{(l,c)} \in I_{(l,c)}$, $j^*_{(l,c)}(i_{(l,c)}) = Arg \max\limits_{j_{(l,c)} \in J_{(l,c)}} b_{i_{(l,c)} j_{(l,c)}}$ is determined. Under the algorithmic aspect, it can be as follows: for any row $i_{(l,c)}$ of the local matrix $B_{(l,c)}$ all maximal elements on this row are highlighted.

3. The graph of the application $i^*_{(l,c)}(\cdot)$ is built, i.e.

$$gr\_i^*_{(l,c)} = \left\{ \left(i_{(l,c)}, j_{(l,c)}\right) : i_{(l,c)} = i^*_{(l,c)}\left(j_{(l,c)}\right), \forall j_{(l,c)} \in J_{(l,c)} \right\}$$

and the graph of the application $j^*_{(l,c)}(\cdot)$ is built as well, i.e.

$$gr\_j^*_{(l,c)} = \left\{ \left(i_{(l,c)}, j_{(l,c)}\right) : j_{(l,c)} = j^*_{(l,c)}\left(i_{(l,c)}\right), \forall i_{(l,c)} \in I_{(l,c)} \right\}.$$

4. The equilibrium profiles are all the profiles belonging to the intersection of these graphs: $NE\left[\Gamma_{(c,l)}\right] = gr\_i^*_{(l,c)} \cap gr\_j^*_{(l,c)}$. Expressed as an algorithm: in the local matrices $A_{(l,c)}$ and $B_{(l,c)}$ we check for all the highlighted elements and the indices of the elements whose positions coincide both in the matrix $A_{(l,c)}$ and in the matrix $B_{(l,c)}$ will be the equilibrium profiles.

Let's analyse the following problem. Consider $\left(i^*_{(l,c)}, j^*_{(l,c)}\right) \in NE\left[(A_{(l,c)}B_{(l,c)})\right]$, so, we have one equilibrium profile in the game $\Gamma_{(c,l)}$, and thus, we have the array of equilibrium profiles $\left\{\left(i^*_{(l,c)}, j^*_{(l,c)}\right)\right\}_{(l,c)\in L\times C}$ and we shall determine what relations there exist between these equilibrium profiles and the equilibrium profile $(i^*, j^*) \in NE[\Gamma]$? That is to say, if in parallel the Nash equilibrium profile $\left(i^*_{(l,c)}, j^*_{(l,c)}\right) \in NE\left[(A_{(l,c)}B_{(l,c)})\right]$ was determined based on the Algorithm 3.2, then how can one construct the equilibrium profile $(i^*, j^*) \in NE[\Gamma]$ without further solving some optimization problems. It is clear that the solution of the stated problem will depend on the division and matrix distribution algorithm. So, the general problem is: what properties must the division and matrix distribution algorithm possess so that having the solutions of the subgames we can build (without solving some optimization problems) solutions of problems with initial matrices.

Here we are going to analyse the following particular problem: *if we divide and distribute the matrices using the 2DBCMD&D algorithm, which equilibrium profiles from the set* $\left\{\left(i^*_{(l,c)}, j^*_{(l,c)}\right)\right\}_{(l,c)\in L\times C}$ *can be considered as equilibrium profiles in the initial problem i.e. which of them belong to the* $NE[\Gamma]$ *set.* In other words, if a given process with the coordinates (l, c) determined using the Algorithm 3.2 $\left(i^*_{(l,c)}, j^*_{(l,c)}\right) \in NE\left[(A_{(l,c)}, B_{(l,c)})\right]$, then which conditions should be checked so that the given equilibrium profile of subgame to be an equilibrium profile for the initial game too (with the global matrices), namely $\left(\varphi_{(l,c)}\left(i^*_{(l,c)}\right), \psi_{(l,c)}\left(j^*_{(l,c)}\right)\right) \in NE[\Gamma]$.

