Designing Strategic Games With Preestablished Nash Equilibrium Through Artificial Inference and Global Learning

Hime A. e Oliveira Jr.

Federal University of Rio de Janeiro, PEE/COPPE/UFRJ, Rio de Janeiro, Brazil

Corresponding: `hime@engineer.com`

Abstract

This work presents significant results obtained by the application of global optimization techniques to the design of finite, normal form games with mixed strategies. To that end, the Fuzzy ASA global optimization method is applied to several design examples of strategic games, demonstrating its effectiveness in obtaining payoff functions whose corresponding games present a previously established Nash equilibrium. In other words, the game designer becomes able to choose a convenient Nash equilibrium for a generic finite state strategic game and the proposed method computes payoff functions that will realize the desired equilibrium, making it possible for the players to reach the favorable conditions represented by the chosen equilibrium. Considering that game theory is a very significant approach for modeling interactions between competing agents, and Nash equilibrium represents a powerful solution concept, portraying situations in which joint strategies are optimal in the sense that players cannot benefit from individually modifying their current strategies provided that other players do not change their strategies as well, it is natural to infer that the proposed method may be very useful for strategists in general. In summary, it is a genuine instance of artificial inference of payoff functions after a process of global machine learning, applied to their numerical components.

Keywords:
Nash equilibria; game and mechanism design; simulated annealing; fuzzy ASA; artificial inference; global machine learning

1. Introduction

Initially, game theory aimed to study and model socio-economic phenomena and the interaction between agents (or players), whose actions drive the overall process - typically the decision of a specific player may modify the course of the whole game. Common assumptions are that players will monitor well-defined targets (modeled by payoff functions) and use their a priori expectations about the other players' possible decisions in order to maximize their individual "gains". Today, game theory has been applied to many fields, including communication protocol design [7], strategic management [1], control engineering [13] and mainly microeconomic theory and practice [14, 20].

Being an important analytical tool when studying interactions between players, in which the final results depend on the collective strategies chosen by agents, the idea of equilibrium in game theory is considered a good model for a stable outcome of a given game, working well as a solution concept [8]. Although there are many proposed equilibrium concepts in the literature, Nash equilibrium is considered the most important one and is certainly the most studied. A Nash equilibrium may be described as a set of choices such that (in a certain setting) each player's strategy is an optimal response to other players' strategic decisions [10, 11] – the concept catches a special type of steady state in which each player acts rationally and holds (hopefully) correct expectations with respect to the other agents' possible decisions.

Although obtaining Nash equilibria of complex normal form games is a nontrivial problem, several methods have been discovered in the last decades that are able to compute them, provided certain conditions are met [10, 11, 20]. Some algorithms for solving finite games are implemented in the Gambit package [12], which contains good software tools for the design and analysis of extensive and normal form games - the Gambit system was used to validate our results, described in this article.

The choice of the most effective approach for computing Nash equilibria of a given finite game depends on various conditions, including whether the immediate need is to find pure or mixed strategy equilibria, or finding only one or all equilibria. In this fashion, it is sensible to define a solution as a systematic description of the outcomes that may emerge out of a game [20] and in this paper only strategic (normal form) games are considered.

As mentioned above, the present work is aimed at introducing an optimization-based method of strategic game design and also demonstrating its effectiveness in obtaining payoff functions whose resulting games feature previously established (by the strategist) Nash equilibria. That is, the game designer becomes able to choose a convenient Nash equilibrium for a generic finite state, normal form strategic game and the new algorithm computes payoff functions that will realize the desired equilibrium, making it possible for players to reach the conditions represented by the chosen steady state.

The global optimization method used in this paper to find payoff functions corresponding to a prescribed Nash equilibrium of strategic games may be viewed as an evolution of the original Adaptive Simulated Annealing method [5], considering that despite its widespread utilization and excellent quality, it did not approach the global optima of specific cost functions, under tough conditions. Aiming to improve its performance, a fuzzy controller was added to the original implementation of ASA, and has been shown to work well in several scenarios, as explained in [18]. The resulting implementation is known as Fuzzy ASA, and makes use the concept of quenching in order to avoid sub-optimal attraction regions.

Presently it is widely known that the problem of finding Nash equilibria of finite strategic games admits a handful of alternative (sometimes partial) solutions, although computing such solutions is still a difficult task, and algorithms unable to compute several (or all) Nash equilibria can be not so useful for many practical tasks.

Although the classical problem of finding Nash equilibria of finite normal form games may be solved by finding fixed points of some specific functions [11], a very significant theoretical result relatively to the same topic states that the problem may be faced as a global optimization one [10, 11, 21]. The latter approach opens up the way for the synthesis of a large number of possible methods of solution for the problem at hand, considering the substantial number of effective optimization methods available in the literature, especially those using computational intelligence techniques [19]. Surprisingly, the same approach happens to be adequate to solve the inverse problem (studied in the present paper), that is, once chosen a Nash equilibrium, how to find at least one acceptable game featuring that equilibrium.

Therefore, it is possible to say that the problem to be tackled is an example of artificial inference (of payoff functions) by means of global machine learning (the optimization process aims to reach the deepest or global minimum of a certain numerical objective function by adapting numerical pa-

rameters, or "teaching" them how to change in order to get there). As a matter of fact, by assuming that "artificial inference" means "production of conclusions from given evidences by using computer programs, without direct human intervention" and "global learning" means "reaching the best values for parameters of a given model by using global optimization methods on special objective functions", the statement above makes total sense.

