Applied Mathematics and Computation 251 (2015) 442–452

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Presenting an algorithm to find Nash equilibrium in two-person static games with many strategies

Mohammad Mahdi Mohtadi *, Kazem Nogondarian

Industrial Engineering School, Iran University of Science and Technology, Hengam Street, Resalat Square, Tehran, Iran

ARTICLE INFO

Keywords:
Game theory
Static Game
Big problems
Computing Nash equilibrium
Mixed strategy
Game solver algorithm (GSA)

ABSTRACT

This article attempts to present an algorithm to find Nash equilibrium point in large two-person static games. In static games, each player chooses his/her strategy simultaneously with other players. Equilibrium can be easily recognized in static games with a limited number of strategies. This paper focuses on two-person games in which each player deals with many strategies. After reviewing the literature, this article considers concepts such as game, equilibrium, and rationalism in order to identify the problem and the required methodology to solve it; eventually, a heuristic-meta heuristic algorithm will be presented to solve such problems which have been validated using a sample problem.

1. Introduction

Games are defined as decision making when faced a rival.

In this situation, to maximize profit, first any player must be analyzed reaction of other players and choose best alternative candidate with respect to reaction of competitor and consideration of the same choice for them.

Game theory were developed to assimilation of behavior in this positions. It is one of the most important paradigm with application in different field like Economy, Mathematics, trade marketing, biology and behavior science nowadays.

In the literature review, different type of game classification is available. On the one view, games were categorized into two types, static and dynamic group. Choosing strategy is sequentially in dynamic games. But in static games, rivals choose their strategies simultaneously and without knowing others' decisions.

The key purpose in the game theory is finding of optimum behavior of players. Optimized reaction in the game is determined acceptable behavior between players with maximum profit that it was defined as Nash equilibrium (Pure or mixed strategy) in the static games.

Finding the equilibrium behavior in a game is not of particular complexity as long as its dimensions are small. As the literature reveals, however, analyzing static games with many strategies has not been much discussed and in this field, it is seen many papers.

This article attempts to present an algorithm to solve the mentioned games. To this aim, after reviewing the literature, the methodology to find the pure and mixed strategy Nash equilibrium points in small games and the challenges of using them in large games have been studied. Finally, a heuristic-meta heuristic algorithm has been presented to find equilibrium in large games. In order to validate this algorithm, a problem has been designed and solved using a series of sample data; then, the produced responses have been compared with the responses obtained from precisely solving these samples (see Figs. 1-4).

* Corresponding author.
E-mail addresses: Mohtadi@IUST.ac.ir (M.M. Mohtadi), Noghondarian@IUST.ac.ir (K. Nogondarian).

http://dx.doi.org/10.1016/j.amc.2014.11.039
0096-3003/

2. Review of literature

In a book named as the ''Theory of Game and Economic Behavior'', Morgenstern and Neumann [1] attempted to regard the game theory as the basis for a new approach towards the economic science. Nash [2] could generalize the Max–Min theorem in order to involve non-cooperative non-zero sum games. He showed that in each game, there is at least one strategy such that if a player selects a strategy other than that, better results will be definitely obtained. This equilibrium has been known as Nash Equilibrium. Topkis [3] presented an article aiming to find equilibrium in n-person games. Patterson [4] presented a linear programming model to solve complicated two-person games. The payoff matrix is regarded as the input of his model which has transformed it into a linear programming model and been solved through dual computations. His solved model included 15 strategies for each player.

In the field of dynamic games, bi-level programming and sequential games have attracted the researchers' attention. [5,6] However, as was mentioned in the literature, finding equilibrium in simultaneous large games requires more researches which have been performed in this paper.

Recent research in the scope of game theory, focused generally on the evolution game, cooperation game and population game. Wang et al. have some papers on this topic. [7–10] They studied optimal interdependence between networks for the evolution of cooperation. They show that only an intermediate density of sufficiently strong interactions between networks warrants an optimal resolution of social dilemmas [7].

They showed that regardless of the details of the evolutionary game and the interaction structure, the self-organization of fitness and reward gives rise to distinguished players that act as strong catalysts of cooperative behavior and there also exist critical utility thresholds beyond which distinguished players are no longer able to percolate [8].

In another paper by them, the effects of co-evolution on Self-organization towards optimally interdependent networks have been studied. They studied whether it could give rise to elevated levels of cooperation in the prisoner's dilemma game and showed that the interdependence between networks self-organizes so as to yield optimal conditions for the evolution of cooperation [9].

They also studied the outcome of the public goods game on two interdependent networks that were connected by means of a utility function [10].

Wang and Perc introduced a simple rule that influences the selection of players that are considered as potential sources of the new strategy and found that increasing the probability of adopting the strategy from the fittest player, promotes the evolution of cooperation [11].

They indicated that heterogeneity in aspirations may be key to the sustainability of cooperation in structured populations [12].

Perc and Szolnoki reviewed recent works on evolutionary games incorporating co-evolutionary rules and outlined directions for future research, thereby particularly focusing on dynamical effects of co-evolutionary rules on the evolution of cooperation [13].

Perc and Grigolini reviewed Collective behavior and evolutionary games.

Collective behavior has been introduced in this paper as the lack of a central planner. It covers many different phenomena in nature and society such as different games especially evolutionary games [14].

Szabo and Fath in a review paper focused on what sense and how the graph structure of interactions can modify and enrich the picture of long term behavioral patterns emerging in evolutionary games [15].