Using Proposition 2.1 for the **2DBCMD&D** algorithm we can easily prove the following

Proposition 3.3 *Let* $(i^*, j^*) \in NE[\Gamma]$ *in the problem (3.1) 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 (2.1) is verified and* $\left(i^*_{(l,c)}, j^*_{(l,c)}\right) \in NE\left[(A_{(l,c)}, B_{(l,c)})\right]$. *Then* $i^* = \varphi_{(l,c)}\left(i^*_{(l,c)}\right)$ *and* $j^* = \psi_{(l,c)}\left(j^*_{(l,c)}\right)$.

This affirmation 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 theorem, 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 3.4 *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*

$$(l,c) \in L^* \times C^* = \left\{(l,c) \in L \times C : NE\left[\left(A_{(l,c)}, B_{(l,c)}\right)\right] \neq \varnothing\right\}$$

using the algorithm 3.2. If for any process on the column c, namely $(\widetilde{l}, c) \in L^* \times C^*$, *for all* $\widetilde{l} \neq l$, *the condition* $j^*_{(\widetilde{l},c)} \neq j^*_{(l,c)}$ *is fulfilled, and for any process from the line l, namely* $(l, \widetilde{c}) \in L^* \times C^*$, *for all* $\widetilde{c} \neq c$, *the condition* $i^*_{(l,\widetilde{c})} \neq i^*_{(l,c)}$ *is fulfilled, then* $\left(\varphi_{(l,c)}\left(i^*_{(l,c)}\right), \psi_{(l,c)}\left(j^*_{(l,c)}\right)\right) \in NE[\Gamma]$

Proof.

The proof of this theorem results directly from the Algorithm 3.2, and from the properties of the **2DBCMD&D** algorithm described in Proposition 2.1.■

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 $\left(i^*_{(l,c)}, j^*_{(l,c)}\right)$ strategy profile is a Nash equilibrium profile in the initial global matrix game. Based on the theorem 3.4 *any equilibrium profile of a subgame (if it exists) is an equilibrium profile for the initial game also.* The theorem reflects the exact variant of the **2DBCMD&D** algorithm represented in the Table 4 for which, for example, $\left(i^*_{(0,0)}, j^*_{(0,0)}\right) = (3,3) \in NE\left[\left(A_{(0,0)}, B_{(0,0)}\right)\right]$ and $\left(\varphi_{(0,0)}\left(i^*_{(0,0)}\right), \psi_{(0,0)}\left(j^*_{(0,0)}\right)\right) = (5,2) \in NE[\Gamma] = \{(5,2),(1,4),(3,5),(4,3)\}$.

	$c=0$			$c=1$	
$l=0$	(a_{11}, b_{11})	(a_{12}, b_{12})	(a_{15}, b_{15})	(a_{13}, b_{13})	$(\underline{a_{14}}, \underline{b_{14}})$
	(a_{21}, b_{21})	(a_{22}, b_{22})	(a_{25}, b_{25})	(a_{23}, b_{23})	(a_{24}, b_{24})
	(a_{51}, b_{51})	$(\underline{a_{52}}, \underline{b_{52}})$	(a_{55}, b_{55})	(a_{53}, b_{53})	(a_{54}, b_{54})
	(a_{61}, b_{61})	(a_{62}, b_{62})	(a_{65}, b_{65})	(a_{63}, b_{63})	(a_{64}, b_{64})
$l=1$	(a_{31}, b_{31})	(a_{32}, b_{32})	$(\underline{a_{35}}, \underline{b_{35}})$	(a_{33}, b_{33})	(a_{34}, b_{34})
	(a_{41}, b_{41})	(a_{42}, b_{42})	(a_{45}, b_{45})	$(\underline{a_{43}}, \underline{b_{43}})$	(a_{44}, b_{44})

Table 4

Remark 3.1 *As a particular case of the theorem 3.4, one can consider the case when the global matrix A is distributed as shown in Table 3 and the global matrix B is distributed as shown in Table 2. For this distribution method, each process in the process grid* $L \times C$ *determines independently the equilibrium profiles of the the game associated to the process. This equilibrium profile is also an equilibrium profile for the initial game without exchanging any data or comparison operations indicated in the Theorem 3.4*

Let's exemplify the theorem 3.4 by the following examples.

