Computation of Equilibria of Finite Games and other Game Theoretical Computations
Group 20

Per Frost
Marcus Lång
Markus Thurlin
Christer Hedberg
Christopher Engelbrektsson

November 27, 2007

Functional requirements

The user shall be able to: calculate pure Nash equilibria. calculate Nash equilibria in the mixed extension (mixed Nash equilibria). calculate Correlated equilibria. calculate Subgame perfect equilibria. edit finite extensive form games (game trees). edit normal form games.

Non-functional requirements

Performance

It should not take more than 15 seconds to find Nash equilibria in the mixed extension of a normal form game with eight alternatives for each player.

Compatability

The program should work with all versions of Matlab equal to or newer than 7.0.
It should run on Windows XP Service Pack 2.

Reliability

The program should not crash or hang more than once per 1000 problems solved.

Licensing

All external libraries or executables used should as far as possible be licensed under LGPL, X11/MIT, UoI/NCSA or similar open source licenses.
Each respective author retains all rights to his source code.

Support

Help on how the program is used should be included. It must enable any person with some knowledge in computers to install and run the program.

Use cases

Use case: Calculation of Nash Equilibrium I

1. The user invokes the equilibrium calculation program from the console with a dataset representing a normal form game with a switch that indicates that a mixed Nash equilibrium is to be found.

2. The program returns a representation of the set of mixed Nash equilibria.

Use case: Calculation of Nash Equilibria II

1. The user is writing an AI program that operates in a simple environment with actors with different interests but needs a program to calculate Nash equilibria in order to determine what the AI is to do.

2. The user includes a header for the equilibrium calculation program in source file form which he intends to use.

3. The user makes use of the purpose built data structures to represent what he intends to have the program calculate.
4. The user invokes the equilibrium calculation function.
5. The user compiles his program.
6. The user runs his program and uses it to calculate equilibria.

Use case: Creation of game tree in matlab and calculation of subgame perfect equilibria and Nash equilibria

1. The user starts Matlab.
2. The user selects the Matlab plugin.
3. The game tree editing tool starts and the user constructs a game tree with it.
4. The user chooses that subgame perfect equilibria are to be calculated
5. The subgame perfect equilibria are displayed
6. The user chooses that the mixed Nash equilibria are to be calculated
7. The mixed Nash equilibria are to be displayed
8. The user compares the mixed Nash equilibria with the subgame perfect equilibria
9. The user exports those two as representations in the form of matrices in matlab
10. The user studies the matrices by other means with Matlab

Use case: Creation of game tree and normal form game outside of matlab and calculation of correlated equilibria

1. The user starts graphical editing program, which works as the plugin integrated into matlab.
2. The user creates a game tree in the graphical editing interface.
3. The user transforms the game tree into a corresponding normal form game.
4. The user edits the normal form game.
5. The user selects that the correlated equilibria are to be calculated.
6. The correlated equilibria are presented.
7. The user studies the correlated equilibria.