In the sequence the paper will briefly describe the Fuzzy ASA paradigm, its application to the underlying problem and the obtained results.

2. Fuzzy Adaptive Simulated Annealing

The global optimization method used in this paper is based on the ASA method [5, 18], having its roots on the very well-known simulated annealing paradigm, and featuring a substantial number of additional computational possibilities which explain its high level of performance. Among the most important improvements it is worth citing re-annealing, quenching and the large number of control parameters, the latter allowing the construction of internal automatic control "devices", able to drive the whole optimization process. The ASA system was constructed so that users are able to change virtually any subsystem with tiny programming effort. In this manner, it is possible to re-design the behavior of generation/acceptance processes, stopping criteria, or seed synthesis.

Re-annealing is the runtime re-scaling of annealing temperatures, adjusting generating probability distribution functions to each dimension in response to changing sensitivities. That is, if the objective function is not responsive whenever a given independent variable is altered, it may be sensible to extend the search interval amplitude in that dimension etc..

Quenching is related to the possibility of adjusting several structural parameters related to the default behavior of the "cooling" process and drive the evolution of the parametric temperatures, besides allowing the user or any automatic control system to alter the speed of convergence. This possibility is helpful in case of stagnation near suboptimal regions.

ASA was designed to find function minimizers in the interior of a pre-established hyperrectangle and generates candidate points componentwise. Also, the activation of quenching may help to improve the efficiency of the overall process, although there is the possibility of reaching prematurely suboptimal extrema. To overcome this obstacle, the controller of Fuzzy

ASA [16, 18] may be activated. Once this happens, the original ASA dynamics is faced as a MISO (Multiple Input Single Output) dynamical system and the additional code simply implements a feedback loop by sampling its output (current value of objective function) and actuating on its inputs (a subset of run-time adjustable parameters, related to the quenching process) according to a control method that imitates human behavior whenever subject to similar stimulus. Consequently, run-time control and already existing ASA mechanisms may drive temperature evolution, being capable of acting in case of premature convergence.

3. Motivation, significance and contribution of the paper

The problem of finding at least one equilibrium of given finite, normal form strategic games is very well-studied, presenting a large number of solution methods. However, the reverse problem of, given a generic strategic game structure and a desired (Nash) equilibrium, finding particular payoff functions that make that game to present the chosen equilibrium is not very well-known, although not new, and, to the best of our knowledge, no general solution method exists at present. To contribute in this direction, after defining precisely the concepts related to the cited problem, it will be established at the end of the section. Please, notice that the problem under analysis is being concretely stated in the present form for the first time in this article, and therefore the obtained numerical results are the only available for the time being, making it infeasible to compare them, at least quantitatively, with previous findings. Nevertheless, it is closely related to important issues addressed in mechanism design theory [2, 4, 3, 8, 9, 22].

3.1. Motivation and significance

According to Eric Maskin in [8], the concept of Nash equilibrium is John Nash's most important scientific legacy, mainly due to its role as a central solution concept (prediction of behavior) in applications of game theory. It is also stated that, despite presenting some limitations is certain settings, they (the limitations) are not very significant in the context of mechanism design. The theory of mechanism design is considered to be the "engineering" part of economic theory because a specific goal is established in advance (in this

paper a Nash equilibrium representing a favorable result) and a mechanism (finite, normal-form game) must be designed, attaining that target. Accordingly, the game must be found, given the desired result [8]. Also in [8] it is established that, in mechanism design, games can be endowed with convenient properties, as having a unique Nash equilibrium [9], for instance. The proposed method is compatible with this "flexibility of design", taking into account that during the global optimization process further constraints may be satisfied as well, in addition to the minimization of the objective function itself. Also, in [4] it is stated that in mechanism design the goals are determined first and, only after that, the system that implements those objectives is synthesized (if it exists, of course).

Thus, it is possible to infer that the Nash equilibrium concept is adequate for use with mechanism design when the aim is to reach certain patterns of behavior, considering its previous performance as a good predictor in strategic scenarios. On the other hand, the significance of finding strategic games presenting a previously chosen equilibrium is evident, considering the arguments above. The material contrasts the so called predictive economics with mechanism design, commenting that they act in different directions: the former tries to predict the result of a given game, and the latter finds the game that makes it feasible to obtain a given result. In this fashion, the number of possible applications is very large, including regulatory policies aiming to induce well-defined results in democratic contexts by means of incentives, for instance.

3.2. Finite strategic games

Basically, a strategic game is a model for the analysis of the interaction between players, also referred to as decision makers or agents. Each agent has specific alternatives to choose from a set of possible actions [20]. Interactions are described by means of functions that quantify the effect of actions of all participants and their gains or losses (payoff functions). More precisely, we have:

A finite strategic (normal form) game with pure strategies has three components:

- A finite set $\{1, 2, ..., N\}$ of players.

- A set of action sets S_i, each one containing the so-called pure strategies, so that each player has a finite group of possible moves.

$$S_i = \{s_{i_1}, ..., s_{i_{m_i}}\},$$

where m_i is the number of alternatives available to player i.
By a_{-i} it is meant the profile $(a_1, ..., a_{i-1}, a_{i+1}, ..., a_N)$ extracted from $(a_1, ..., a_{i-1}, a_i, a_{i+1}, ..., a_N)$ by leaving out the action of player i. In addition, the notation (a_i, a_{-i}) is used to represent the original profile of actions. Obviously, a (pure) strategy profile is defined as a finite ordered set $(a_1, ..., a_N)$, with $a_i \in S_i$.