Example 3.2 *(The global game doesn't have Nash equilibrium profiles). Consider a bimatrix game with the following matrices*

$$(A,B) = \begin{pmatrix} (\underline{2},1) & (0,0) & (1,\underline{2}) \\ (1,\underline{2}) & (\underline{2},1) & (0,0) \\ (0,0) & (1,\underline{2}) & (\underline{2},1) \end{pmatrix}.$$

Use the theorem 3.4 to determine the Nash equilibrium profiles.

Solution. Underlined elements in matrices show that there are no equilibrium profiles in this game. Applying the **2DBCMD&D** algorithm with 2×2 blocks and $L \times C = 2 \times 2$ process grid, in

general case we obtain:

	$c=0$	$c=1$
$l=0$	$\begin{pmatrix} (a_{11},b_{11}) & (a_{12},b_{12}) \\ (a_{21},b_{21}) & (a_{22},b_{22}) \end{pmatrix}$	$\begin{pmatrix} (a_{13},b_{13}) \\ (a_{23},b_{23}) \end{pmatrix}$
$l=1$	$\begin{pmatrix} (a_{31},b_{31}) & (a_{32},b_{32}) \end{pmatrix}$	$\begin{pmatrix} (a_{33},b_{33}) \end{pmatrix}$

and for our game

	$c=0$	$c=1$
$l=0$	$\begin{pmatrix} (\underline{2},\underline{1}) & (0,0) \\ (1,\underline{2}) & (\underline{2},1) \end{pmatrix}$	$\begin{pmatrix} (\underline{1},\underline{2}) \\ (0,\underline{0}) \end{pmatrix}$
$l=1$	$\begin{pmatrix} (\underline{0},0) & (\underline{1},\underline{2}) \end{pmatrix}$	$\begin{pmatrix} (\underline{2},\underline{1}) \end{pmatrix}$

Thus, we obtain that $NE\left[\Gamma_{(0,0)}\right] = \{(1,1) = \left(i_{(0,0)}, j_{(0,0)}\right) = (1,1)\}$, $NE\left[\Gamma_{(0,1)}\right] = \{(1,1) = \left(i_{(0,1)}, j_{(0,1)}\right) = (1,3)\}$, $NE\left[\Gamma_{(1,0)}\right] = \{(1,2) = \left(i_{(1,0)}, j_{(1,0)}\right) = (3,2)\}$, $NE\left[\Gamma_{(1,1)}\right] = \{(1,1) = \left(i_{(1,1)}, j_{(1,1)}\right) = (3,3)\}$. Now we apply the theorem 3.4 and we get that none of these equilibrium profiles is the equilibrium profile in the initial game because, for example, for the strategies profiles in $NE\left[\Gamma_{(0,0)}\right]$ on the $1st$ row there are marked elements from the $1st$ column (if we look at it as at a single matrix).

Let's analyse the case when there are (l,c) processes in the process grid, so that $\left(i^*_{(l,c)}, j^*_{(l,c)}\right) \in NE\left[(A_{(l,c)}, B_{(l,c)})\right]$ 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.*