Wang et al. analyzed the time course of cooperation evolution under different evolution rules [16].

Perc et al. summarized basic results concerning the public goods game on well-mixed populations [17].

Wang et al. considered two-layer scale-free networks with all possible combinations of degree mixing, wherein one network layer is used for the accumulation of payoffs and the other is used for strategy updating [18].

Helbing and Yu showed that the outbreak of prevailing cooperation, when directed motion is integrated in a game-theoretical model, is remarkable, particularly when random strategy mutations and random relocations challenge the formation and survival of cooperative clusters [19].

Nowak discussed some mechanisms for the evolution of cooperation according to this principle that natural selection implies competition and therefore opposes cooperation unless a specific mechanism is at work [20].

3. Game theory and equilibrium

Standard decisions are made in the face of nature, which is considered as a neutral factor. But the decision game is in the face of an intelligent factor. Competition is a game instance. The competitor is considered as an intelligent factor that is seeking for interests which normally are in conflict with those of the opposed competitor. In these situations, a game will be formed. Each game, player and a set of behaviors a player can choose among them, are called strategy. In real problems, the game can have more than two players. Although the two-player games have many applications in real problems. Each choice facing players, is called Strategy and the benefit gained by the players at the end of each game, is called revenue [21].

The most critical question in the game is about the optimal behavior of competitors, or the so-called balance point of the game. In equilibrium, each player uses the strategy that has the best response to the other players' strategies. In other words, each of these players, is looking for the solution to this question that in the face of the competitor decision and understanding the fact that he or him is thinking similarly, he or she should choose which strategy to bring the most revenue for him or her. The first approach that can be used to solve this question, is using the principle of rationality. As an initial model, and according to this principle, the strategy whose choice in the face of every single competitors' strategy, brings most revenues for the player, is known as the dominant strategy and its choosing, is preferred [22].

Suppose in a game, Player 1 has N and player 2 has M different strategies in their face. If player 1, chooses strategy i and player 2 choose strategy j, U_{ij} is the player 1 revenue. If for each M strategies of player 2, we can find a strategy among strategies of player 1 whose revenue is more than strategy k; hence, player 1 will never choose this strategy.

Using this concept, and through eliminating the strategies that are not best responses, we can further restrict the strategy space of players. But even after eliminating the failed and non-rational strategies, there is no dominant strategy in many real problems i.e. in some chosen strategies of the competitor, one strategy and in some other strategies, another strategy is preferable. In these situations, the Nash equilibrium is noteworthy in order to achieve the optimal point. This equilibrium is achieved when each player chooses his or her strategy based on a good faith towards the competitor choice. The strategies that players choose by this method, from its Nash equilibrium strategy.

The requirement of achieving this equilibrium, is that each player chooses the strategy that has most revenues for him or her, based on his or her belief about the competitor choice. To find the mentioned equilibrium point, the optimal response of player 2 to any strategies of player 1 and also the optimal response of player 1 to any strategies of player 2 must be determined. The Nash equilibrium is the point to which the two parties have agreed to it implicitly. It means that if the best response of player 2 to the strategy S_i of Player 1 is strategy S'_j and the best response of player 1 to the strategy S'_j of Player 2 is strategy S_i these paired strategies is from the pure strategy Nash equilibrium [21,23].

In Some games, the pure strategy Nash equilibrium, means that there is not one unique Nash equilibrium point or there are more than one pure Nash equilibrium point. In this situation, the mixed strategy Nash equilibrium is used. Mixed strategy represents a mental uncertainty of a player towards the competitor choice. This uncertainty is represented by probability. Thus a mixed strategy is a mixture of pure strategies of the player with a certain probability distribution. Considering the fact that the game is repeated many times, we can consider these probabilities as the choosing frequency of each strategy over time by each player and determine the revenue of each player by using that [24,25].

To determine the mixed strategy Nash equilibrium, we can use the KKT optimality situation and Lagrange function. In this situation, after eliminating the Non-rational strategies and the formation of the following system of linear equations, we can determine the strategy of each player and his choosing probability.

$$\sum_{i=1}^{n} U'_{ij} p_i = \lambda_1 \quad j = 1, 2, \ldots, n, \tag{1}$$

$$\sum_{j=1}^{m} U_{ij} q_j = \lambda_2 \quad i = 1, 2, \ldots, m, \tag{2}$$

$$\sum_{i=1}^{n} p_i = 1, \tag{3}$$

$$\sum_{j=1}^{m} q_j = 1, \tag{4}$$

$$p_i, q_j \geqslant 0 \quad i = 1, 2, \ldots, n \quad j = 1, 2, \ldots, m, \tag{5}$$

where:

p_i Probability of choosing the strategy i By Player 1(i = 1,2, ..., n).
q_j Probability of choosing the strategy j By Player 2 (j = 1,2, ..., n).
U_{ij} Player 1 Consequence in case of choosing the strategy i By Player 1 and strategy j By Player 2.
U'_{ij} Player 2 Consequence in case of choosing the strategy i By Player 1 and strategy j By Player 2.

In summary it can be said that to determine the equilibrium in a game, at first we must remove the non-rational strategies sequentially so that there would not be any other non-rational strategy for removing. If 1 strategy remains for each player, this strategy is introduced as pure strategy Nash equilibrium. Otherwise, by formation of linear equations system, the mixed probability of choosing each strategy is determined.