- A set of N payoff functions

$$u_i : S \to \mathbb{R} \ , i \in \{1, 2, ..., N\}$$

$$S = S_1 \times S_2 \times ... \times S_N,$$

mapping profiles to real values.

3.3. Nash equilibrium - meaning and intuitive interpretation

A fundamental issue in the study of strategic games is the prediction of actions which will be taken by players in certain settings. Usually it is assumed that players will choose the most profitable actions, according to existing payoff functions. This optimal decision depends on the other players' actions and, when choosing an action, an agent must construct a belief about what other players' decisions will be. Hence, it seems sensible to investigate in what conditions and how can such a belief be obtained. Besides, one of the underlying premises for the analysis is that each player must be able to obtain degrees of belief from the previous experiences when playing the game, and this anterior knowledge should be sufficient to predict how other players will behave - this will make it feasible to endow the model with mixed strategies, which represent the improvement of the model with provision for an extra level of uncertainty modeling. Naturally players are not able to know precisely the actions opponents will choose, but previous observations allow

them to infer the frequency of these actions. Therefore, the natural path for solving this problem is based on each agent choosing its decision according to the belief of the other players' choices. These factors are contained in the common definition of a Nash (mixed) equilibrium, that can be informally formulated by saying that it is an action profile with the property that no player can inidividually profit from changing his/her present "position". In summary, when the game attains a Nash equilibrium, no decision maker would have advantages with a change. From another point of view, Nash equilibria may represent stable social configurations, meaning that if all individuals are adherent to it, none is willing to deviate from that satisfactory state of affairs.

Below, the notation used in the rest of the paper is defined - it is based on [21]. Δ_i represents the set of probability mass functions on S_i and Δ the Cartesian product $\prod_{i=1}^{N} \Delta_i$. Hence, $\Delta \subset \mathbb{R}^m (m = \sum_{i=1}^{N} m_i)$ and the elements of Δ_i have finite domains, being real valued functions defined on S_i, symbolically $p_i : S_i \to \mathbb{R}$. They also satisfy

$$\sum_{s_{ij} \in S_i} p_i(s_{ij}) = 1$$

with $p_i(s_{ij}) \geq 0$.

The symbol s_{ij} represents strategies $p_i \in \Delta_i$ with $p_i(s_{ij}) = 1$ - it is a expression for pure strategy profiles and (s_{ij}, p_{-i}) is the state in which player i chooses pure strategy s_{ij}.

In order for the concept of degrees of belief to be used, the idea of mixed strategies is essential. The probability mass funcions (p_i) are exactly the devices by means of which these strategies show up - players are capable of assigning weights to each possible action of each player by tuning them - this makes it possible to compute an aggregate payoff function, representing possible expected values of each game configuration. The function for player i is:

$$\mathbf{U}_i(p) \triangleq \sum_{s \in S} p(s) u_i(s) \tag{1}$$

with

$$p(s) = \prod_{i=1}^{N} p_i(s_i) \tag{2}$$

$$s = (s_1, s_2, \ldots, s_N) \tag{3}$$

Definition (Nash equiilbrium)

Given a finite normal form game, a mixed strategy profile $p^* = (p_1^*, p_2^*, \ldots, p_N^*) \in \Delta$ is a **Nash equilibrium** if

$$\mathbf{U}_i(p_i, p_{-i}^*) \leq \mathbf{U}_i(p^*)$$

$i = 1, \ldots, N$ and $p_i \in \Delta_i$.

This means that, when a strategy profile p^* is a Nash equilibrium, no player has alternative actions that may result in a higher gain than she or he receives by choosing p_i^*, assuming that remaining players keep their positions as p_j^* $(j \neq i)$.

3.4. Nash equilibrium as a global minimizer

According to a very useful result of R. McKelvey [10], it is possible to obtain Nash equilibria of finite normal form games by global minimization of a special real valued function. In order to apply that method, 3 auxiliary matrix functions have to be established, these being the components of the final objective function that will produce the Nash equilibria for the game under study. Namely:

$$x_{ij}(p) \triangleq \mathbf{U}_i(s_{ij}, p_{-i}) \tag{4}$$

$$z_{ij}(p) \triangleq x_{ij}(p) - \mathbf{U}_i(p) \tag{5}$$

$$g_{ij}(p) \triangleq max(z_{ij}(p), 0) \tag{6}$$

$p \in \Delta$.

The function used to obtain Nash equilibria (Liapunov function) is defined as

$$v(p) \triangleq \sum_{i=1}^{N} \sum_{j=1}^{m_i} (g_{ij}(p))^2, \ p \in \Delta. \tag{7}$$

Hence, p^* is a Nash equilibrium if and only if it is a global minimum of v in Δ (refs.[10, 11, 21]).

3.5. Statement of the problem

Although the usual problem is, given a complete game (payoffs included), to find some or all of its mixed strategy equilibria, this paper addresses the inverse problem of synthesizing payoff functions that realize a desired Nash equilibrium, with a preestablished generic structure (number of players and alternatives available to each player etc.) - this setting may certainly characterize practical situations that need proper treatment. In other words, while in the more frequent problem the game is the input and the equiilbria represent the output, by using the same theoretical basis it is possible to solve the inverse problem.