Theorem 3.5 *Supposing for a given $(l,c) \in L \times C$ process, using the algorithm 3.2, we found strategy profile $\left(i^*_{(l,c)}, j^*_{(l,c)}\right) \in NE\left[(A_{(l,c)}, B_{(l,c)})\right]$. If for fixed c and all $\widetilde{l} \neq l$ such that $(\widetilde{l},c) \in L \times C$ the conditions $\mathfrak{a}_{i^*_{(l,c)}, j^*_{(l,c)}} \geq \mathfrak{a}_{i^*_{(\widetilde{l},c)} j^*_{(l,c)}}$ are fulfilled, where $i^*_{(\widetilde{l},c)} \equiv i^*_{(\widetilde{l},c)}(j^*_{(l,c)}) = \arg\max_{i_{(\widetilde{l},c)} \in I_{(\widetilde{l},c)}} \mathfrak{a}_{i_{(\widetilde{l},c)} j^*_{(l,c)}}$ and for fixed l and all $\widetilde{c} \neq c$ such that $(l,\widetilde{c}) \in L \times C$ the conditions $\mathfrak{b}_{i^*_{(l,c)}, j^*_{(l,c)}} \geq \mathfrak{b}_{i^*_{(l,c)} j^*_{(l,\widetilde{c})}}$ are fulfilled where $j^*_{(l,\widetilde{c})} \equiv j^*_{(l,\widetilde{c})}(i^*_{(l,c)}) = \arg\max_{j_{(l,\widetilde{c})} \in J_{(l,\widetilde{c})}} \mathfrak{b}_{i^*_{(l,c)} j_{(l,\widetilde{c})}}$ then $\left(\varphi_{(l,c)}\left(i^*_{(l,c)}\right), \psi_{(l,c)}\left(j^*_{(l,c)}\right)\right) \in NE[\Gamma]$.*

Proof. We analyse for every fixed column c in the process grid the submatrices of the global matrix A distributed across the grid and at the same time, for any fixed line l in the process gird the submatrices of the global matrix B distributed across the grid. According to the algorithm 3.2 if for process $(l,c) \in L \times C$ the strategy profile $\left(i^*_{(l,c)}, j^*_{(l,c)}\right) \in NE\left[(A_{(l,c)}, B_{(l,c)})\right]$ then we have $\left(i^*_{(l,c)}, j^*_{(l,c)}\right) \in gr\_i^*_{(l,c)} \cap gr\_j^*_{(l,c)}$. Similarly, according to the algorithm 3.1 if the strategy profile $(i^*, j^*) \in NE[\Gamma]$ than we have $(i^*, j^*) \in gr\_i^* \cap gr\_j^*$. As a result it is sufficient to prove that based on the conditions of the theorem, the strategy profile $\left(\varphi_{(l,c)}\left(i^*_{(l,c)}\right), \psi_{(l,c)}\left(j^*_{(l,c)}\right)\right) \in gr\_i^* \cap gr\_j^*$. In other words, the values $i^*_{(l,c)}$ and $j^*_{(l,c)}$ should be the solution to the following system $\begin{array}{l} \varphi_{(l,c)}\left(i^*_{(l,c)}\right) = \arg\max\limits_{i \in I} a_{ij}, \\ \psi_{(l,c)}\left(j^*_{(l,c)}\right) = \arg\max\limits_{j \in J} b_{ij}, \end{array}$ i.e. $\varphi_{(l,c)}\left(i^*_{(l,c)}\right) = \arg\max\limits_{i \in I} a_{i\psi_{(l,c)}\left(j^*_{(l,c)}\right)}$ and $\psi_{(l,c)}\left(j^*_{(l,c)}\right) = \arg\max\limits_{j \in J} b_{\varphi_{(l,c)}\left(i^*_{(l,c)}\right) j}$. It is easy to see that according to the conditions of the theorem, properties a), b) of the **2DBCMD&D** algorithm from Proposition 2.1, the last two conditions are verifiable and we get that $\left(\varphi_{(l,c)}\left(i^*_{(l,c)}\right), \psi_{(l,c)}\left(j^*_{(l,c)}\right)\right) \in NE[\Gamma]$. ■

The Theorem 3.5 reflects the exact variant of the **2DBCMD&D** algorithm represented in the Table 5, where the elements that are at the intersection of line l and column c represent a submatrix. Here is assumed that the strategy profile $\left(i^*_{(0,0)}, j^*_{(0,0)}\right) = (3,2) \in NE\left[\left(A_{(0,0)}, B_{(0,0)}\right)\right]$ and $\left(\varphi_{(0,0)}(3), \psi_{(0,0)}(2)\right) = (5,2) \in NE[A,B]$.