If the players' target functions are specified rather than the revenue matrix, we must identify the whole areas of justifiable solutions of 1 player, and for each extracted solution, the optimal strategy of opposed player and revenue amount of both players in both paired strategies should be identified and then, the above measures should be taken.

Taking these issues into account, the following algorithm can be presented to determine the equilibrium point.

1- Remove irrational strategies as mentioned below.
   1-1- For each strategy of player 1, select the best strategy for player 2 and label it. Repeat this until the last strategy of the player 1.
   1-2- If there is an unlabeled strategy for player 2, remove it and assume i = 0; otherwise, i = 1.
   1-3- For each strategy of player 2, select the best strategy for player 1 and label it. Repeat this until the last strategy of the player 2.
   1-4- If there is an unlabeled strategy for player 1, remove it and assume i = 0; otherwise, i = 1.
   1-5- If i * j = 1, go to step 2; otherwise, go to step 1–1.
2- If only one strategy remains for each player, the mentioned strategy is introduced as the pure strategy Nash equilibrium of the game; otherwise, go to step 3.
3- After the formation of the correspondent linear equations (Eqs. (1)–(5)), the probability of selecting each strategy will also be determined.

4. An algorithm to solve large problems

The mathematical model with two goal functions, which are related to each of the players, is defined as below:
In a competitive mode P_1 solves:

in a competitive mode P_1 solves :

$$\mathrm{Max}_{x_1} f_1(x_1, x_2) = c_{11}^T x_1 + c_{12}^T x_2, \tag{6}$$

& Statically P2 solves:

$$\mathrm{Max}_{x_2} f_2(x_1, x_2) = c_{21}^T x_1 + c_{22}^T x_2, \tag{7}$$

s.t.

$$A_1 x_1 = b_1, \tag{8}$$

$$B_1 x_2 = b_2, \tag{9}$$

$$A_2 x_1 + B_2 x_2 = b_3. \tag{10}$$

Many of the large mathematical programming problems are also of computational complexities even under noncompetitive conditions. Of course, the existence of any rival in such a situation can make the problem more complicated. Although the explained approach in the previous section is suitable to find equilibrium points in small games, it cannot be used for complicated and large problems, since having the strategies' total payoff matrix is necessary to use it and this issue cannot be materialized in large problems due to having many responses; it cannot be solved in a logical period even with the help of computers.

Provided algorithm in this paper is a solution for this problems. Principles and rule that is used in development of algorithm are following:

1- Any static game are involved Nash equilibrium. (pure or mixed strategy).
2- If selected strategies for player 1 and player 2 were i and j respectively, and this selection were accepted between them in the serial choice procedure, it is Nash equilibrium.
3- If it is uncertainly in the choice of competitor, Nash equilibrium with combined strategy will be developed in the form of mentioned player probable combined strategies. Number of answers is unrestricted in this combination.
4- Since any players were effort to maximize profits, selection to answers was limited. In fact it caused that range of appropriate answers was be limited and then rang of searchable answer in the latter selection was decreased.
5- In the approximately algorithm of problem solving, answers were accepted when they were convergent and their variation was be minimized in successive solution.

On the basis of this principle, developed algorithm for this problems are included following steps:
Step 1- Create the empty sets of£ and £′ as the storage location of definite strategies and the empty sets of $\mho$ and $\mho'$ as the storage location of the temporary strategies.
Step 2- Determine the values of a (the least repetition), b (the number of returns to the previous generations to compare the responses), x (the number of suitable primary responses), y (the amount of advancement in each repetition), and z (the maximum acceptable changes in the compared repetitions).
Step 3- Assume k = 1.
Step 4- This step includes the secondary steps as below:
   4-1- Assume i = 0.
   4-2- Define a set of randomly justified responses with X number for player 1 and name them as A_i.

4-3- If A_i is empty, go to step 4–4; otherwise, go to step 4–5.
4-4- If i = 0, go to step 5; otherwise, go to step 4–13.
4-5- Select a random member of A_i and call it k_i.
4-6- Remove k_i from the set of A_i and add it to set $\mho$.
4-7- Correct the value of i from the equation i = i + 1.
4-8- Per k_{i-1}, extract the optimal responses of player 2 from the following linear programming model and call it A'_i.

$$\mathrm{Max}_{P_2} f_2(x_1, x_2) = c_{21}^T x_1 + c_{22}^T x_2, \tag{11}$$

s.t :

$$B_1 x_2 = b_2, \tag{12}$$

$$A_2 x_1 + B_2 x_2 = b_3, \tag{13}$$

$$x_1 = k_{i-1}. \tag{14}$$

4-9- If i = 1, go to step 4–13; otherwise, go to step 4–10.
4-10- If there is a member among the members of set A'_i whose value is equal to one of the values of k'_m (m = 1, .., i−1), go to step 4–11; otherwise, go to step 4–13.
4-11- Register the values of k'_j (j = m, .., i−1) as the members of£ ′ and the values of k_j (j = m, .., i−1) as the members of £.
4-12- Apply the following updates.

$$A'_i = A'_i - \mho' - £'. \tag{15}$$

4-13- If A'_i is empty, go to step 4–14; otherwise, go to step 4–15.
4-14- Correct the value of i from Equation i = i−1 and go to step 3–4.
4-15- Select a random member of A'_i and call it k'_i.
4-16- Remove k'_i from the set of A'_i and add it to set $\mho'$.
4-17- Per k'_i, extract the optimal responses of player 1 from the following linear programming model and call it A_i.

