KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 209–216

AN EVOLUTIONARY MODEL FOR SOLVING MULTIPLAYER NONCOOPERATIVE GAMES

RODICA LUNG(1) AND D. DUMITRESCU(2)

Abstract. Computing equilibria of multiplayer noncooperative normal form games is a difficult computational task. In games having more equilibria mathematical algorithms are not capable to detect all equilibria at a time. Evolutionary algorithms are powerful search tools for solving difficult optimization problems. It is shown how an evolutionary algorithm designed for multimodal optimization can be used for solving normal form games.

1. Introduction

Game theory is one of the fields of mathematics that has the largest impacts in the economic and social fields.

What economists call game theory psychologists call the theory of social situations, which is an accurate description of what game theory is about [1]. There are two main branches of game theory: *cooperative* and *non cooperative* game theory. Non cooperative game theory deals largely with how intelligent individuals interact with one another in an effort to achieve their own goals. That is the branch of game theory discussed here.

Solving multiplayer normal form games presenting multiple Nash equilibria is a difficult task that is addressed here by using evolutionary algorithms (EAs). The aim is to show that EAs can be used to detect multiple solutions of a game by transforming the game into a multimodal optimization problem.

2. Prerequisites

Notations and basic notions related to game theory that are necessary for this work are presented in this section.

A finite strategic game is defined by $\Gamma = ((N, S_i, u_i), i = \overline{1, N})$ where:

- N represents the number of players;

2000 *Mathematics Subject Classification.* 91A06, 91A10,68T20.
Key words and phrases. Nash equilibria, evolutionary multimodal optimization.

- for each player $i \in \{1, ..., N\}$, S_i represents the set of actions available to him, $S_i = \{s_{i1}, s_{i2}, ..., s_{im}\}$; $S = S_1 \times S_2 \times ... \times S_N$ is the set of all possible situations of the game;
- for each player $i \in \{1, ..., N\}$, $u_i : S \to R$ represents the payoff function.

The following notations are based on [6]

Let $\mathcal{P}_i$ be the set of real valued functions on S_i. The notation $p_{ij} = p_i(s_{ij})$ is used for elements $p_i \in \mathcal{P}_i$.

Let $\mathcal{P} = \times_{i=1,...,N}\mathcal{P}_i$ and $m = \sum_{i=1}^{n} m_i$. Then $\mathcal{P}$ is isomorphic to R^m.

We denote elements in $\mathcal{P}$ by $P = (P_1, P_2, ..., P_N)$ where $P_i = (p_{i1}, p_{i2}, ..., p_{im_i})$.

If $p \in \mathcal{P}$ and $P_i' \in \mathcal{P}_i$ then (P_i', P_{-i}) stands for the element $Q \in \mathcal{P}$ that satisfies $Q_i = P_i'$ and $Q_j = P_j$ for $j \neq i$.

Let Δ_i be the set of probability measures on S_i. We define $\Delta = \times_{i=1,...,N}\Delta_i$. Elements $p_i \in \Delta_i$ are real valued functions on S_i: $p_i : S_i \to R$ and it holds that

$$\sum_{s_{ij} \in S_i} p_i(s_{ij}) = 1,\ p_i(s_{ij}) \geq 0,\ \forall s_{ij} \in S_i.$$

We use the abusive notation S_{ij} to denote the strategy $P_i \in \Delta_i$ with $p_{ij} = 1$. Hence, the notation (S_{ij}, P_{-i}) represents the strategy where player i adopts the pure strategy S_{ij} and all other players adopt their components of P.

The payoff function u_i is extended to have domain R^m by the rule

$$u_i(P) = \sum_{s \in S} P(s) u_i(s),$$

where

$$P(s) = \prod_{i=1}^{N} P_i(s_i).$$

A strategy profile $P* = (P_1*, P_2*, ..., P_N*) \in \Delta$ is a *Nash equilibrium (NE)* if for all $i \in \{1, ..., N\}$ and all $P_i \in \Delta_i$, we have

$$u_i(P_i, P_{-i}*) \leq u_i(P*).$$

Thus a strategy profile $P*$ is a Nash equilibrium if no player can unilaterally increase its payoff when all the other keep theirs unchanged. A normal form game can present more than one NE. Nash [8] proved that there exists at least one NE for any normal form game.