	$c=0$			$c=1$	
$l=0$	(a_{11},b_{11})	(a_{12},b_{12})	(a_{15},b_{15})	(a_{13},b_{13})	$(\underline{a_{14}},\underline{b_{14}})$
	(a_{21},b_{21})	(a_{22},b_{22})	(a_{25},b_{25})	(a_{23},b_{23})	(a_{24},b_{24})
	(a_{51},b_{51})	$(\underline{a_{52}},\underline{b_{52}})$	(a_{55},b_{55})	(a_{53},b_{53})	$(\underline{a_{54}},\underline{b_{54}})$
	(a_{61},b_{61})	(a_{62},b_{62})	(a_{65},b_{65})	(a_{63},b_{63})	(a_{64},b_{64})
$l=1$	(a_{31},b_{31})	(a_{32},b_{32})	(a_{35},b_{35})	(a_{33},b_{33})	(a_{34},b_{34})
	(a_{41},b_{41})	$(\underline{a_{42}},\underline{b_{42}})$	(a_{45},b_{45})	$(\underline{a_{43}},\underline{b_{43}})$	(a_{44},b_{44})

Table 5

Let's exemplify the theorem 3.5 by using the following examples

Example 3.3 *Consider the following game (all the strategies profiles in the initial game are equilibrium profiles)*

$$(A,B) = \begin{pmatrix} (\underline{1},2) & (\underline{1},2) & (\underline{1},2) & (\underline{1},2) & (\underline{1},2) \\ (\underline{1},2) & (\underline{1},2) & (\underline{1},2) & (\underline{1},2) & (\underline{1},2) \\ (\underline{1},2) & (\underline{1},2) & (\underline{1},2) & (\underline{1},2) & (\underline{1},2) \\ (\underline{1},2) & (\underline{1},2) & (\underline{1},2) & (\underline{1},2) & (\underline{1},2) \\ (\underline{1},2) & (\underline{1},2) & (\underline{1},2) & (\underline{1},2) & (\underline{1},2) \end{pmatrix}.$$

Using the theorem 3.5 to determine the Nash equilibrium profiles.

Solution. Underlined elements in matrices show that all strategies profiles are the Nash equilibrium profiles. Applying the **2DBCMD&D** algorithm with 2×2 blocks and $L \times C = 2 \times 3$ process grid, we obtain the following set of subgames:

	$c=0$	$c=1$	$c=2$
$l=0$	$\begin{pmatrix} (\underline{1},\underline{2}) & (\underline{1},\underline{2}) \\ (\underline{1},\underline{2}) & (\underline{1},\underline{2}) \\ (\underline{1},\underline{2}) & (\underline{1},\underline{2}) \end{pmatrix}$	$\begin{pmatrix} (\underline{1},\underline{2}) & (\underline{1},\underline{2}) \\ (\underline{1},\underline{2}) & (\underline{1},\underline{2}) \\ (\underline{1},\underline{2}) & (\underline{1},\underline{2}) \end{pmatrix}$	$\begin{pmatrix} (\underline{1},\underline{2}) \\ (\underline{1},\underline{2}) \\ (\underline{1},\underline{2}) \end{pmatrix}$
$l=1$	$\begin{pmatrix} (\underline{1},\underline{2}) & (\underline{1},\underline{2}) \\ (\underline{1},\underline{2}) & (\underline{1},\underline{2}) \end{pmatrix}$	$\begin{pmatrix} (\underline{1},\underline{2}) & (\underline{1},\underline{2}) \\ (\underline{1},\underline{2}) & (\underline{1},\underline{2}) \end{pmatrix}$	$\begin{pmatrix} (\underline{1},\underline{2}) \\ (\underline{1},\underline{2}) \end{pmatrix}$

It is easy to see that in this case we can't use the theorem 3.4 while using the theorem 3.5 we obtain that all equilibrium profiles in subgames are equilibrium profiles in the initial game with the global matrices.