$$\mathrm{Max}_{P_1} f_1(x_1, x_2) = c_{11}^T x_1 + c_{12}^T x_2, \tag{16}$$

s.t :

$$A_1 x_1 = b_1, \tag{17}$$

$$A_2 x_1 + B_2 x_2 = b_3, \tag{18}$$

$$x_2 = k'_i. \tag{19}$$

4-18- If there is a member among the members of set A_i whose value is equal to one of the values of k_m (m = 0,. . ., i−1), go to step 4–19; otherwise, go to step 4–20.
4-19- Register the values of k_j (j = m,..., i−1) as the members of £ and the values of k'_j (j = m + 1,..., i−1) as the members of £′.
4-20- Apply the following update.

$$A_i = A_i - \mho - £. \tag{20}$$

4-21- If i = y, go to step 4–13; otherwise, go to step 4–3.

Step 5- Correct the value of k from the equation k = k + 1. If k $\geqslant$ a − b, go to step 6; otherwise, go to step 4.

Step 6- This step includes the secondary steps as below:

6-1- Remove the repetitive members from the sets of £ and £′, if any exists.
6-2- Label £ from S_1 to S_n and £′ from S'_1 to S'_m.
6-3- For each pair of (S_i, S'_j), calculate the value of goal function of player 1 and player 2 (f_1, f2) and label them as (U_{ij}, U'_{ij}).
6-4- Remove irrational strategies of the two players based on the presented algorithm in the previous section.
6-5- Calculate the mixed strategy Nash equilibrium of the players and call them L_k and L'_k, respectively.

Step 7- If k $\leqslant$ a, go to step 4; otherwise, go to step 8.

Step 8- If the average relative change of L_k and L'_k in relation with $L_{(k\text{-}b)}$ and $L'_{(k\text{-}b)}$ is less than z, go to step 9; otherwise, go to step 4.

Step 9- Mention those p_i and q_i which have received positive values along with the correspondent S_i and $S_i^{'}$ as the final responses of the problem. Mention the values of L_k and $L_k^{'}$ as the payoff of each player at the end of the game. The algorithm is now completed.

5. Algorithm analysis

The main idea of this algorithm which is named game solver algorithm (GSA) is to find the rational strategies; it starts searching based on this reality that each rational strategy is a response to at least one rational strategy. Therefore, if a strategy cannot be regarded as a response to any of the rational strategies of the rival players, it will not be rational.

Assume that player P_1 selects the response x_{11} and player P_2 is aware of this decision. In this case, the problem of player P_2 will be turned as below:

$$\mathrm{Max}x_{P_2} f_2(x_1, x_2) = c_{21}^T x_1 + c_{22}^T x_2, \tag{21}$$

s. t:

$$B_1 x_2 = b_2, \tag{22}$$

$$A_2 x_1 + B_2 x_2 = b_3, \tag{23}$$

$$x_1 = x_{11}, \tag{24}$$

The response to this model is regarded as the unique response of x_{21}. Now, if player P_1 decides to modify his own strategy appropriate to this behavior of player P_2, the following model must be solved:

$$\mathrm{Max}x_{P_1} f_1(x_1, x_2) = c_{11}^T x_1 + c_{12}^T x_2, \tag{25}$$

s. t:

$$A_1 x_1 = b_1, \tag{26}$$

$$A_2 x_1 + B_2 x_2 = b_3, \tag{27}$$

$$x_2 = x_{21}, \tag{28}$$

If the response to this model is also the unique point of x_{11}, the problem is solved and the point (x_{11}, x_{21}) is regarded as one of the Nash equilibrium points of the game. However, if the response to the mentioned problem is something other than x_{11}, the problem related to player 2 must be re-solved considering this response. Continuing these subsequent solutions, a series of responses will be obtained as below.

In the discrete space, the number of responses will be finite and in this subsequent solution, a point such as x_{1n} or x_{2n} will be obtained which is equal to one of the previous responses (such as x_{1i} or x_{2i}).

$$x_{11} \to x_{21} \to x_{12} \to x_{22} \to x_{13} \to x_{23} \to \cdots \to x_{1i} \to x_{2i} \to \ \ldots \to x_{1n} \tag{29}$$

If the equation $x_{1i} = x_{1n}$ is true, the response of player P_2 to x_{1n} will be x_{2i}; the previous responses will occur in a sequence until the response x_{1n} will be obtained again. The same is true when the response of x_{2n} has been obtained. It means that the problem has been placed in a loop. The same points can be introduced as the strategies of the both parts.

If multiple responses are obtained, each response separately creates a chain of responses.

Each obtained loop is considered as Nash equilibrium and their collection forms the mixed strategy Nash equilibrium of the game. In order to develop the presented algorithm, these loops in the problem must be found. However, considering the dimensions of the problem and its computational complexities, it is not possible to precisely solve it regarding all the response space and identify all the loops. Consequently, the presented algorithm starts searching based on a series of random primary responses and identifies the loops. Parameter x is the number of the random primary responses from which the searching process starts. The greater the value of this parameter, the vaster space will be researched and the more loops are identified. Each random response enters one or several loops with an uncertain number of subsequent solutions. Parameter y of this model has determined the maximum number of the mentioned subsequent solutions. This parameter is for cases when the distance between the randomly selected points to the loop is much and it is defined in order to prevent too much increase of the computations' volume in the algorithm.