The problem of finding the NEs of a normal form game can be formulated as the problem of detecting all the global minima of a real valued function [5]. This function is constructed using three functions: x, z and g, all defined on $\mathcal{P}$ and having values in R^m. We define the ijth value for this functions for any $p \in \mathcal{P}$, $i \in \{1, ..., N\}$ and $S_{ij} \in S_i$ as

$$x_{ij}(P) = u_i(S_{ij}, P_{-i}) \quad (1)$$
$$z_{ij}(P) = x_{ij}(P) - u_i(P) \quad (2)$$
$$g_{ij}(P) = \max(z_{ij}(P), 0) \quad (3)$$

We define the real valued function $v : \Delta \rightarrow R$ by

$$v(P) = \sum_{i=1}^{N} \sum_{j=1}^{m_i} (g_{ij}(P)^2).$$

Function v is continuous, differentiable and satisfies the inequality $v(P) \geq 0$ for all $P \in \Delta$.

A strategy profile $P*$ is a NE if and only if is a global minimum of v, i.e. $v(P^*) = 0$ [5, 4].

3. Evolutionary multimodal optimization

The main problem in dealing with multimodal optimization is to detect and preserve both local and global solutions.

Over the years, various population diversity mechanisms have been proposed that enable Evolutionary algorithms (EAs) to evolve and maintain a diverse population of individuals throughout its search, so as to avoid convergence of the population to a single peak and to allow EAs to identify multiple optima in a multimodal domain. However, various current population diversity mechanisms have not demonstrated themselves to be very efficient as expected. The efficiency problems, in essence, are related to some fundamental dilemmas in EAs implementation. Any attempt of improving the efficiency of EAs has to compromise these dilemmas, which include:

- The elitist search versus diversity maintenance dilemma: EAs are also expected to be global optimizers with unique global search capability to guarantee exploration of the global optimum of a problem. So the elitist strategy is widely adopted in the EAs search process. Unfortunately, the elitist strategy concentrates on some "super' individuals, reduces the diversity of the population, and in turn leads to the premature convergence.
- The algorithm effectiveness versus population redundancy dilemma: For many EAs, we can use a large population size to improve their effectiveness including a better chance to obtain the global optimum and the multiple optima for a multimodal problem. However, the large population size will notably increase the computational complexity of the algorithms and generate a lot of redundant individuals in the population, thereby decrease the efficiency of the EAs.

Two main evolutionary approaches to multimodality have been adopted:

- *Implicit approaches* that impose an equivalent of either geographical separation or of speciation
- *Explicit approaches* that force similar individuals to compete either for resources or for survival

4. Roaming optimization

A recent evolutionary approach to multimodal optimization called Roaming optimization (RO)[2] is presented.

Within Roaming, the tasks of exploitation and exploration are separated. The first one is performed by a group of elitist individuals belonging to an external population called *the archive* while the second one is realized by subpopulations evolving in isolation.

One of the problems facing multimodal optimization techniques is how to decide when an optimum has been detected. Roaming surpasses this problem by introducing a stability measure for subpopulations. This stability measure enables the characterization of subpopulations as stable or unstable.

A subpopulation is considered stable if no offspring is better in terms of fitness function than the best individual in the parent population. Subpopulations that produce offspring better than the best parent are considered unstable and evolve in isolation until they reach stability.

The best individual in a stable subpopulation is considered to be a potential local optimum and included into the archive using a special archiving strategy.

The number of subpopulations is a parameter of the algorithm and it is not related to the expected number of local optima. This confers flexibility and robustness to the search mechanism.

The archive contains individuals corresponding to different optimum regions. The exploitation task is realized by refining the elite individuals in the archive.

The output of the algorithm is represented by the archive - the set of elitist individuals containing local optima.

5. Deflection technique

The deflection technique [3] is an alternative technique that allows the detection of multiple optima during a single run of an optimization algorithm. Let $f : D \to R$, $D \subset R^n$ be the original objective function under consideration.

