

# Synthesis of strategic games with multiple Nash equilibria - a global optimization approach

Hime A. e Oliveira Jr.\*

*Federal University of Rio de Janeiro, Program of Electrical Engineering, COPPE/UFRJ,  
Rio de Janeiro, Brazil*

---

## Abstract

This paper presents an extension of the results obtained in previous work concerning the application of global optimization techniques to the design of finite strategic games with mixed strategies. In that publication the Fuzzy ASA global optimization method was applied to many examples of synthesis of strategic games with one previously specified Nash equilibrium, evidencing its ability in finding payoff functions whose respective games present those equilibria, possibly among others. That is to say, it was shown it is possible to establish in advance a Nash equilibrium for a generic finite state strategic game and to compute payoff functions that will make it feasible to reach the chosen equilibrium, allowing players to converge to the desired profile, considering that it is an equilibrium of the game as well. Going beyond this state of affairs, the present article shows that it is possible to "impose" multiple Nash equilibria to finite strategic games by following the same reasoning as before, but with a slight change: using the same fundamental theorem of Richard D. McKelvey, modifying the original prescribed objective function and globally minimizing it. The proposed method, in principle, is able to find payoff functions that result in games featuring an arbitrary number of Nash equilibria, paving the way to a substantial number of potential applications.

### *Keywords:*

Nash equilibria, Strategic game design, Simulated annealing, Fuzzy ASA, Global optimization

---

\*Corresponding author

*Email address:* [hime@engineer.com](mailto:hime@engineer.com) (Hime A. e Oliveira Jr.)

*Preprint submitted to Preprints.org*

*November 24, 2019*

## 1. Introduction

Although game theory has been traditionally used in the investigation of economic systems, in recent years its importance is being evidenced in several fields, like network protocol design and analysis [7], computational resources allocation and several other phenomena [1]. This trend may be explained by the fact that game theory is all about modelling interactions among agents and respective results, and this type of context can be frequently found almost everywhere and in several kinds of scenarios. Often the decision of a single player may affect drastically the result of a given game. In order to model players' benefits (or losses) when acting during the game, payoff functions are used, and to cope with uncertainty, mixed strategies are employed, representing the belief opponents have with respect to possible moves of other participants. In this fashion, and as a significant analytical tool, it was endowed with the concept of Nash equilibrium, which has been and is considered a good model for stable outcomes of strategic games. So, despite the existence of other equilibrium concepts in the literature, John Nash's one is typically considered the most interesting and/or useful.

Being a condition that may be described as a set of strategies representing optimal responses to other players' strategic decisions [10, 11], it catches a special type of stable condition in which each player acts selfishly, searching for the maximum gain and holding supposedly right expectations with respect to other agents' decisions.

As expected, computing Nash equilibria of nontrivial finite normal form games is often a hard problem and, over the years, many powerful methods have been developed to address this issue [10, 11, 21]. In general, the most adequate approach for computing Nash equilibria of strategic games depends on several factors, as whether the the objective is to obtain pure or mixed strategy equilibria, or it is necessary to find all existing equilibria.

The present article intends to introduce an optimization-based method of strategic game design aimed at obtaining payoff functions whose resulting games have previously established multiple Nash equilibria. That is to say, it becomes possible to specify in advance a set of Nash equilibria for a generic finite state, normal form game and the proposed algorithm computes payoff functions that will realize the desired equilibria - this work may be viewed as the continuation and extension of [20], that solved the problem for a single Nash equilibrium.

Although any effective evolutionary global optimization method could

be used to find the desired payoff functions, the Fuzzy Adaptive Simulated Annealing method [5] is again the chosen algorithm, considering its amazing performance and flexibility in preliminary simulations.

Among the successful initiatives for finding Nash equilibria of finite games are those using global optimization techniques [10, 11, 22]. One of them, due to R.D. McKelvey [10], may be used not only in the "given game => Nash equilibria" direction, but also in the opposite one: "given Nash equilibria => new game" - this viewpoint was explored in [20] and in this paper as well, demonstrating the large potential of the original result.