4. Proposed method

Let us focus on the proposition above: a mixed profile p is the global minimizer of function v if and only if it is the Nash equilibrium of the game that originated v, where the object to be found is the profile. What could happen if the roles are swapped, by choosing a fixed profile (desired Nash equlibrium) and evolving the payoff functions towards one global minimum of the same function? The natural answer, given implicitly in the statement of the proposition, is: provided it is possible to find payoffs which globally minimize v, the preestablished point is a Nash equilibrium for the game associated to those functions.

The moral is: the theorem of Richard McKelvey in [10] is bidirectional and, when used in the "reverse" direction, may be decisive in the solution of mechanism design problems. Figure 1 illustrates the general idea.

Furthermore, the relationship is many to many - each Nash equilibrium may

give rise to several games and each game may feature several Nash equilibria when using the theorem and the chosen global optimization method.
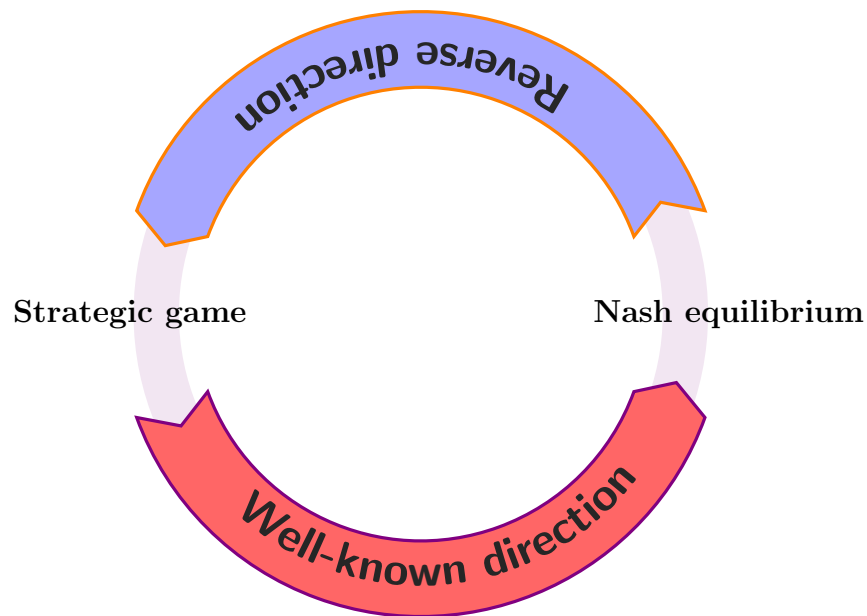
Figure 1: Two perspectives of Richard McKelvey's theorem

Hence, the underlying reasoning is analogous to the Bayesian one in statistics, according to which fixed parameters are viewed as random variables obeying a certain statistical distribution, and so on. Therefore, the global optimization process "designs" a new game in an autonomous way so that the desired steady state becomes one of its Nash equilibria. Besides, by design, the optimization method used in the solution allows the strategist to limit the values of each "incentive" present in the payoff functions in a particular numerical interval. In truth, even more constraints may be incorporated without too much effort, so as to make it easier to reach customized conditions for several types of problems - total amount available for use in a given strategic context, for instance.

It is also worth saying that other equilibria may occur (simultaneously to the desired one), associated to the game solving the original problem, but that is not actually a problem, taking into account that, typically, some degrees

of freedom are left, making it possible to adjust certain incentives/payoffs so as to drive players to convenient solutions.

Highlighting the crucial point, the payoff functions are viewed as parametric devices and their defining parameters evolve, driven by the global optimization process until one global minimum is reached, if any.

4.1. Approach for solution

As expected, besides the theoretical basis, the proposed algorithm depends on the minimization ability of Fuzzy ASA. After having defined precisely the relevant objects, now it is possible to state the following algorithm:

- For a given game structure, establish the dimension of the problem (number of variables to evolve)

- Build the objective function corresponding to (7)

- Set Fuzzy ASA parameters, according to problem dimension;

- Launch the minimization process [Step M];

- Examine whether the global minimum has been found;

- If not, go to [Step M] above, provided the maximal allowed number of iterations was not attained;

- In positive case, set the payoff functions and stop.

4.2. Numerical examples

The tests used to illustrate the proposed method were chosen so as to allow the reader to assess the efficacy of overall proposal in a simple and direct way. Besides, they are really able to represent real world problems and may help in the design process. The main difference, when compared to the literature, is that here the equilibrium is given and the payoff functions (the "soul" of the game) are calculated. Each Nash equilibrium may be interpreted as representing a sustainable and convenient configuration which each player is expected to reach, in order to achieve a desired effect. It is worth saying that in all cases Fuzzy ASA was able to find the respective

global minimizers, assuring that adequate payoff functions got found. For the sake of independent testing, the Gambit software package [12] has been used, and it can be freely downloaded from http://www.gambit-project.org. The final results are furnished in one of the formats accepted by the program Gambit, so that the interested reader may easily check the given information - for the first equilibria of examples 1 and 3 the payoff functions are also given as multidimensional matrices in the format used and explained in [19]. In order to facilitate the practical checking process, a separate ASCII file is provided to the reader so that no problems occur while using the Gambit specifications. So, it is advisable to use that file instead of trying to extract the information directly from the text, which may contain spurious control characters, for example.