Let $x_i^*, i = 1, ..., k$ be the k optima (minima) of f. The deflection technique defines a new function $F : D \to R$ as follows:

$$F(x) = T_1(x; x_1^*, \lambda_1)^{-1} \cdot ... \cdot T_k(x; x_k^*, \lambda_k)^{-1} f(x)$$

where λ_i, $i = 1, ..., k$ are relaxation parameters and T_i, $i = 1, ..., k$ are appropriate functions in the sense that the resulting function has exactly the same optima as f except at points x_i^*, $i = 1, ..., k$.

The functions

$$T_i(x; x_i^*, \lambda_1) = \tanh(\lambda_i ||x - x_i^*||),\ i = 1, ..., k,$$

satisfy this property, known as the deflection property as shown in [3].

When an algorithm detects a minimum x_i^* of the objective function, the algorithm is restarted and an additional $T_i(x; x_i^*, \lambda_i)$ is included in the objective function $F(x)$.

6. Experimental results

Roaming optimization is used to solve several normal form games presenting multiple NE. Results are compared with those obtained by two types of heuristics - differential evolution (DE) and particle swarm optimization (PSO) - adapted to detect multiple solutions using the deflection technique. Experimental set-ups and results regarding DE and PSO presented in [9] are used here. Six variants of DE and two variants of PSO were used.

Results are also compared with those obtained using the state-of-art software GAMBIT (ver. 0.2007.01.31) [7], which computes NE by solving systems of polynomial equations.

6.1. Test problems.

The following test problems presenting multiple NE, available with the GAMBIT software are considered.

GAME1. This is a four players each having two strategies available normal form game. GAME1 has three NE. The corresponding GAMBIT file is $2 \times 2 \times 2 \times 2$.nfg.

GAME2. This is a game with four players, each having two strategies available, having five NE. The corresponding GAMBIT file is g3.nfg.

GAME3. This is a five player game, with two strategies available to each player, having five NE. The corresponding GAMBIT file is $2 \times 2 \times 2 \times 2 \times 2$.nfg.

GAME4. This is a three player game, with two strategies available to each of them, having nine NE. The corresponding GAMBIT file is $2 \times 2 \times 2$.nfg.

6.2. Experimental set-up.

The parameter setting for RO are presented in table 1. Common parameters used to run DE1-6, PSOc and PSOi are presented in table 2.

TABLE 1. Parameter settings for Roaming

Parameter	GAME1	GAME2	GAME3	GAME4
Subpopulations number	10	10	10	30
Size of subpopulations	10	5	5	3
Number of generations	200	200	300	500
Iteration parameter	1	1	1	1

TABLE 2. Parametter settings for DE and PSO

Problem	Pop. size	Iterations/restart	No. restarts
GAME1	20	1000	8
GAME2	20	1000	10
GAME3	50	2000	10
GAME4	10	1000	15

6.3. **Dealing with constraints.** In order for a point $X = (x_{ij})_{i=1,...,N;j=1,...,m_i}$ to be a NE it must satisfy the constraints naturally arising from the condition $X \in \Delta$, which is

$$\sum_{j=1}^{m_i} x_{ij} = 1, \ \forall i = 1, ..., N.$$

To evaluate the fitness of each individual X the following normalization is used:

$$x'_{ij} = \frac{||x_{ij}||}{\sum_{j=1}^{m_i} ||x_{ij}||},$$

which ensures that $X' \in \Delta$. This normalization is used only to compute the fitness value of individuals and not to constraint the population to lie in Δ.

6.4. **Results.** Descriptive statistics presenting the mean, standard deviation, min and max number of NE obtained for each problem by each method over 30 runs are presented in tables 3-6.

7. CONCLUSIONS

Detecting multiple Nash equilibria of multi-player games is a difficult task that most of the times is addressed by applying an algorithm several times. When the number of equilibria or the number of player increases classical approaches are difficult to apply and not always successful.

Evolutionary algorithms designed to detect multiple optima can be used to find Nash equilibria because solving a normal form game is equivalent to finding all the minima of a function constructed from the game.