That point of view makes it possible to synthesize games with not only one, but several previously established Nash equilibria. This possibility is due to the way optimization-based design takes place - every design objectives and constraints must be translated into parametric functions that, when optimized, conduct to the optimal result.

Accordingly, in order to impose several equilibria to one single strategic game, it is sufficient to insert additional terms to the overall cost function, each one corresponding to a distinct Nash equilibrium - this approach was used to solve the inverse problem in [20], but using only one term, considering that there was just one given equilibrium, and the statement of the original theorem is compatible with this context. Here, we will show the reasons why it is possible to obtain even more, with the only inconvenient that objective functions get more and more complex, needing very powerful GO methods to find their global minima.

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

## 2. The global optimization method

The global optimization method used in this paper is Fuzzy ASA [18], based on the simulated annealing paradigm and introducing a large set of dynamic and static control possibilities, which contribute for its good performance. Among its most significant features it is possible to cite re-annealing, quenching and several parameters that make it possible to change the behavior of the optimization process even in runtime, allowing the construction of control mechanisms aimed at improving the convergence process. It was designed so that users have the possibility to change the dynamics of the simulations, driving optimization sessions with enormous freedom. Other simple but very useful features of the original ASA method include the ability of adjusting

each dimension of hyperrectangular domains and generating candidates with full control of each parametric dimension. Although the use of quenching may provoke the convergence to suboptimal regions, the Fuzzy ASA controller [16, 18] usually is effective in avoiding such an inadequate behavior, making it possible to reach global minima. Actually, the original ASA structure is viewed as a dynamical system, and the added code implements a feedback loop by sampling the current value of the cost function, changing its inputs according to a control law that simulates a sort of rational behavior.

In summary, fuzzy control and built-in ASA mechanisms conduct temperature profiles, avoiding premature convergence.

### **3. Contribution of the present paper, motivation and related definitions**

Although the problem of finding Nash equilibria of finite strategic games received a great deal of attention during the last decades and many solutions nowadays exist, the inverse problem of obtaining a complete set of payoff functions that realize one desired Nash equilibrium, with preestablished number of agents and individual strategies is new, having already some proposals for finding solutions, obtained by means of evolutionary techniques [20]. The present work aims to go further, also using stochastic global optimization to obtain games with multiple and predefined equilibria. To that end, some associated concepts must be precisely established.

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, 13, 14].

#### *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 in 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. Besides, 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.

Therefore it is possible to infer that the Nash equilibrium concept is adequate for use with mechanism design when the aim is to reach certain types of behavior, considering previous performance as an adequate 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 normal form games*

In principle, finite, normal form games are models for interactions between agents. Each agent owns a number of alternatives to choose from a set of possible decisions. The benefits or gain a given player may receive when taking initiatives are established by means of payoff functions, quantifying the effect of actions of all decision makers.

A finite normal form game with pure strategies is formed by:

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

- The action sets  $S_i$ , containing pure strategies for each player - a finite group of alternatives.

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

being  $m_i$  the number of actions available to agent  $i$ .

By  $a_{-i}$  is denoted the strategic profile  $(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_N)$  obtained from  $(a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_N)$  by excluding the action of player  $i$ . Besides, the notation  $(a_i, a_{-i})$  represents the original profile of actions.

- $N$  payoff functions

$$u_i : S \rightarrow \mathbb{R}, \quad i \in \{1, 2, \dots, N\}$$

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

associating real values to profiles.

### 3.3. Nash equilibrium - definition and some digressions

The motivation for the study of strategic games is strongly related to the prediction of decisions to be made by players in various situations. Almost always it is accepted that players will follow the most profitable path, the optimal decisions depending on the other players' actions and, when deciding, agents have to form a set of possibilities about the likelihood of other players' decisions. Therefore, it seems reasonable to investigate a method by means of which this can be made. Among the underlying premises for the issue at hand is that each participant should estimate degrees of possibility from past experiences, and this previous information should be sufficient to predict how other players will choose their future actions - this will endow the game with mixed strategies, representing a refinement and creating an additional dimension of uncertainty handling. Although it is not possible to know exactly what will be the decisions of the players, past observations may serve as inputs to estimators that should be able to construct the desired distributions. The definition of Nash (mixed) equilibrium includes all these

factors, and it is possible to say that it is a decision profile such that players cannot individually benefit from changing their present state - when a game reaches a Nash equilibrium, no agent would have advantages with unilateral changes. Under another perspective, Nash equilibria can be said to represent stable states, configuring a kind of consensus among group members so that nobody wishes to abandon the present setting. Some definitions used in the rest of the paper follow:

