15-418/618 Spring 2023
Parallel Computation of Nash Equilibria - Project Proposal

Xiaoyu Zhang:	xiaoyuz4@andrew.cmu.edu
Yizhou chen:	yizhouch@andrew.cmu.edu

URL

SUMMARY

In this project, we will implement multiple parallelized algorithms for finding Nash equilibria in games with varying numbers of players and strategy types (pure or mixed).

BACKGROUND

Nash equilibrium is a concept in game theory that describes a state in which each player in a game is making the best possible decision given the decisions of all the other players. In other words, no player can improve their own outcome by changing their strategy, assuming that all other players keep their strategies unchanged.

Given a standard input of a game, there are many algorithms for computing Nash equilibria: pure searching, solving systems of polynomial equations and inequalities, extreme point enumeration, iterated polymatrix approximation, linear complementarity, function minimization, etc.

THE CHALLENGE

Finding Nash equilibria involves solving a system of nonlinear equations and it this is a non-convex problem, there is no trivial algorithms for solving it, we are going to spend considerable amount of time understanding and designing parallizing the algorithms.
What's more, the number of players and strategy types can significantly increase the search space, making the problem more complex. The widely used Python library Nashypy can only be used to find Nash equilibria for two players, scaling the problem can increase the complexity significantly.
We plan to implement our own library that support multiple algorithms, for each algorithm, we have to go through the process of finding the part to be parallelized, design the algorithm and implement, which involves considerable workload.

RESOURCES

1. Reference Codebase: **Gambit**

   Gambit is a library of game theory software and tools. It implements several Nash equilibrium solving algorithms.

2. Reference Paper: **Parallel Computation of Nash equilibria in N-Player Games**

   This paper proposed a parallel algorithm for solving Nash equilibria in n-player games.

3. Computing Resources

   We are considering using GHC machine & PSC machine for developing parallel algorithms.

GOALS AND DELIVERABLES

75% Implement and parallelize three algorithms for solving Nash equilibria in 2-player games: enumeration, solving systems of polynomial equations and inequalities, and the global Newton method.

100% Extend the algorithms to computing Nash equilibria in N-player games.

125% Implement a demo of applying the algorithms in a real-world game scenario.

PLATFORM CHOICE

We choice to use C++ to implement our algorithms for several reasons: 1. C++ is a compiled language that can be optimized by the compiler for maximum performance, and performance is what we care in this project. 2. We are comfortable using techniques like CUDA, OpenMP and MPI in C++, it will save our learning costs.

SCHEDULE

04/01 - 04/10 Implement the sequential version of three algorithms for Nash equilibria in 2-player games
04/11 - 04/22 Implement parallel version of three algorithms for Nash equilibria in 2-player games
04/22 - 05/01 Scale algorithm to N-player games and implement both sequential and parallel version. And integrate all algorithms to our own Python library.