TABLE 3. GAME1 results - number of NE detected

Technique	Mean	St. Dev	Min	Max
RO	**3**	0	3	3
DE1	2.97	0.18	2	3
DE2	2.93	0.25	2	3
DE3	2.97	0.18	2	3
DE4	3	0	3	3
DE5	3	0	3	3
DE6	3	0	3	3
PSOc	2.97	0.18	2	3
PSOi	3	0	3	3
GAMBIT	3	0	3	3

TABLE 4. GAME2 results - number of NE detected

Technique	Mean	St. Dev	Min	Max
RO	**5**	0	5	5
DE1	4.73	0.45	4	5
DE2	4.30	0.47	4	5
DE3	4.63	0.49	4	5
DE4	4.33	0.48	4	5
DE5	0.87	0.51	0	2
DE6	4.47	0.51	4	5
PSOc	4.67	0.48	4	5
PSOi	4.90	0.31	4	5
GAMBIT	5	0	5	5

TABLE 5. GAME3 results - number of NE detected

Technique	Mean	St. Dev	Min	Max
RO	**5**	0	5	5
DE1	3.10	0.55	2	4
DE2	1.20	0.71	0	3
DE3	3.17	0.75	2	4
DE4	3.03	0.72	2	5
DE5	1.63	0.76	0	3
DE6	2.57	0.82	1	4
PSOc	3.00	0.69	2	4
PSOi	3.37	0.72	2	5
GAMBIT	5	0	5	5

Table 6. GAME4 results - number of NE detected

Technique	Mean	St. Dev	Min	Max
RO	**8.83**	0.46	7	9
DE1	6.70	1.09	4	9
DE2	7.17	1.05	5	9
DE3	7.27	0.87	6	9
DE4	7.90	0.76	7	9
DE5	6.80	1.13	4	9
DE6	7.57	0.90	5	9
PSOc	7.03	0.76	5	9
PSOi	6.90	0.96	5	9
GAMBIT	7	0	7	7

Thus six instances of the Differential Evolutionary algorithm and two of the Particle Swarm Optimization algorithm have been adapted using a deflection technique to detect multiple optima. Results are compared with an evolutionary algorithm designed for multimodal optimization called Roaming optimization.

Numerical experiments indicate that EAs are efficient in this task. Among evolutionary techniques, Roaming proved to have the best results for the test problems taken into account.

References

[1] last accessed May 2006 levine.sscnet.ucla.edu/general/whatis.htm.
[2] Rodica Ioana Lung and D. Dumitrescu. A new subpopulation model for multimodal optimization. In *IEEE proceedings, SYNASC05*, pages 339–342, Timisoara, September 2005.
[3] G. D. Magoulas, M. N. Vrahatis, and G. S. Androulakis. On the alleviation of the problem of local minima in back-propagation. *Nonlinear Anal.*, 30(7):4545–4550, 1997.
[4] R. McKelvey and A. McLennan. Computation of equilibria in finite games. 1996.
[5] R. D. McKelvey. A Liapunov function for Nash equilibria. Technical report, 1991.
[6] Richard D. McKelvey and Andrew McLennan. Computation of equilibria in finite games. In H. M. Amman, D. A. Kendrick, and J. Rust, editors, *Handbook of Computational Economics*, volume 1 of *Handbook of Computational Economics*, chapter 2, pages 87–142. Elsevier, 1 1996.
[7] Richard D. McKelvey, Andrew M. McLennan, and Theodore L. Turocy. Gambit: Software tools for game theory. Technical report, Version 0.2007.01.07, 2006.
[8] John F. Nash. Non-cooperative games. *Annals of Mathematics*, 54:286–295, 1951.
[9] N. G. Pavlidis, K. E. Parsopoulos, and M. N. Vrahatis. Computing nash equilibria through computational intelligence methods. *J. Comput. Appl. Math.*, 175(1):113–136, 2005.

(1) Babeş-Bolyai University
E-mail address: srodica@cs.ubbcluj.ro

(2) Babeş-Bolyai University
E-mail address: ddumitr@cs.ubbcluj.ro