- $\Delta_i$  is the set of probability mass functions on  $S_i$
- $\Delta \triangleq \prod_{i=1}^N \Delta_i$  ,  $\Delta \subset \mathbb{R}^m$  ( $m = \sum_{i=1}^N m_i$ )
- The elements of  $\Delta_i$  are  $p_i : S_i \rightarrow \mathbb{R}$  ,  
satisfying  $\sum_{s_{ij} \in S_i} p_i(s_{ij}) = 1$  , with  $p_i(s_{ij}) \geq 0$ .
- $s_{ij}$  denotes a strategy  $p_i \in \Delta_i$  for which  $p_i(s_{ij}) = 1$  (pure strategy profile)
- $(s_{ij}, p_{-i})$  is the global profile in which the agent  $i$  chooses pure strategy  $s_{ij}$ .

The concept of mixed strategies is fundamental for the practical quantitative use of the perception of players with respect to the behavior of opponents. In this direction, probability mass functions ( $p_i$ ) are the devices by means of which this information is made effective - players are able to associate importances to each possible action of each player. By using them it is possible to compute expected payoffs for each agent. The corresponding formula for player  $i$  is:

$$U_i(p) \triangleq \sum_{s \in S} p(s) u_i(s) \quad (1)$$

for

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

$$s = (s_1, s_2, \dots, s_N) \quad (3)$$

A collective mixed strategy profile  $p^* = (p_1^*, p_2^*, \dots, p_N^*) \in \Delta$  is a Nash equilibrium if

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

for all  $i = 1, \dots, N$  and  $p_i \in \Delta_i$ .

Accordingly, for a Nash equilibrium  $p^*$ , no player has alternative actions that may result in higher gains than received by  $p_i^*$ , provided the others fix their positions in  $p_j^*$  ( $j \neq i$ ).

#### 3.4. The key proposition

As cited above, an amazing result due to McKelvey [10], relating global minimization of a specific function and Nash equilibria of the corresponding strategic game, made it possible to solve nontrivial finite normal form games by finding even certain unknown equilibria by means of evolutionary global minimization techniques [19]. The main cost function is founded on three matrix functions, defined below. Namely:

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

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

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

where  $p \in \Delta$ .

The final cost function is

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

The original statement and use of the proposition may be expressed as:

$p^* \in \Delta$  is a Nash equilibrium if and only if it is a global minimum of function  $v$  in  $\Delta$  (refs.[10, 11, 22]).

### 3.5. *The problem of attributing multiple Nash equilibria to a single game*

Given a preestablished generic strategic game structure (number of players and alternatives available to each player), the problem of synthesizing payoff functions which realize one desired Nash equilibrium has been successfully addressed in [20]. However, in many practical settings it may be convenient to impose several mixed equilibria upon a single game in order to model real world scenarios. Hence, a proposal for attaining that goal follows.

## 4. Proposed method

Considering the previous proposition, relating a mixed profile which is a global minimizer of function  $v$  to a Nash equilibrium of the game that originated  $v$  (where the object to be found is a mixed profile), the problem to be solved now is: departing from a finite set of mixed profiles, would it be possible to find payoff functions whose associated game has those profiles as its Nash equilibria? As noted above, if there is only one Nash equilibrium to be treated, the answer is affirmative, as found in [20] - it is also tacitly conveyed by the original proposition, as highlighted in the just cited work.