It is very important to note that the numerical patterns of each Nash equilibrium in the examples have been chosen so that it would be probabilistically infeasible to find the associated strategic games by means of "brute force" methods in the direction [game → equilibrium], that is, the validation is done by using regular configurations of probabiilities, mainly in the mixed equilibria.

Example 1. This example uses a two player game, with 4 pure strategies available to each one. Four instances of Nash equilibria were employed, resulting in 4 different sets of payoff functions, namely:

- (0.1 , 0.3 , 0.5 , 0.1 , 0.333 , 0.333 , 0.333 , 0.001) [mixed]

- (0.1 , 0.2 , 0.3 , 0.4 , 0.4 , 0.3 , 0.2 , 0.1) [mixed]

- (0.0 , 0.5 , 0.0 , 0.5 , 0.0 , 0.6 , 0.0 , 0.4) [mixed]

- (0 , 1 , 0 , 0 , 0 , 0 , 0 , 1) [pure]

with 2 probability mass functions and 4 probabilities assigned to each one. Each payoff function has 16 entries, and the game features a total of 32 parameters to be adjusted

The Gambit specifications for the obtained games are, respectively:

NFG 1 R "4x4 coordination game" { "Player 1" "Player 2" } { 4 4 }

8.12560165770491 6.78430696539257 7.31010049366073 8.65425109483446
7.36972240286680 8.40966259264342 7.95841435246965 7.09390703804013
8.79838202013501 9.75273973364617 8.16957238636490 8.81593865766219
8.70577709895051 6.81502149058320 8.65036229607014 11.61361376974147
7.11609483783797 7.04971646187594 8.57461956579382 8.02765759028749
7.97211597658017 8.64462244333312 7.42924144644191 7.53347524192814
15.72239330238064 14.75617933282098 10.98911767737581 16.11875089775158
13.21260906875376 1.08415928886944 16.40851049730778 13.35604741462234

The corresponding matrix form is:

$$
\begin{bmatrix}
\begin{bmatrix}
8.12560165770491 & 8.79838202013501 & 7.11609483783797 & 15.72239330238064 \\
7.31010049366073 & 8.16957238636490 & 8.57461956579382 & 10.98911767737581 \\
7.36972240286680 & 8.70577709895051 & 7.97211597658017 & 13.21260906875376 \\
7.95841435246965 & 8.65036229607014 & 7.42924144644191 & 16.40851049730778
\end{bmatrix} \\
\begin{bmatrix}
6.78430696539257 & 9.75273973364617 & 7.04971646187594 & 14.75617933282098 \\
8.65425109483446 & 8.81593865766219 & 8.02765759028749 & 16.11875089775158 \\
8.40966259264342 & 6.81502149058320 & 8.64462244333312 & 1.08415928886944 \\
7.09390703804013 & 11.61361376974147 & 7.53347524192814 & 13.35604741462234
\end{bmatrix}
\end{bmatrix}
$$

NFG 1 R "4x4 coordination game" { "Player 1" "Player 2" } { 4 4 }

8.55646108407104 6.61083012586943 2.09290372612453 6.87679925981298
7.69511412155359 8.76127499315698 7.51816508518421 7.79070189605439
6.85097028031237 11.07323064993880 12.53387386864416 7.63382786249227
8.08121477248773 6.18757740358033 10.21743091929604 8.22686080177430
11.35648846675449 8.80600699604271 15.09421447187267 10.63591833595587
7.95767483631248 5.19320849374325 8.90536396433466 8.03839777204151
8.94815595062046 11.66000399981259 10.27822470945610 1.07177831534612
15.50043790541926 13.37032868872476 7.90420737923836 5.97412879877799

NFG 1 R "4x4 coordination game" { "Player 1" "Player 2" } { 4 4 }

4.35869196580868 3.78066385972927 3.83976660108770 5.34465162432900
9.67653208191883 14.68617501169030 8.53263593273009 10.80569831331021
4.07402398461863 8.10899430070610 8.30960307170967 9.34593338633606
0.96963981178754 1.70347723067538 7.76892862057447 8.73274777700484
6.34872540737566 8.35070087842524 5.17798849140231 1.48148600656647
2.02871338483991 10.14633352426310 3.06945892537969 14.99607660392897
12.55770270469755 1.26581788821816 7.55327872356642 3.02974615982961
7.93855527387914 8.32667121256707 8.36429015958482 15.04893481872175

NFG 1 R "4x4 coordination game" { "Player 1" "Player 2" } { 4 4 }

1.90370691332408 3.38385139742191 0.99631586919959 7.94852025600399
14.60145468704844 3.58213473283369 11.50571453105955 7.98712734084287
6.11592276274574 7.95620896697095 8.46305395197740 7.46410235422238
9.16143391530872 7.91446138833546 9.11710916203764 14.09214363453170
5.46211148552422 2.57309278050745 7.10055010117476 7.94610668291059
8.67337430098052 8.40681139489820 6.63113015513867 7.71439123492959
1.05756151198085 5.81290591302008 11.98675653132158 8.39772502092892
11.80125334801696 7.06275730069118 2.83903615956531 5.59789289824827

For this example it is advisable to choose the two alternatives "Compute as many Nash equilibria as possible" and "by solving a linear complementarity game" when using the Gambit package.