Example 3.4 *Determine the equilibrium profiles in the following bimatrix game*

$$(A,B) = \begin{pmatrix} (\underline{400},0) & (0,\underline{200}) & (0,100) & (0,0) & (\underline{0},-100) \\ (300,0) & (\underline{300},0) & (0,\underline{100}) & (0,0) & (\underline{0},-100) \\ (200,\underline{0}) & (200,\underline{0}) & (\underline{\mathit{200}},\underline{\mathit{0}}) & (0,\underline{0}) & (\underline{0},-100) \\ (100,\underline{0}) & (100,\underline{0}) & (100,\underline{0}) & (\underline{\mathit{100}},\underline{\mathit{0}}) & (\underline{0},-100) \\ (0,\underline{0}) & (0,\underline{0}) & (0,\underline{0}) & (0,\underline{0}) & (\underline{\mathit{0}},\underline{\mathit{0}}) \end{pmatrix}$$

when the matrices are divided and distributed using the 2D cyclic algorithm.

Solution. We determine the equilibrium profiles by the method of the intersection of the graphs of the multiset applications of the best-response type. So $i^*(1) = \{1\}, j^*(1) = \{2\}$; $i^*(2) = \{2\}$, $j^*(2) = \{3\}$; $i^*(3) = \{3\}$, $j^*(3) = \{1,2,3,4\}$; $i^*(4) = \{4\}$, $j^*(4) = \{1,2,3,4\}$; $i^*(5) = \{1,2,3,4,5\}$, $j^*(5) = \{1,2,3,4,5\}$; $i^*(6) = \{1,2,3,4,5\}$, $j^*(6) = \{1,2,3,4,5,6\}$. Now we calculate the graphs $gr\_i^* = \{(1,1), (2,2), (3,3), (1,3), (2,3), \boxed{(3,3)}, \boxed{(4,4)}, (1,5), (2,5), (3,5), (4,5), \boxed{(5,5)}, (1,6), (2,6), (3,6), (4,6), (5,6)\}$ and $gr\_j^* = \{(1,2), (2,3), (3,1), (3,2), \boxed{(3,3)}, \boxed{(3,4)}, \boxed{(4,4)}, (4,1), (4,2), (4,3), (4,4), (5,1), (5,2), (5,3), (5,4), \boxed{(5,5)}, (6,1), (6,2), (6,3), (6,4), (6,5)\}$. It's obvious that $gr\_i^* \cap gr\_j^* = \{(3,3), (4,4)\}$. The equilibrium profiles of this game are $NE = \{(3,3), (4,4), (5,5)\}$. We use the **2DBCMD&D** algorithm when the block dimension is 2×2, the size of the process grid is 2×3 and in general case we get the following matrix distribution:

	$c=0$		$c=1$		$c=2$
$l=0$	(a_{11}, b_{11})	(a_{12}, b_{12})	(a_{13}, b_{13})	(a_{14}, b_{14})	(a_{15}, b_{15})
	(a_{21}, b_{21})	(a_{22}, b_{22})	(a_{23}, b_{23})	(a_{24}, b_{24})	(a_{25}, b_{25})
	(a_{51}, b_{51})	(a_{52}, b_{52})	(a_{53}, b_{53})	(a_{54}, b_{54})	$\boxed{(a_{55}, b_{55})}$
$l=1$	(a_{31}, b_{31})	(a_{32}, b_{32})	$\boxed{(a_{33}, b_{33})}$	(a_{34}, b_{34})	(a_{35}, b_{35})
	(a_{41}, b_{41})	(a_{42}, b_{42})	(a_{43}, b_{43})	$\boxed{(a_{44}, b_{44})}$	(a_{45}, b_{45})

So, we obtain the following set of subgames:

	$c=0$		$c=1$		$c=2$
$l=0$	(400,0)	(0, 200)	(0, 100)	(0, 0)	(0, −100)
	(300, 0)	(300, 0)	(0, 100)	(0, 0)	(0, −100)
	(0, 0)	(0, 0)	(0, 0)	(0, 0)	$\boxed{(0,0)}$
$l=1$	(200, 0)	(200, 0)	$\boxed{(200,0)}$	(0, 0)	(0, −100)
	(100, 0)	(100, 0)	(100, 0)	$\boxed{(100,0)}$	(0, −100)