But another subtle interpretation of the theorem is worth noting: by constructing not one, but many " $v$  functions", one for each desired Nash equilibrium, and summing them, one obtains a new "collective" cost function  $V$  - it happens that a global minimizer for  $V$  is also a global minimizer for each individual  $v$ , making all of them Nash equilibria for the same game, specified by the optimal parameters representing the payoff functions. Hence, as in so many other optimization-based design tasks, the global optimization process produces solutions in an automatic way. As before, the optimization method used gives the designer the possibility of limiting the values of each component of payoff functions.

As expected, additional numerical conditions may be ensured by means of constraints so as to customize solutions for several types of problems.

### 4.1. *Approach for solution*

It is worth noting that, although the theoretical foundations of the proposed method are sound, its successful practical implementation totally depends on

the good quality of the available global minimization apparatus. That said, the algorithm is now described (using Fuzzy ASA):

- Departing from the chosen game structure, compute the dimension of the configuration space;
- Find, for each Nash equilibrium, the function corresponding to (7);
- Construct the expression of the overall objective function by summing all functions obtained in the previous step;
- Adjust Fuzzy ASA parameters;
- Start a new minimization session [Step N];
- Global minimum found or maximum allowed number of iterations reached ?
- If not, go to [Step N] above;
- In positive case, obtain the payoff functions and finish.

#### *4.2. Numerical simulations*

In order to illustrate the efficacy of the proposed paradigm, a small but representative group of tests was selected and will be used here to obtain numerical results. Also, the presented tests will serve to demonstrate the power of the optimization-based design as well.

Comparing with the related literature it is possible to observe that in this paper (as in [20]) the opposite direction is taken, that is, the equilibria are given and a possible game is obtained.

In terms of performance, in all tests Fuzzy ASA was able to reach global minimizers, furnishing the guarantee that proper payoff functions (and respective games) were found. This is so despite the fact that each cost function evaluation during the optimization process demands the creation and assessment of a different game - even so, most synthesis sessions have demonstrated low demand of computational resources.

As usual, the Gambit software package [12] (<http://www.gambit-project.org>) was (and may be) used to compute sets of equilibria for games produced

by the new algorithm, although in a few cases Gambit was not able to find all equilibria. Specifications compatible with Gambit are provided, so that the given information may be easily confirmed by means of an independent tool. To facilitate the checking procedure, an ASCII file is furnished so as to minimize possible problems while copying parts of the body of the text and using them as input to the Gambit system. This is so because the file containing the text may contain control characters, for example, what can cause the program to reject the descriptions without an apparent reason.

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  $\rightarrow$  equilibrium], that is, the validation is done by using regular configurations of probabilities, mainly in the mixed equilibria.

**Example 1.** This test uses the structure of a generic two player game, each one having 4 pure strategies. Five groups of simultaneous Nash equilibria were employed, each one giving rise to 2 different sets of payoff functions, as shown below:

With two simultaneous equilibria

(Search interval for each parameter = [0,6] and CF final value = 0) :

- ( 1.0 , 0.0 , 0.0 , 0.0 , 0.0 , 1.0 , 0.0 , 0.0 ) [pure]
- ( 0.1 , 0.2 , 0.3 , 0.4 , 0.4 , 0.3 , 0.2 , 0.1 ) [mixed]

With three simultaneous equilibria

(Search interval for each parameter = [0,6] and CF final value = 0) :

- ( 1.0 , 0.0 , 0.0 , 0.0 , 0.0 , 1.0 , 0.0 , 0.0 ) [pure]
- ( 0.1 , 0.2 , 0.3 , 0.4 , 0.4 , 0.3 , 0.2 , 0.1 ) [mixed]
- ( 0.0 , 0.5 , 0.5 , 0.0 , 0.0 , 0.0 , 1.0 , 0.0 ) [mixed]

With three simultaneous equilibria

(Search interval for each parameter = [0,6] and CF final value = 7.46069825961309788E-18) :

- ( 1.0 , 0.0 , 0.0 , 0.0 , 0.0 , 1.0 , 0.0 , 0.0 ) [pure]
- ( 0.1 , 0.2 , 0.3 , 0.4 , 0.4 , 0.3 , 0.2 , 0.1 ) [mixed]
- ( 0.0 , 0.5 , 0.5 , 0.0 , 0.0 , 0.5 , 0.5 , 0.0 ) [mixed]