Example 2. Here the game structure features 3 players, with 3 alternatives each. Siamilarly to the preceding test, 3 Nash equilibria were used, again with 3 different sets of payoff functions, namely:

- (0 , 0 , 1 , 0 , 1 , 0 , 0 , 0 , 1) [pure]

- (0.65 , 0.35 , 0 , 0.8 , 0.2 , 0 , 0 , 0 , 1) [partially mixed]

- (0.999 , 0.001 , 0 , 0.5 , 0.3 , 0.2 , 0.85 , 0.03 , 0.12) [mixed]

with 3 probability mass functions and 3 probabilities assigned to each one. Each payoff function has 27 entries, and the game features a total of 81 parameters to be adjusted

The Gambit specifications for the obtained games are, respectively:

NFG 1 R "3 players with 3 alternatives each"

"Player 1" "Player 2" "Player 3" } { 3 3 3 }

1.36345891599414 1.16858068185864 1.29225651896388 2.17966585271679
2.00486713645631 2.09588555999147 2.11039742663884 1.82564772932150
2.48211941900597 2.16007267050984 2.85244253006460 2.77919232418201
3.08585576391490 3.80237894751942 3.67871636201125 3.80591132451746
4.46008947147559 4.16879390282802 1.50552185268723 4.77802378036095
4.32433531525713 4.93117017558921 5.08563371926272 3.89256376210155
4.65761814574796 5.26666332234965 5.75881009352541 6.41830149657877
6.21806596565421 6.47628260155340 6.57209293391821 8.73726411569690
6.90777746982877 4.01507269882438 10.25766839260028 5.19589737111824
7.25858309087220 4.77063346695330 8.13324711871070 8.22704993610752
8.37907986600110 8.55702474919437 8.76112962580940 6.18782022351024
9.16206597120781 9.31927392362835 9.83624779355680 11.10088353621432
9.79862336458683 5.75831714053956 1.81238396022934 1.44343330802902
16.16268704471777 10.66670158036774 11.13126587104876 10.05011856050844
11.60066328157840 11.57246315161944 5.81816089578675 2.68244282843182
4.39506161649868 12.28958083051261 12.23604902616678 12.05899155132227
12.61609046732656 13.05292351056636 12.75316411560529 4.82598741531985
21.74089277980387 13.71264489808863 17.10088193263447 14.17517388738893
14.35568505551015 14.47921123094396 20.00364343975601 14.75697554696659
15.09644696648075 25.12556304383047 15.03181380137126 14.77877815741613
16.16427083668837

NFG 1 R "3 players with 3 alternatives each"

{ "Player 1" "Player 2" "Player 3"} { 3 3 3 }

3.81255431784455 3.86459943943599 0.90178652006425 9.21429866485542
10.35337481680581 8.96506754986564 10.31464790602485 4.74367209082389
13.56047480790961 15.04187116191155 4.51023424585922 10.69626583124843

12.87226668810991 11.39428657816939 0.43069574145777 15.82111664712050
3.55494158178712 0.98476106304402 0.10174836891656 1.69142684923271
6.88157292920098 2.03171091136950 10.96528043148307 8.56503684507226
12.22005337345855 6.34507982981415 10.66697636738847 6.23358855821085
1.57290019410349 6.76247930040646 13.17319841459954 11.35704149897789
13.91511951556889 15.00602137737694 11.19094656639152 1.56261304639149
14.15280335627073 13.96304455960912 2.46240583116461 15.05980892679728
9.05203035937897 14.91257956322229 11.98893453289638 1.90658114082116
2.11232100125991 8.78430696942486 9.06729810311297 14.61964681256636
12.75805239241750 14.79900625202411 10.62148378436572 9.59155770739561
13.42298550921913 3.17723874451773 9.13667288222516 5.88604309221813
12.37844566976981 10.38771718511320 9.67732053160474 16.02961421183255
0.99022406172573 4.33267359668940 13.72141544136324 13.48238911671322
5.42683371801427 14.37846442074516 8.47821233822193 10.53013776207730
9.50399429676473 9.14240192303081 14.49571033082774 8.31297320498839
0.19801721361240 5.98056250003807 16.24587916573304 9.91712734093317
6.73487340186752 4.78558455528777 11.29319526012159 4.42424217771342
10.41486740789210

NFG 1 R "3 players with 3 alternatives each"

{ "Player 1" "Player 2" "Player 3" } { 3 3 3 }

0.41907224805908 2.54668798691978 3.69143764853352 1.49401203300557
1.23632919587067 5.18236095876369 1.37027203154710 2.61451879654539
5.34124191647484 3.87020781033289 2.48817157590621 1.11678097046411
1.58344115692282 4.81091638578868 1.13777631584919 1.67024343677765
0.52974146569368 3.83094739737172 2.09772362947140 2.53228470347876
5.73566700239931 5.10974729394207 1.10496228207064 2.58616760998074
0.78694485008052 2.77374219561731 4.42031294929609 4.69099355345739
4.25997265409763 2.63293683335051 3.68697844385232 1.47556890473259
4.67996572329187 2.80277329449233 3.55409479650421 2.73494930140644
1.32466531657703 5.10761567862753 4.97733193826110 3.35541485686289
3.37526971605859 2.45700925290406 4.75693049211413 1.27383439376930
0.04947877839446 0.89056940082427 0.59002193001851 2.58651191643623
6.05407656963823 1.59753937334298 6.43993902445402 4.39205681469938
5.43810595848940 3.51390521346355 3.48756764232238 2.03059885056539