Here the elements that are at the intersection of line l and column c reprezent a submatrix. The Nash equilibrium profiles in subgames and the corresponding strategy profiles in the initial games of the global matrices are represented in the following table:

(l,c)	$\left(i^*_{(l,c)}, j^*_{(l,c)}\right)$	$\left(\varphi_{(l,c)}\left(i^*_{(l,c)}\right), \psi_{(l,c)}\left(j^*_{(l,c)}\right)\right)$	*Theorem* 3.5
(0,0)	(2,2)	(2,2)	*No*
(0,1)	{(1,1); (2,1); (3,1); (3,2)}	(1,3); (2,3); (5,3); (5,4)	*No*
(0,2)	{(1,1); (2,1); (3,1)}	(1,5); (2,5); (5,5)	(3,1)
(1,0)	(1,1); (1,2)	(3,1); (3,2)	*No*
(1,1)	(1,1); (2,2)	(3,3); (4,4)	(1,1); (2,2)
(1,2)	(1,1); (2,1)	(3,4); (4,5)	*No*

As a result we have built the $NE = \{(3,3), (4,4), (5,5)\}$ set.

Theorem 3.5 generates the next parallel algorithm for determining the Nash equilibrium profiles from the $NE[\Gamma]$ set, using the **2DBCMD&D** algorithm.

Algorithm 3.6

1. For a parallel DMM system a virtual communicator (or context) is generated with a two dimensional Cartesian topology $L \times C$ of a network type. Each $(l,c) \in L \times C$ process, in parallel and independently, using the **2DBCMD&D** algorithm, builds a corresponding $\Gamma_{(c,l)} = \langle I_{(c,l)}, J_{(c,l)}, A_{(c,l)}, B_{(c,l)} \rangle$ matrix subgame.

2. Each process $(l,c) \in L \times C$ of a parallel DMM system determines independently the set of the Nash equilibrium profiles $NE[\Gamma_{(c,l)}]$ using the algorithm 3.2.

3. According to the theorem 3.5, based on the elements of $NE[\Gamma_{(c,l)}]$ the elements from $NE[\Gamma]$ are built. In other words, for any given strategy profile $\left(i^*_{(l,c)}, j^*_{(l,c)}\right) \in NE\left[\left(A_{(l,c)}, B_{(l,c)}\right)\right]$ we verify if $\left(\varphi_{(l,c)}\left(i^*_{(l,c)}\right), \psi_{(l,c)}\left(j^*_{(l,c)}\right)\right) \in NE[\Gamma]$ in the following way:

   a) for fixed c and all $\widetilde{l} \neq l$ such that $(\widetilde{l},c) \in L \times C$ we verify if $a_{i^*_{(l,c)}j^*_{(l,c)}} \geq a_{i^*_{(\widetilde{l},c)}j^*_{(l,c)}}$ where $i^*_{(\widetilde{l},c)} \equiv i^*_{(\widetilde{l},c)}(j^*_{(l,c)}) = \arg \max_{i_{(\widetilde{l},c)} \in I_{(\widetilde{l},c)}} a_{i_{(\widetilde{l},c)}j^*_{(l,c)}}$.

   b) for fixed l and all $\widetilde{c} \neq c$ such that $(l,\widetilde{c}) \in L \times C$ we verify if $b_{i^*_{(l,c)}j^*_{(l,c)}} \geq b_{i^*_{(l,c)}j^*_{(l,\widetilde{c})}}$ where $j^*_{(l,\widetilde{c})} \equiv j^*_{(l,\widetilde{c})}(i^*_{(l,c)}) = \arg \max_{j_{(l,\widetilde{c})} \in J_{(l,\widetilde{c})}} b_{i^*_{(l,c)}j_{(l,\widetilde{c})}}$.