With four simultaneous equilibria

(Search interval for each parameter = [0,2] and CF final value = 2.90248E-16)

:

- ( 0.0 , 2.0/3.0 , 0.0 , 1.0/3.0 , 0.25 , 0.0 , 0.0 , 0.75 ) [mixed]
- ( 0.2 , 0.4 , 0.3 , 0.1 , 0.1 , 0.1 , 0.1 , 0.7 ) [mixed]
- ( 0.0 , 0.1 , 0.9 , 0.0 , 0.01 , 0.0 , 0.99 , 0.0 ) [mixed]
- ( 1.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 1.0 ) [pure]

With five simultaneous equilibria

(Search interval for each parameter = [0,2] and CF final value = 2.123774890853668789e-16) :

- ( 0.0 , 2.0/3.0 , 0.0 , 1.0/3.0 , 0.25 , 0.0 , 0.0 , 0.75 ) [mixed]
- ( 0.2 , 0.4 , 0.3 , 0.1 , 0.1 , 0.1 , 0.1 , 0.7 ) [mixed]
- ( 0.0 , 0.1 , 0.9 , 0.0 , 0.01 , 0.0 , 0.99 , 0.0 ) [mixed]
- ( 1.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 1.0 ) [pure]
- ( 0.7 , 0.0 , 0.3 , 0.0 , 0.0 , 0.3 , 0.0 , 0.7 ) [mixed]

As expected, each global equilibrium has two probability mass functions and four probabilities assigned to each one, corresponding to four pure strategies. Each payoff function is formed by sixteen entries, and the game needs a total of 32 parameters to be defined. The respective Gambit specifications for the obtained games are:

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

0.98267224427866 4.21395317499410 4.45924895751567 3.72676363915463  
 4.95287782771976 0.74554895736572 5.90894720126257 5.03363507876793  
 4.90879611771289 5.10058338470355 0.98138660134146 3.90117603900288  
 1.84313155798613 3.42478969101750 0.18734463294673 2.71534077251105  
 3.13416415173927 3.99677337520616 4.38020969853245 2.95940398682505  
 0.51173901028550 4.15321282880840 1.63333367136465 2.91586186813719  
 5.79721049897973 3.72865242409535 1.18104088354410 5.61055432167966  
 4.35823201800598 3.96402619911103 3.25812748755708 1.79920703203677

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

2.19190732275435 3.59382681741100 3.92178637890438 1.25236559367649  
 3.50527333099260 4.14811766856772 3.45224297323453 0.32717782098832  
 5.63810903953332 5.51902806589179 3.21182071078489 0.90022948279260  
 3.60219877640239 2.82830987194108 3.16653995044072 1.01180140355985  
 0.41064997398857 3.59396167641395 0.41727390069669 1.29652996430380  
 0.41727390375206 4.14453938585969 0.03225639393592 0.30774552320562  
 1.09936892013825 2.85103299403893 1.44547133649356 1.22396158210748  
 1.94038845277630 2.68366429471404 4.22952134769444 1.62541814528757

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

2.47673529403952 0.61967466403382 3.31059538286365 1.50927992332563  
 2.82769949014882 1.82951784910409 4.98132065964718 4.07449183235372  
 5.46562766022071 4.06384407428594 2.33874338228706 1.36027999548710  
 3.81778287717519 2.39135696885551 0.40939350562141 2.86657026041609  
 0.89730821732603 1.71005745783477 4.02419258549284 2.19019967476280  
 2.54515344409012 1.56143736992105 4.11122847173052 3.66249681163535  
 2.26397215093938 3.46043685750669 2.05541576135265 1.45107670708347  
 2.50795990612965 0.61806402549496 0.98649200930137 4.30199353335779

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

0.15877001900309 0.84001937046689 1.64390862296053 0.65821454704507  
 0.98845032177355 0.84440341596286 1.06017512109401 0.59873441240232