Parameter A determines the least repetition of the algorithm's main step. After a-time repetition of the algorithm, the produced response is compared with response B of the previous generation. If the created change is less than parameter z as the least defined change, the required precision has been obtained and the algorithm stops. In case of high values, the primary responses of (x) and parameters (A) and (B) along the the low value of parameter z of the solution will converge to the precise value.

This algorithm owns a memory and the produced responses in each step are used in the next steps.

6. Modeling the sample problem and the solution algorithm

In order to validate the presented algorithm, a competitive problem is modeled as below.

$$\mathrm{Max}_{p_1} L_1 = \sum_{i=1}^{n} Z_i(a_i + b_i), \tag{30}$$

$$\&\,\text{statically}\ \ p_2\ \text{Solves}:$$

$$\mathrm{Max}_{p_2} L_2 = \sum_{i=1}^{n} Z'_i(a_i + b'_i), \tag{31}$$

s.t.

$$\sum_{i=0}^{n} X_{ij} \leqslant 1 \quad j = 1,2,\ldots,n, \tag{32}$$

$$\sum_{j=0}^{n} X_{ij} - \sum_{i=0}^{n} X_{ij} = 0 \quad i = 1,2,\ldots,n, \tag{33}$$

$$X_{ij} + X_{ji} \leqslant 1 \quad i = 0,1,\ldots,n \quad j = 0,1,\ldots,n, \tag{34}$$

$$Z_i \leqslant \sum_{j=0}^{n} X_{ij} \quad i = 1,2,\ldots,n, \tag{35}$$

$$t_j = \sum_{i=0}^{n} (t_i + d_i + t_{ij}) X_{ij} \quad j = 1,2,\ldots,n, \tag{36}$$

$$t_0 = 0, \tag{37}$$

$$\sum_{i=0}^{n} Y_{ij} \leqslant 1 \quad j = 1,2,\ldots,n, \tag{38}$$

$$\sum_{i=0}^{n} Y_{ij} - \sum_{j=0}^{n} Y_{ij} = 0 \quad i = 1,2,\ldots,n, \tag{39}$$

$$Y_{ij} + Y_{ji} \leqslant 1 \quad i = 0,1,\ldots,n \quad j = 0,1,\ldots,n, \tag{40}$$

$$Z'_i \leqslant \sum_{j=0}^{n} Y_{ij} \quad i = 1,2,\ldots,n, \tag{41}$$

$$t'_j = \sum_{i=0}^{n} (t'_i + d'_i + t'_{ij}) Y_{ij} \quad j = 1,2,\ldots,n, \tag{42}$$

$$t'_0 = 0, \tag{43}$$

$$\frac{t'_i - t_i}{M} \sum_{j=0}^{n} X_{ij} \leqslant Z_i \quad i = 1,2,\ldots,n, \tag{44}$$

$$\frac{t_i - t'_i}{M} \sum_{j=0}^{n} Y_{ij} \leqslant Z'_i \quad i = 1,2,\ldots,n, \tag{45}$$

$$Z_i + Z'_i \leqslant 1 \quad i = 1,2,\ldots,n, \tag{46}$$

$$X_{ij} \in \{0, 1\} \quad i = 0, 1, \ldots, n \quad j = 0, 1, \ldots, n, \tag{47}$$

$$Y_{ij} \in \{0, 1\} \quad i = 0, 1, \ldots, n \quad j = 0, 1, \ldots, n, \tag{48}$$

$$Z_i \in \{0, 1\} \quad i = 1, 2, \ldots, n, \tag{49}$$

$$Z'_i \in \{0, 1\} \quad i = 1, 2, \ldots, n. \tag{50}$$

This problem has been designed in the environment of a graph and the number of nodes of this graph, which has been shown by n, determines the size of this graph. This problem's parameters have been determined randomly. Mohtadi and Nogondarian [26] have modeled this problem; they showed that solving problems with more than 10 nodes by using the exact game solution algorithm is not possible in a logical period of time.

Genetic algorithm and tabu search algorithm are used to solve this problem. Genetic algorithm considers each solution as a Chromosome and begins its work with Chromosomes as the initial population and at each stage, by changing in present population, creates new generations. This algorithm performs its work for single-objective problems with encoding, evaluating, combining, mutation and decoding [27].

Encoding perhaps is the most difficult step of problem solving by genetic algorithm method and the other steps depend on this step. The main methods of encoding include binary coding, permutation coding, value coding and tree coding. Evaluation function (fitting) is achieved by applying the appropriate transformation on the objective function. This function evaluates each solution with a numerical value that indicates the level of solution suitability. The most important operator in genetic algorithm, is Crossover operator. The Crossover is a process in which the older generation chromosomes are combined and mixed together to create a whole new generation of chromosomes. The pairs that were considered as a parent in the selection section, interchange their genes in this section and create new members.

Mutation is another operator that creates other possible solutions. In genetic algorithm after a new member is created in new population, each gene mutates with a constant probability of mutation. In mutation, there is possibility that a gene is removed from the set of genes or a gene is added that had not been in the population. Mutation of a gene means a change in that gene, and various mutation methods are used depending on encoding method.

Decoding is the inverse operation of encoding. At this step, after the algorithm provides the best solution to the problem, it is necessary to perform the encoding inverse or decoding on solutions to obtain the real version of the solution.