4 So, if the conditions a) and b) are verified simultaneously, then $\left(\varphi_{(l,c)}\left(i^*_{(l,c)}\right), \psi_{(l,c)}\left(j^*_{(l,c)}\right)\right) \in NE[\Gamma]$.

4 Conclusions and future work

It is well known that the following steps must be taken to develop any parallel algorithm: a) partition the input into several partitions of almost equal sizes and distribute this data on a parallel computing system; b) solve recursively the subproblem defined by each partition of the input; c) combine or merge the solutions of the different subproblems into a solution for the overall problem. In this article, to solve the bimatrix games in complete and imperfect information over the sets of pure strategies we elaborate the parallel algorithm for which: step a) is achieved by using the **2DBCMD&D** algorithm; to perform step b) the Algorithm 3.1 is used; and step c) is carried out using the Theorem 3.2. Thus, the main results of this article can serve as a basis for solving bimatrix games with very large matrices.

We consider that, as a continuation of the research, can be used different ways of dividing and sharing (partitioning) matrices and the launching of a comparative study of the execution time in solving the generated subgames. Also future work can include the design of CPU/GPU-based algorithms implementing other methods for computing Nash equilibria.

References:

[1] J. Dongarra and D. Walker, "Software Libraries for Linear Algebra. Computations on High Performance Computers," SIAM Review, vol. 37, no. 2, pp. 151-180, 1995.

[2] V. Kumar, A. Grama, A. Gupta, and G. Karypis, "Introduction to Parallel Computing". Redwood City, Calif.: Benjamin/Cummings. Publishing Company, Inc., 1994

[3] Antoine P. Petutet and Jack J. Dongarra "Algoritmic Redistribuition Mtehods for Block-Cyclic Decompositions" IEEE Transactions on parallel adn distributed systems, vol.10, no, 12 1999

[4] M. Dayde, I. Duff, and A. Petitet, "A Parallel Block Implementation of Level 3 BLAS for MIMD Vector Processors" ACM Trans.Mathematical Software, vol. 20, no. 2, pp. 178-193, 1994.

[5] Jack J. Dongarra , Iain S. Duff , Danny C. Sorensen and Henk A. van der Vorst "Numerical Linear Algebra for High-Performance Computers". 1998

[6] Choi, J.; Dongarra, J. J.; Pozo, R.; Walker, D. W. "ScaLAPACK: a scalable linear algebra library for distributed memory concurrent computers". Proceedings of the Fourth Symposium on the Frontiers of Massively Parallel Computation. p. 120. doi:10.1109/FMPC.1992.234898. ISBN 978-0-8186-2772-9. 1992

[7] Piotr Luszczek, Jack J. Dongarra "Linear algebra - software issues." Scholarpedia, 2011. [Online]. Available: http://www.scholarpedia.org/article/Linear_algebra_-_software_issue

[8] Olivier Beaumont, Brett A. Becker, Ashley DeFlumere, Lionel Eyraud-Dubois, Thomas Lambert, and Alexey Lastovetsky. "Recent Advances in Matrix Partitioning for Parallel Computing on Heterogeneous Platforms". 2018. [Online]. Available: https://hal.inria.fr/hal-01670672v2.

[9] Joseph Jaja, *An Introduction to Parallel Algorithms*, Addison-Wesley Publishing Company, Inc., 1992.

[10] "ScaLAPACK – Scalable Linear Algebra PACKage". [Online]. Available: http://www.netlib.org/scalapack/.

Date despre autori:

Boris HÂNCU, doctor în ştiinţe matematice, conferenţiar universitar, Universitatea de Stat din Moldova.

Email: boris.hancu@gmail.com.

Emil CATARANCIUC, doctorand *Şcoala doctorală de Ştiinţe Fizice, Matematice, ale Informaţiei şi Inginereşti,* Universitatea de Stat din Moldova.

Email: ecataranciuc@gmail.com

Prezentat la 16.12.2020