CPSC 490 Project Proposal

Peter Nguyen
Advisor: Andrew Sherman

Background:

Game theory is the study of strategic interactions: the formalization of rational choice towards optimal decision making in a situation among multiple actors, given each actors' own strategies and preferences.

Since its inception in the 1920s by John von Neumann, game theory has been studied and applied to numerous fields, including economics, biology and international relations. In the field of economics, game theory can be used to model bargaining and contract situations between multiple firms. These models can explain the intentions behind each firm's actions as well as suggest a strategically dominant position one may take given the environment. Evolutionary biology draws on game theory to understand how some "strategies", in this case genetic traits, perform among alternative traits, and how the outperforming strategies are linked to evolutionarily acquired traits. International relations theory and foreign policy map closely to concepts in game theory: the idea of multiple, self interested states acting among each other in a strategic environment has been used to model the strategic relationship between the United States and the USSR during the Cold War. These models have helped shape US foreign policy in the face of mutually assured destruction (MAD) as well as current issues as the US enters an era of multipolar powers.

One important concept in game theory is the Nash Equilibrium, or NE. Given a set of n-players, their strategy sets, and a payoff function given each combination of the players' strategies, a nash equilibrium is the strategy vector such that no individual player can receive a strictly better payoff by choosing an alternative strategy. There are two main forms of NE: pure strategy nash equilibrium (PSNE) and mixed strategy nash equilibrium (MSNE). The MSNE differs from the PSNE in that rather than choosing from 1 of an enumerable number of strategies, a player may choose a subset of their strategies over a probability distribution. This presents a problem computationally, as the problem of finding MSNEs grows exponentially on the number of strategies available to each player.

Unlike for many other computationally hard problems, it has been proven that for all games, there exists a MSNE. Although this proof of existence is a welcome fact, this unfortunately does not solve the issue of the computational expensiveness of finding such equilibria, especially the question of scalability given ever growing problem sizes. The infeasibility of computing MSNEs in an acceptable amount of time remains a problem in the application of game theory in large datasets, including in the other aforementioned fields. This point motivates our project. We explore miscellaneous algorithms used to compute these equilibria and discuss the possibilities of optimizing these algorithms as much as we can using high performance parallel computation.

Project:

For this project, we will examine multiple algorithms for solving MSNEs, determine their parallelizability and implement parallel implementations of them on various parallel computing platforms. Platforms that will be used include NVIDIA GPUs, MPI, and the Xeon Phi coprocessor.

The different platforms above provide particular advantages and disadvantages. NVIDIA GPUs enable potentially massive parallel computations via its single instruction, multiple thread (SIMT) architecture. However, a downside to this architecture is a situation called thread divergence, where control flow and branching can hinder performance on the GPU. Because each group of threads must execute the same instruction in parallel, the threads cannot execute separate branches in a control flow statement in parallel and instead must execute each branch serially and wait for all threads to complete. The Xeon Phi coprocessor is a more recent development. It allows for a more flexible code, allowing for less consideration on thread divergence and more general purpose parallel threads. However, the Xeon Phi has significantly less cores than GPUs. MPI grants us the use of multiple machines in decentralized networks. However, we may find that data interchange between nodes can be a bottleneck to performance.

We will survey the algorithms that have been proposed in literature. Potential algorithms that will be explored include the Lemke-Howson algorithm (to solve 2-player games), the Govindan-Wilson algorithm (for n-player games), as well as the brute force enumeration methods for both 2 and n-player games. We will also examine current parallel algorithms, particularly those proposed by Grosu. After gaining a working understanding of these algorithms, we will select a set of these algorithms, and design and implement parallel versions of the algorithms on the various platforms as described above. We will benchmark and analyze the performance of each algorithm on each of the platforms, and discuss how the unique advantages of each platform and the parallelizability of each algorithm affect the resulting performance.

In order to determine what games to use as input to these algorithms, we will also research various classes of games that carry particular properties with respect to computational cost. In order to obtain sample games, we will use the GAMUT suite to generate instances of the classes of games.

Deliverables:

The deliverables will primarily consist of the code used to implement these parallel algorithms written in C and possibly some Fortran, with annotations for parallelization based on the platform used. The report will contain the benchmarking and performance data of each algorithm on each platform, and an analysis regarding the potential advantages and pitfalls of each algorithm paired with each platform.

References:

https://www.princeton.edu/mudd/news/faq/topics/Non-Cooperative_Games_Nash.pdf
http://people.csail.mit.edu/costis/simplified.pdf
http://www.cs.wayne.edu/~dgrosu/pub/cse09.pdf
http://epubs.siam.org/doi/pdf/10.1137/0112033
https://www.cs.ubc.ca/~hutter/EARG.shtml/earg/papers04-05/computing-nash-eq.pdf

- Xeon Phi Implementation
    - Coprocessor
    - More general purpose
    - Less threads, slower
    - PCI Express connection
    - x86 native code

http://www.cs.wayne.edu/~dgrosu/pub/cse09.pdf: Parallel, N player games http://people.csail.mit.edu/costis/simplified.pdf: Proof of hardness of computing nash

https://www.princeton.edu/mudd/news/faq/topics/Non-Cooperative_Games_Nash.pdf: Proof of Existance Nash

http://epubs.siam.org/doi/pdf/10.1137/0112033: Lemke-Howson

https://www.cs.ubc.ca/~hutter/EARG.shtml/earg/papers04-05/computing-nash-eq.pdf: stanford alg, claims to be faster than Govindan-Wilson and Lemke-Howson