The parameters of this problem are estimated as in Table 1 by using experimental designs and Taguchi method.

Tabu search algorithm at first, moves from an initial response. Then the algorithm chooses the best neighboring solution among the current neighboring solutions. If this solution is not in the tabu list, the algorithm moves to the neighboring solution. Otherwise, the algorithm, will check a criterion called aspiration criterion. Based on the aspiration criterion, if the neighboring solution is better than the best solution that has been found so far, the algorithm will move to it, even if the solution is in the Tabu list. After the algorithm moves to the neighboring solution, tabu list is updated. i.e. the previous one moves through that, it achieves to the neighboring solution, and is placed in the tabu list to prevent the algorithm to return to that solution and make a cycle. Actually the tabu list is a tool in tabu search algorithm through which the algorithm

Table 1
The parameter values of the employed algorithms.

Game solver algorithm		Tabu search algorithm		Genetic algorithm	
Parameter	Value	Parameter	Value	Parameter	Value
X	10	Maxit	100	Npop	20
Y	5	TL0	0.7	Maxit	30
Z	15			Pc	0.4
A	5			Pm	0.3
B	0.03				

Table 2
Exact game algorithm, exact and approximation problem solution algorithms.

	N	Exact algorithm			Genetic algorithm				Tabu search algorithm			
		P1	P2	t	P1	P2	t	e (%)	P1	P2	T	e (%)
1	3	726	800	3	726	800	2	0.00	726	800	3	0.00
2	4	799	1105	4	798	1105	4	0.06	798	1105	5	0.06
3	5	1616	801	5	1615	801	11	0.03	1616	800	14	0.00
4	6	1803	1046	6	1803	1045	54	0.00	1803	1045	68	0.00
5	7	1898	1740	7	1901	1736	127	0.08	1900	1737	159	0.05
6	8	2059	2108	8	2067	2099	338	0.20	2064	2102	423	0.12
7	9	2622	1825	9	2631	1815	1583	0.18	2614	1832	1979	0.16
8	10	2236	2838	10	2220	2853	2701	0.35	2248	2825	3390	0.26

will be prevented from being placed in a local optimization. After placing the previous move in the tabu list, some moves that previously were placed in the tabu list are removed from the list. The time that the moves and placed in the Tabu list, is determined by a time parameter called tabu tenure. Move from the current solution to a neighboring solution continues until the termination situation is met. Different termination situations can be considered for the algorithm. For example, the limitation of the number of moves to neighboring solution can be a termination situation [28].

The parameters of these two algorithms and also the game solution algorithm have been presented in Table 1.

The heuristic and exact algorithms of game solution have been separately encoded. Besides, in order to solve the linear programming problem of these two algorithms, the tabu search algorithm and genetic algorithm have also been encoded.

7. Numerical analysis

In order to validate the model, a series of sample problems in small and medium scales have been designed and solved using two exact and heuristic algorithms; their results have also been compared with each other. Aiming to investigate the presented model, a series of random data has been produced and the model has been compared with them. Table 2 demonstrates the results of solving the mentioned problem using exact game solution algorithm and exact and approximation problem solving algorithms. Table 3 shows the same problems with the approximate game algorithm.

In the chart .1,time of problem solving against size and based on separation of Game solving method and problem solving method(exact: e and genetic: G and tabu search: T) were provided, also in the chart .2 amount of errors in the any approximate algorithm based on size of problem solving was characterized.

Analyzing the results mentioned in the tables and charts above, it can be concluded that the presented heuristic algorithm is suitable in solving the game due to saving more time in solving the problem such that large problems, which cannot be solved by the exact algorithm in a logical period of time, can be solved with it. Besides, this algorithm's error falls in an acceptable range.

In addition, time of genetic algorithm is lessen from tabu search algorithm but in return amount of errors was increased .nevertheless totally both algorithm was included acceptable time and accuracy in the problem solving.

Furthermore, Table 4 reports the results of solving several large problems using the mentioned heuristic algorithm.

Table 3
Approximate game algorithm, exact and approximate problem solution algorithms.

	n	Exact algorithm				Genetic algorithm				Tabu search algorithm			
		P1	P2	t	e (%)	P1	P2	t	e (%)	P1	P2	t	e (%)
1	3	726	800	2	0.00	726	800	4	0.00	726	800	5	0.00
2	4	799	1105	2	0.00	798	1106	7	0.06	799	1105	8	0.00
3	5	1616	801	2	0.00	1615	802	9	0.03	1617	800	12	0.03
4	6	1804	1045	2	0.03	1805	1044	10	0.06	1805	1044	14	0.06
5	7	1902	1736	10	0.11	1906	1732	14	0.22	1904	1734	20	0.16
6	8	2067	2100	54	0.20	2077	2090	20	0.45	2073	2094	24	0.35
7	9	2633	1814	91	0.22	2642	1805	51	0.39	2607	1840	60	0.28
8	10	2214	2860	539	0.48	2193	2881	94	0.95	2269	2805	103	0.75

Table 4
The solution of the large sample problems with game algorithm.

