DETERMINAREA SITUAȚIILOR NASH DE ECHILIBRU ÎN BAZA ALGORITMULUI 2D-CICLIC

Cristian CEMÎRTAN,

student, anul III, Facultatea Matematică și Informatică

CZU: 519.833 cemirtan.cristian@outlook.com

Determinarea situațiilor de echilibru Nash implică găsirea perechilor de indici (i^*,j^*) în două matrice de câștig ce fac parte din jocul bimatricial $\Gamma = (I,J,A,B)$ [1]. Numim perechea de indici (i^*,j^*) o situație de echilibru dacă se satisface condiția:

$$\begin{cases} a_{i^\bullet j^\bullet} \geq a_{ij^\bullet}, \forall i \in I \\ b_{i^\bullet j^\bullet} \geq b_{i^\bullet j}, \forall j \in J \end{cases} \quad (1)$$

Având la bază condiția (1), fie mulțimea NE totalitatea situațiilor de echilibru:

$$NE = \left\{(i^*,j^*)\middle| i^* \in I, j^* \in J, a_{i^*j^*} = \max_{i \in I} a_{ij^*}\right\} \cap \left\{(i^*,j^*)\middle| i^* \in I, j^* \in J, b_{i^*j^*} = \max_{j \in J} b_{i^*j}\right\} \quad (2)$$

Luând în considerare că matricele pot avea dimensiuni mari, găsirea soluțiilor va fi costisitoare în resursele de calcul pentru un singur proces (procesor sau fir de execuție), de exemplu: timpul de execuție mare sau utilizarea excesivă a memoriei operative.

Evitarea acestui neajuns se realizează prin paralelizare – partiționarea matricelor în submatrice (blocuri) mai mici pentru toate procesele dintr-un comunicator, și apoi interclasarea soluțiilor de la procese. Utilizând algoritmul 2D-ciclic (altfel zis, ca distribuție bloc-ciclică bidimensională), paralelizarea se realizează astfel încât dimensiunile submatricelor să fie cât mai uniforme pentru toate procesele.

În lucrarea [2] autorii și-au expus opinia că: în afară de contribuția lor, „*nu există alte implementări ce rezolvă jocurile bimatriciale pe baza algoritmului 2D-ciclic*”. Implementarea algoritmului din lucrarea [2] constă dintr-un program, în limbajul C++, ce utilizează biblioteca ScaLAPACK pentru a gestiona o grilă de procese asociate algoritmului 2D-ciclic.

Însă în viziunea noastră, respectiva implementare prezintă următoarele obstacole:

- Se utilizează gestionarea manuală a memoriei dinamice pentru generarea submatricelor;
- Submatricele generate sunt compuse din valori cu tipul de date `double`.

Algoritmul vizat poate fi reimplementat, utilizând biblioteca STL (Standard Template Library) din limbajul de programare C++, astfel încât să fie evitată complet gesti-

onarea manuală a memoriei dinamice [3]. De asemenea, în loc de ScaLAPACK, putem utiliza biblioteca MPI pentru a crea tipuri noi de date ce ne permite să generăm submatrice compuse din orice tip de date [4].

În prezenta lucrare, autorul vine cu o implementare proprie a algoritmului ce determină situațiile de echilibru Nash în jocurilor bimatriciale, în baza algoritmului 2D-ciclic, în limbajul de programare C++, cu următoarele adaptări în utilizarea tehnicilor de programare:

1. Submatricele nu devin încărcate imens în memoria operativă (RAM);
2. Se vor utiliza structurile de date dinamice din STL.

În concluzie, algoritmul implementat constă din două paralelizări:

- La nivel de date – procesele în mod individual preiau submatricele dintr-un fișier comun;
- La nivel de operații – fiecare proces determină indicii locali și apoi îi expediază procesului root, care va calcula mulțimea finală NE din indicii globali.

Implementarea algoritmului a fost elaborată în limbajul de programare C++14, utilizând doar biblioteca standard STL și MPI.

Programul a fost compilat și testat pe clusterul modern HPC al Universității de Stat din Moldova: http://hpc2.usm.md. Programul realizat este disponibil pe linkul: https://github.com/cemti/usm-mpi/blob/main/Lab5/lab5.cpp. Implementarea realizată este scalabilă și poate fi rulată pe orice număr de procese.

Referințe:

1. HÎNCU, Boris, CALMÎȘ, Elena. Modele de programare paralelă pe clustere. Programare MPI, CEP USM, 2016, Chișinău, 115 pagini. În:https://www.studocu.com/row/document/academia-de-studii-economice-din-moldova/programarea-aplicatiilor-distribuite/modele-de-programare-paralela-pe-clustere-programare-mpi-editia-2/8121018. Accesat 17.03.2024.
2. HÂNCU, Boris, CATARANCIUC, Emil. Parallel algorithm to solving 2D block-cyclic partitioned bimatrix games. In: Studia Universitatis Moldaviae (Seria Ştiinţe Exacte şi Economice), 2021, nr. 2(142), pp. 26-39. ISSN 1857-2073. În: https://doi.org/10.5281/zenodo.5094574 . Accesat 17.03.2024.
3. Microsoft Learn, C++ Standard Library (STL) overview, În:https://learn.microsoft.com/en-us/cpp/standard-library/cpp-standard-library-overview. Accesat 19.03.2024.
4. Microsoft Learn, MPI Reference, În:https://learn.microsoft.com/en-us/message-passing-interface/mpi-reference. Accesat 19.03.2024.

Recomandat
***Maria CAPCELEA**, dr., conf. univ.*