2.84730450064386 0.99891702813551 1.11828155870289 3.03268246888011
4.21162073127774 3.8457890683483 1.51512211173076 5.73724822722894
2.20583395869105 3.45530840374112 2.24472878172965 2.64215165789108
2.65897245925045 1.22011516424018 4.09189848189151 0.81940931416885
6.41557497587752 3.04848468463415 4.34143525572801 0.37741904334620
3.64210331258567 2.45527790196677 6.35370814927986 4.53721415752022
0.05791437287359

For this example it is advisable to choose the two alternatives "Compute as many Nash equilibria as possible" and "by solving systems of polynomial equations" when using Gambit.

Example 3. Here the game structure features 3 players, with 2 alternatives each. Three mixed Nash equilibria were chosen and 3 different strategic games have to be found. The equilibria are:

- (0.55 , 0.45 , 0.9 , 0.1 , 1 , 0) [partially mixed]

- (0.5 , 0.5 , 0.3 , 0.7 , 0.155 , 0.845) [mixed]

- (0.1 , 0.9 , 0.75 , 0.25 , 0.999 , 0.001) [mixed]

with 3 probability mass functions and 2 probabilities assigned to each one. Each payoff function has 8 entries, and the game features a total of 24 parameters to be adjusted.

The Gambit specifications for the obtained games are, respectively:

NFG 1 R "2x2x2 Example" { "Player 1" "Player 2" "Player 3" } { 2 2 2 }

1.68474352284291 0.69715937980155 2.18646832664289 1.38897602537620
2.61918532434923 2.15098819909268 0.95036193998976 1.36147229906010
0.27594734298669 3.61226985267416 1.80724724980295 1.37446463280907
2.01346266092545 3.70587669079759 1.66347501004608 1.49993875644457
0.59464815068307 1.28379671725897 3.17083285321985 3.80474993003341

2.96129542830236 1.53751977696334 5.77432392748911 5.10577609471429

The corresponding matrix form is:

$$
\begin{bmatrix}
\begin{bmatrix} \begin{bmatrix} 1.68474352284291 & 2.01346266092545 \\ 0.95036193998976 & 3.17083285321985 \end{bmatrix} & \begin{bmatrix} 1.38897602537620 & 1.49993875644457 \\ 3.61226985267416 & 1.53751977696334 \end{bmatrix} \end{bmatrix} \\
\begin{bmatrix} \begin{bmatrix} 0.69715937980155 & 3.70587669079759 \\ 1.36147229906010 & 3.80474993003341 \end{bmatrix} & \begin{bmatrix} 2.61918532434923 & 0.59464815068307 \\ 1.80724724980295 & 5.77432392748911 \end{bmatrix} \end{bmatrix} \\
\begin{bmatrix} \begin{bmatrix} 2.18646832664289 & 1.66347501004608 \\ 0.27594734298669 & 2.96129542830236 \end{bmatrix} & \begin{bmatrix} 2.15098819909268 & 1.28379671725897 \\ 1.37446463280907 & 5.10577609471429 \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

NFG 1 R "2x2x2 Example" { "Player 1" "Player 2" "Player 3" } { 2 2 2 }

1.72469927982884 1.19670164168805 0.12826309082048 0.18546853400222
1.95593794377773 3.50074861795324 1.93148667739596 2.67598639617612
2.33811245711324 2.92843281425482 3.97398261913416 2.36006193087863
3.16547556790241 3.71383380871931 2.60986961699400 2.04811722578377
8.56068580853486 4.43725113766387 5.28539697614088 4.92181957420406
2.87553445706927 5.70239780158931 6.71117832475187 0.35773582528001

NFG 1 R "2x2x2 Example" { "Player 1" "Player 2" "Player 3" } { 2 2 2 }

1.46406656096406 0.45437915671140 0.03301717940687 1.35028081639669
3.08155696002606 2.32589988718928 1.37478270680648 1.86245148018433
4.77565078136680 1.72537987038170 2.92362506028257 3.68408067207654
6.51016888467915 1.56891977015656 4.28043896521942 2.61297298246851
0.01730807221530 0.61700888122216 4.27759603277916 4.90773326419176
2.03702278843193 6.73851204040896 1.12441859317888 7.69923848451411

For this example it is also advisable to use the two alternatives "Compute as many Nash equilibria as possible" and "by solving systems of polynomial equations".

5. Interpretation of results and recommendations for independent testing

Once more it is worth to note that the problem and solution presented in this work are new and, as such, can not be compared to previous results, considering they simply do not exist. Despite this, the furnished Gambt specifications make it possible to easily verify that the synthesized games have the initially given equiibria. Another important fact is that, in all cases, the final objective function value was under 10^{-16}, that is, the global minimum was reached in practice.

Let us now comment the examples:

- In Example 1 it is presented a scenario with two actors deciding about four distinct initiatives - this could be 2 food industries regulated by a central management unit which is trying to establish the most convenient dynamics so that the proportion for production of 4 different products of each one be given by each one of the 4 proposed Nash equilibria, for instance. In this fashion, with the help of the proposed algorithm it is possible to open this possibility by positioning the preestablished Nash equilibrium into the overall set of equilibria of the resulting game.