	n	Genetic algorithm			Tabu search algorithm		
		P1	P2	t	P1	P2	t
1	16	2878	4896	280	2907	4867	330
2	17	4063	4038	300	4309	3792	351
3	18	6263	2840	330	6245	2858	380
4	19	3818	5270	388	3864	5224	454
5	20	4812	5041	414	4850	5003	493
6	21	4966	5421	482	4518	5869	554
7	22	5598	4864	570	5542	4920	678
8	23	7030	4432	621	6985	4477	714
9	24	6894	5068	766	7352	4610	881
10	25	5373	7067	832	4615	7825	990
11	26	5635	7182	1102	4815	8002	1311
12	27	5274	7679	1261	5325	7628	1475
13	28	8661	4861	1412	8053	5469	1694
14	29	8334	5688	1731	8395	5627	1991
15	30	8608	5879	1894	9285	5202	2216

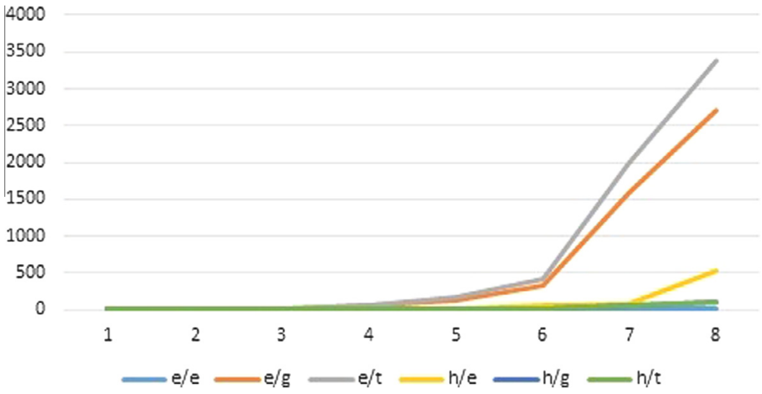

Fig. 1. Time of each algorithm in little problems.

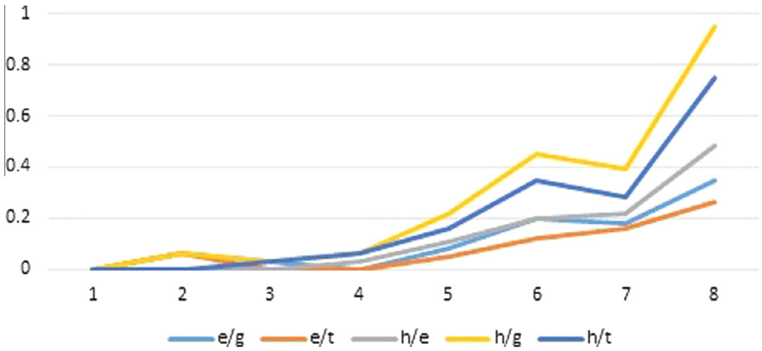

Fig. 2. Error in little problems Compared exact alghorithm.

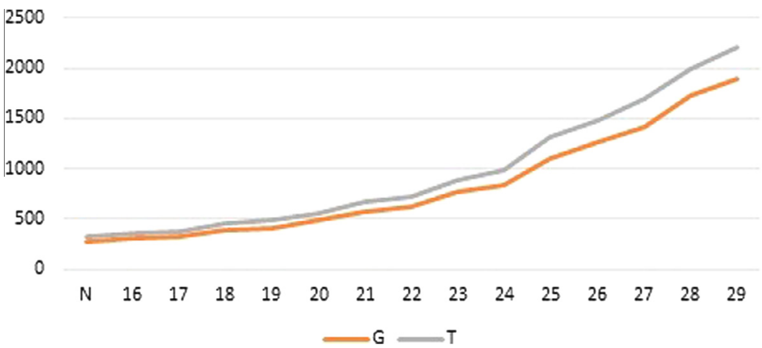

Fig. 3. Time of answer in big problems by two algorithms.

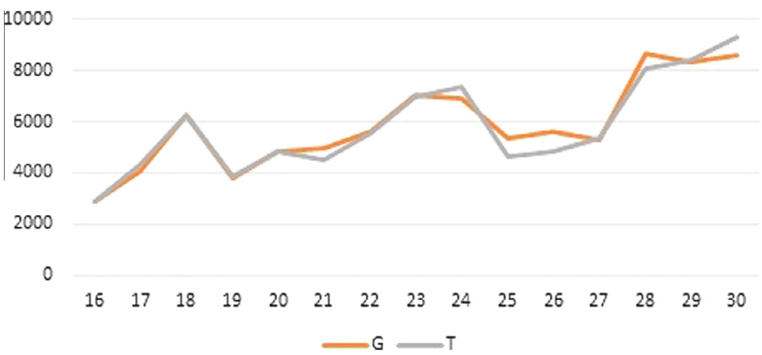

Fig. 4. Answer of problem (player 1) in big problems by two algorithms.

In the Table 3. comparison of problem solving time against size and with separation of type of problem solving algorithm (genetic, tabu search) was shown .Furthermore, in the table .4 answer of problem based on reaction of player 1against size of problem and with separation of type of algorithm was characterized .Such as small problem, Time of problem solving through genetic algorithm is lessen than tabu search algorithm that is shown in the both chart. Altogether answers in the both algorithm is acceptable.

8. Conclusion

This paper attempted to present an algorithm to solve two-person static games with many strategies. To this aim, after reviewing the literature, the algorithm for solving small games was presented and it was observed that this algorithm could not be used in solving large problems. Then, focusing on subjects such as rationality and analyzing its concept, a heuristic-metaheuristic algorithm was presented to solve such games.