0.73556658254596 1.36174768319683 1.49130410488584 0.35273463909088  
 1.35705798833897 0.84865412448053 0.84309006650862 0.76444532715671  
 0.79821032091653 1.01060684555844 0.97790022009455 0.45350618661411  
 0.98452101754703 0.86715057858854 0.84780304703714 1.00815120406471  
 1.29359859652837 1.57535046426617 0.94780348660830 0.91396877511062  
 1.05967264725668 0.18367982657270 1.14238129676957 0.08722602119876

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

0.66894605336500 0.48195922204941 0.89060423222173 1.00000304116267  
 0.98008100337764 0.73160073332938 0.97247549882239 1.26515465975231  
 0.34603630032437 1.32509198788646 0.12426716447978 1.02072255928597  
 0.75493088363963 0.15650204831742 0.45527047033130 1.22130759686060  
 0.20643777330041 1.03149346736749 0.71399560581207 0.64455542006802  
 0.71309182798410 0.77109649000536 0.49215394143855 1.46938988942684  
 0.68282303819701 1.37986654012774 0.61033064125703 1.15645480342609  
 0.50758251817803 0.02869483549246 0.58304023123923 0.95225129155605

**Example 2.** This example uses a game structure with 3 players and 3 alternatives available to each one. One group of two equilibria is employed to illustrate the method:

With two simultaneous equilibria

(Search interval for each parameter = [0,6] and CF final value = 5.115906944666653114e-17) :

- ( 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]

presenting three probability mass functions and three probabilities attributed to each one. Each payoff function has 27 entries, and the game, 81 parameters to adjust.

The corresponding Gambit specification for the obtained game is:

NFG 1 R "3 players with 3 alternatives each - payoff functions have 27 elements (total = 81 parameters)"

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

4.87730799152229 3.28347123536406 4.19022799303256 5.03945808654530 3.75626602044382  
 2.36356144705913 2.79307406680758 1.03560672799776 0.71043231752848 4.15912462035912  
 3.20876613885584 3.23709524493038 1.83092331432987 1.88478608994127 0.99619382569040  
 3.40255543255050 0.01155429977109 2.59500535001506 1.27770663303709 3.43126639071127  
 4.16123800949969 4.04815456019240 5.35715880794089 1.86609311693128 5.85772181878253  
 3.70577263358947 4.08064018613775 2.74518568567186 1.17827099221037 4.89948502493694  
 4.30734889361273 4.41362543083864 4.93291061635495 0.22302814776210 3.66751545732906  
 4.59141858009954 4.06806067042159 2.06028844493351 1.73026028046544 2.70945325293788  
 4.81154384143595 3.76436062284467 4.67958624543570 5.22058262048351 4.41165686782695  
 3.87407385479868 1.95780166785328 4.63898337123355 3.77377553388769 2.40938367341950  
 0.64588924213741 4.07293809725935 4.06908562514580 3.61010521940686 2.97976272358055  
 2.01933554486692 4.17859841323923 3.09324345160248 4.52108961111780 5.58013913264098  
 2.74974663346853 2.86022729450489 5.56556815410852 5.17343002806785 2.34176191381577  
 5.27850904561861 4.71950765829178 3.92229779292374 4.23576260064333 0.15436222580867  
 4.97988824033411 3.60924187076557 2.18345588574352 0.76568104079887 1.11338162363681  
 4.38056187814023 5.56141218115571 3.76033251691326 4.93558325114978 4.81228113741681  
 4.46676815533804

**Example 3.** Here the game structure features 3 players, with 2 alternatives each - two groups of Nash equilibria were chosen with 2 different strategic games to be found. The equilibria sets are:

With two simultaneous equilibria

(Search interval for each parameter = [0,2] and CF final value = 6.0e-17)

- ( 0.1 , 0.9 , 0.75 , 0.25 , 0.999 , 0.001 ) [mixed]
- ( 0.55 , 0.45 , 0.9 , 0.1 , 1.0 , 0.0 ) [partially mixed]

With three simultaneous equilibria

(Search interval for each parameter = [0,50] and CF final value = 5.0e-15)

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

- ( 0.55 , 0.45 , 0.9 , 0.1 , 1.0 , 0.0 ) [partially mixed]
- ( 0.5 , 0.5 , 0.3 , 0.7 , 0.0 , 1.0 ) [partially mixed]

with 3 probability mass functions and 2 probabilities assigned to each one. Payoff functions have eight entries, and the game has twenty four parameters to be found.