- Example 2 represents a context with 3 agents and 3 distinct actions available to each one - this setting may correspond to 3 firms forming a productive cluster and controlled by a central office which wishes to reach a certain production profile in the overall site so that the proportion of the production level of 9 different products (3 x 3) be given by each one of the proposed Nash equilibria. The obtained payoff functions contain the "rewards" capable of making a certain equilibrium attractive and feasible.

- Example 3 portrays again the interaction of 3 players, this time each one has 2 alternatives to choose from. As in the previous examples, the 3 chosen profiles were chosen to demonstrate the diversity of situations the proposed algorithm can handle and, moreover, the regular numeric patterns help to evidence the precision and efficacy of the reverse design.

6. Conclusion

In this paper it was shown that it is possible, with the help of a general purpose evolutionary global optimization method, to synthesize finite strategic games with given structure and Nash equilibrium in an automatic way. The new results were obtained thanks to an important and very useful proposition and the effectiveness of the Fuzzy ASA method relatively to its ability to find the correct structure of the games in all test cases, by finding the global minimizer of each cost function.

The usefulness of the presented results resides mainly in the simplicity and the built-in possibility of limiting each individual payoff from the beginning, given that even the original ASA method allows users to define the domain corresponding to each dimension. Besides, the spectrum of possible applications is very large.

In addition, the problem under study illustrates a kind of artificial inference of payoff functions by means of global machine learning (reaching the deepest or global minimum of a certain numerical objective function by adapting/driving its arguments towards the global minimum).

Final words

Worse than to state something without scientific proof may be denying it without scientific proof.

—— H.A.O.J.

References

[1] K. Chatterjee, W. Samuelson (Editors), Game Theory and Business Applications, Springer, New York, 2014.

[2] D. Diamantaras, A Toolbox for Economic Design, Palgrave Macmillan, New York, 2009.

[3] L. Hurwicz, S. Reiter, Designing Economic Mechanisms, CUP, New York, 2006.

[4] M. Hurwicz, Mechanism Design, http://leonidhurwicz.org/mechanism-design/ , accessed in Apr 23 2019.

[5] L. Ingber, Adaptive simulated annealing (ASA): Lessons learned, Control and Cybernetics, 25 1 (1996) 33–54.

[6] L. Ingber, B. Rosen, Genetic algorithms and very fast simulated re-annealing: a comparison, Mathematical Computer Modelling, 16 11 (1992) 87–100.

[7] S. Kim, Game theory applications in network design, IGI Global, Hershey, 2014.

[8] E. Maskin, Nash equilibrium and mechanism design, Games and Economic Behavior, 71 (2011) 9-11.

[9] E. Maskin., T. Sjöström, T., 2002. Implementation theory. In: Arrow, K., Sen, A., Suzumura, K. (Eds.), Handbook of Social Choice Theory, vol. I. North-Holland, Amsterdam, pp. 237-288.

[10] R. D. McKelvey, A Liapunov function for Nash equilibria, Technical Report, California Institute of Technology, 1991.

[11] R. D. McKelvey, A. McLennan, Computation of equilibria in finite games, in: H.M. Amman, D.A. Kendrick, J. Rust (Eds.), Handbook of Computational Economics, Handbooks in Economics (13), 1, North–Holland, Amsterdam (1996) 87–142.

[12] R. D. McKelvey, A. McLennan, T. L. Turocy, Gambit: Software tools for game theory, available at http://www.gambit-project.org, 2016.

[13] T.L.Molloy, J.J. Ford, T. Perez, Inverse Noncooperative Dynamic Games, IFAC PapersOnLine 50-1 (2017) 11788-11793.

[14] Y. Narahari, Game Theory and Mechanism Design, World Scientific, Singapore, 2014.

[15] J. F. Nash, Noncooperative games, Ann.Math., 54 (1951) 289–295.

[16] H. A. e Oliveira Jr., A. Petraglia, Global optimization using dimensional jumping and fuzzy adaptive simulated annealing, Applied Soft Computing, 11 (2011) 4175–4182.

[17] H. A. e Oliveira Jr., A. Petraglia, Global Optimization Using Space-Filling Curves and Measure-Preserving Transformations, in: A. Gaspar-Cunha et al. (Eds.), Soft Computing in Industrial Applications, Springer-Verlag, Berlin–Heidelberg (2011) 121–130.

[18] H. A. e Oliveira Jr., L. Ingber, A. Petraglia, M.R. Petraglia and M.A.S. Machado, Stochastic Global Optimization and Its Applications with Fuzzy Adaptive Simulated Annealing, Springer–Verlag, Berlin–Heidelberg, 2012.

[19] H. A. e Oliveira Jr., Evolutionary Global Optimization, Manifolds and Applications, Springer-Verlag, Cham, Switzerland, 2016.

[20] M. J. Osborne, A. Rubinstein, A Course in Game Theory, MIT Press, Cambridge, MA, 1994.

[21] N. G. Pavlidis, K.E. Parsopoulos and M.N. Vrahatis, Computing Nash equilibria through computational intelligence methods, Journal of Computational and Applied Mathematics, 175 (2005) 113-136.

[22] Y. Pei, Algorithmic Mechanism Design of Evolutionary Computation, Computational Intelligence and Neuroscience, (2015) 591954.