Provided algorithm with random selection of justified answers from player 1 was characterized optimized answers of player 2 and then were determined answers of player 1 proportional to mentioned answers. This process to finding of mutual answers was continues as long answers were repetitive.

In order to increasing of accuracy in the answers, search was accomplished in the several steps with primary answers. On the basis of answers, Nash equilibration (mixed or pure strategy) was calculated. This procedure are following until searches no shown significant impact on final answer. Parameters of this algorithm are as follows: number of primary answers, repetition, maximum search in the any repetition, interval of number of compared searches and observed minimum variation.

Eventually, this algorithm was tested on a number of sample problems and the results of their exact and approximation algorithm solutions were compared. The results proved that saving a significant amount of computation time, the presented algorithm possessed the acceptable precision. Besides, the genetic algorithm was faster and had more error percentage in comparison with the tabu search algorithm.

References

[1] O. Morgenstern, J.V. Neumann, Theory of Games and Economic Behavior, Princeton University Press, New Jersey, 1944.
[2] J. Nash, Equilibrium points in n-person games, Nat. Acad. Sci. 36 (1950) 48–49.
[3] D.M. Topkis, Equilibrium points in nonzero-sum n-person sub modular games, SIAM J. Control Optim. 17 (1979) 773–787.
[4] M.C. Patterson, A linear programming model for solving complex 2-person zero, Econ. Finance 13 (2) (1990) 20–31.
[5] B. Colson, P. Marcotte, G. Savard, An overview of bilevel optimization, Ann Oper Res. 153 (2007) 235–256.
[6] B.V. Stengel, Efficient Computation of behavior strategies, Games Econ. Behav. 14 (1996) 220–246.
[7] Z. Wang, A. Szolnoki, M. Perc, Optimal interdependence between networks for the evolution of cooperation, Sci. R 3 (2013) 2470.
[8] Z. Wang, A. Szolnoki, M. Perc, Rewarding evolutionary fitness with links between populations promotes cooperation, J. Theor. Biol. 349 (2014) 50–56.
[9] Z. Wang, A. Szolnoki, M. Perc, Self-organization towards optimally interdependent networks by means of coevolution, New J. Phys. 16 (2014) 033041.
[10] Z. Wang, A. Szolnoki, M. Perc, Interdependent network reciprocity in evolutionary games, Sci. R. 3 (2013) 2470.
[11] Z. Wang, M. Perc, Aspiring to the fittest and promotion of cooperation in the prisoner's dilemma game, Phys. Rev. 82 (2010) 021115.
[12] M. Perc, Z. Wang, Heterogeneous aspirations promote cooperation in the prisoner's dilemma game, PLoS ONE 5 (2010) 15117.
[13] M. Perc, A. Szolnoki, Coevolutionary games-A mini review, BioSystems 99 (2010) 109–125.
[14] M. Perc, P. Grigolini, Collective behavior and evolutionary games – An introduction, Chaos, Solitons Fractals 56 (2013) 1–5.
[15] G. Szabo, G. Fath, Evolutionary games on graphs, Phys. R 446 (2007) 97–216.
[16] Z. Wang, S. Kokubo, J. Tanimoto, E. Fukuda, K. Shigaki, Insight into the so-called spatial reciprocity, Phys. Rev. Lett. 88 (2013) 042145.
[17] M. Perc, J.G. Gardeñes, A. Szolnoki, L.M. Floría, Y. Moreno, Evolutionary dynamics of group interactions on structured populations: a review, J. Royal Soc. Interface 10 (2013) 20120997.
[18] Z. Wang, L. Wang, M. Perc, Degree mixing in multilayer networks impedes the evolution of cooperation, Phys. Rev. 89 (2014) 052813.
[19] D. Helbing, W. Yu, The outbreak of cooperation among success-driven individuals under noisy conditions, PNAS 106 (2009) 3680–3685.
[20] M.A. Nowak, Five rules for the evolution of cooperation, Sci. Mag. 314 (2006) 1560–1563.
[21] R.B. Myerson, Game Theory: Analysis of Conflict, Harvard University Press, Cambridge, 1991.
[22] K.L. Brown, Y. Shoham, Essentials of game theory: a concise multidisciplinary introduction, Synth. Lectures Artif. Intell. Mach. Learn. 2 (2008) 1–88.
[23] R. Porter, E. Nudelman, Y. Shoham, Simple search methods for finding a Nash equilibrium, Games Econ. Behav. 63 (2008) 642–662.
[24] P.J. Reny, A.J. Robson, Reinterpreting mixed strategy equilibria: a unification of the classical and Bayesian views, Games Econ. Behav. 48 (2004) 355–384.
[25] G. Jeronimo, D. Perrucci, J. Sabia, A parametric representation of totally mixed Nash equilibria, Comput. Math. Appl. 58 (2009) 1126–1141.
[26] M.M. Mohtadi, K. Nogondarian, Solving the traveling salesman problem in competitive situations using the game theory, Appl. math. Eng. Manage. Technol. 2 (2014) 311–325.
[27] L.M. Schmitt, C.L. Nehaniv, R.H. Fujii, Linear analysis of genetic algorithms, Theory Comput. Sci. 208 (1998) 111–148.
[28] F. Glover, M. Laguna, Tabu Search, Handbook of Applied Optimization, Oxford University Press, London, 2002.