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

Per Frost
Marcus Lång
Markus Thurlin
Christer Hedberg
Christopher Engelbrektsson

1 The users of the program and the problem the system solves for them

Certainly it would be absurd to suggest that a person who is educated in mathematics and game theory would have any greater interest in a computer program to solve finite games, having long since lost interest in the analysis of particular finite games, and being able to transform any finite game into problems solvable by means of existing numerical analysis packages. Game theory is however useful not only to specialists, but also to non-specialists since there is a lack of intuitive software that they can use. This lack of software greatly hinders game theory's application to more trivial matters where it despite the insignificance of the problems would come to great use.

For these reasons it is intended that the system is to be used by persons who know of game theory, but do not really know how to apply it, and who would make use of it in their working environment, for example software engineers and people with a need to analyse situations that can be modelled by game theory but who either lack the resources to hire specialists or have been placed in positions in which they are expected to produce analyses of this type. With access to appropriate documentation such a person could reasonably easily learn, for example, how to construct game trees, assign values to coalitions, and make other simple representations of situations which he has need to analyse. Despite not knowing how to transform these representations into linear programming problems and in particular not knowing how to solve these resulting systems, these persons would only need to be able to interpret the solutions that the program itself outputs and then realise that they are valid and nontrivial to find.

2 The main uses of the system

The main uses of the system would be the construction and editing of representations of finite games, the calculation of subgame perfect equilibria, Nash equilibria, and the development of other computer programs that need to make use of optimal strategies in finite games. Provided that the documentation is sufficiently good and includes examples of modelling real world data, the system would probably not be (at least purposelessly) used to support bad decisions.

3 Usage narratives

3.1

Consider for a moment Bob, a software engineer who works in a small software company with a total of about 40 employees. In his line of work tacit collusion (a kind of untold agreement between companies concerning a certain market strategy) frequently occurs, but is not viable for the company to hire an pro-

fessional analyst to calculate the profits given situations such as "Company A lowers prices 5

When dealing with tacit collusion Nash equilibria is highly relevant. While he may sometimes be able to calculate the Nash equilibria he will often be faced with extensive games where he simply won't be able to apply the theory himself.

Bob has a program that can solve these kind of problems for him in an instant. He starts up the program, and inputs the data for his specific problem. Given that the data and the game is correct, the program then presents a solution to the problem. His company then gets a competitive advantage over other other companies who do not make use of these type of calculations.

3.2

A student named Mary is studying game theory wishes to find the solutions to a horrid but mathematically elementary problem and could by means of the software find equilibria, model the game; and even, due to the ease of finding equilibria use the system to explain empirical data. She could then, at a computer in school or at home, use the program to do exactly this by simply entering the relevant data of the problem. In this context the program is used in parallel with studying the related theory. She is now familiar with the program and can after the course use it for when needed.

3.3

Joe runs a small company and is in the need to evaluate a contract. In this early stage he wishes to get some insight without hiring a specialist. The system can then be used in this early phase of evaluation i.e. to get an idea of how much it will actually cost to fulfill the agreement as preparation for more detailed auction theoretical analysis by specialists. Mechanism design can also be applied to this so that Joe can try to achieve the desired outcome. With the aquired data he is then able to continue and base his actions on more knowledge.

4 The context/environment in which the system is to be usedon

The program will normally be used on modern office computers, due to the target user group being people in small to mid-sized companies with limited technology. However given a high performance environment it may be used to solve more extensive and complex problems.

The system will be used both, in the form of its visual frontend as a tool for analysing data, the editing of game trees, normal form games and as aid to the interpretation of equilibria, but will be insufficient to perform a complete analysis of any more involved problems, and will therefore by tightly integrated with numerical analysis software.

Although it would be nice the program will most definitely not be platform independent, for we wish to leverage the platforms that exist and to have their rather extensive userbases simply install it almost as if it were a plugin.

5 The scope of the system

The system will need to have support for the editing of finite games, the calculation of subgame perfect equilibria, Nash equilibria, have limited support for ccertain slightly exotic solution concepts, support graphical editing of game trees, but there will almost certainly be no time for the adding of auction theoretic tools, user defined solution concepts or any form of support for infinite games; which we summarize in the table below

Topic	In	Out
Constructing/editing representations of finite games	x	
Calculating subgame perfect equilibria	x	
Calculating Nash equilibria	x	
Limited support for some slightly exotic solution concepts	x	
Graphical editing of game trees, normal form games etc	x	
Integration with numerical analysis packages	x	
Auction theory		x
User defined solution concepts		x
Infinite games		x

6 The main factors that need to be taken in to account when designing and building the system

Due to the mathematical nature of the program its correctness is of great importance and a lot of effort should be directed to assuring it. While some understanding of the theory is necessary to use the software, it should be a goal to try and make it understandable for as many people as possible. It should be a design goal to integrate the program with Matlab. This serves to insure that even specialists will use features to construct game trees and process the resulting representations with whatever proprietary software they use.

7 Technologies and Risks

The system would be written in C++ and make minimal use of ad-hoc numerical algorithms and instead making use of mature libraries and software such as GMP, GSL and MPSolve. Each subprogram being essentially a reduction

between problems in game theory, the totality which is to be possible to string together either by means of pipe, by the planned frontend, or by a software developer using the modules as black-box solvers.

Possible risks could be using and ,as already mentioned, problems with connecting the different modules due to unfamiliarity with required third party software and libraries. Another possible risk is the the need for groupmembers to learn the required mathematics and the need to study relevant algorithms, which may lead, not necessarily to setbacks, but to that features that are particularly difficult to implement may go unimplemented.