The Gambit specifications for the obtained games are, respectively:

NFG 1 R "2x2x2 - 3 players - 2 alternatives each" { "Player 1" "Player 2"  
"Player 3" } { 2 2 2 }

0.68654068362563 1.20415785256587 0.16005644417248 0.68551523430569  
0.21672288341703 0.72385626457349 0.94603041180153 1.20603499802767  
0.82080724343005 0.95525932309452 0.21442854297666 0.59440384714411  
1.58811995784320 0.00481222342603 0.07862032235506 0.00000000000034  
0.00097451632240 0.74864277513768 1.38174997318192 0.93952070938722  
0.72140928873838 0.00000000000088 1.98080243831927 0.55823388580237

NFG 1 R "2x2x2 - 3 players - 2 alternatives each" { "Player 1" "Player  
2" "Player 3" } { 2 2 2 }

11.33245988562835 17.49457489995949 18.78946857407635 11.33865066419950  
11.37933304228297 21.37036001412119 9.62275990779322 17.49340852482133  
3.27692016945539 9.56703810301457 11.38075918471280 1.59187344512556  
9.65352003480766 14.19000148397084 16.95160185411393 24.08544722227705  
24.58865378097980 9.63780578169272 26.47126922932137 15.64675855182206  
32.88898628214248 20.28615808100198 23.13189684373731 34.11192550617629

## 5. Interpretation of results and recommendations for independent testing

Once more we remind the readers 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 Gambit specifications make it possible to easily verify that the synthesized games have the initially given equilibria. Another important fact is that, in all cases, the final objective function value was under  $10^{-15}$ , 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 lines of action - 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 2, 3, 4, or 5 possible and simultaneous Nash equilibria, representing multiple favorable consumption configurations, for instance. In this fashion, with the help of the proposed algorithm it is possible to open this possibility by positioning the preestablished Nash equilibria group into the overall set of equilibria of the resulting game.
- Example 2 represents a context with 3 agents and 3 distinct moves available to each one - this setting may correspond to 3 firms forming a productive cluster and controlled by a central office which considers satisfactory to reach one out of 2 production profiles in the overall site so that the proportion of the production level of 9 different products (3 x 3) be given by any one of the proposed Nash equilibria. The obtained payoff functions contain the "rewards" capable of making a certain set of equilibria attractive and feasible.
- Example 3 corresponds again to the interaction among 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. In this case, 2 experiments with 2 and 3 simultaneous Nash equilibria were used, conveying the desire of allowing the occurrence of multiple satisfactory production profiles.

## 6. Conclusion

This work describes a new approach to synthesize finite strategic games with given structure and pre-selected multiple Nash equilibria in an automatic way, showing how that is possible by using an evolutionary global optimization method. The new findings are based on two main pillars: a proposition due to R. D. McKelvey, relating Nash equilibria and global minima of certain objective functions (Liapunov functions for specific dynamical systems, as described in [10]) and the utilization of an effective evolutionary global optimization algorithm (Fuzzy ASA). It is worth to highlight that the results here described could hardly be obtained without the flexible structure of evolutionary methods, which make it possible to employ techniques that may be considered infeasible when dealing with conventional, gradient-based methods. The usefulness of the results resides in the possibility of not only positioning several pre-set equilibria simultaneously, but also fine tuning the frequency of the several alternatives available to each player. Other positive points are the easiness of implementation, independent tuning of the range of payoff functions' components and provision for inserting further constraints.

## Final words

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

## 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] Oliveira Jr., H.A. Designing Strategic Games With Preestablished Nash Equilibrium Through Artificial Inference and Global Learning. *Preprints* **2019** , 2019110148 (doi: 10.20944/preprints201911.0148.v2).
- [21] M. J. Osborne, A. Rubinstein, *A Course in Game Theory*, MIT Press, Cambridge, MA, 1994.
- [22] 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.