ELSEVIER

Mathematics and Computers in Simulation 53 (2000) 139–147

MATHEMATICS AND COMPUTERS IN SIMULATION

www.elsevier.nl/locate/matcom

Genetic algorithm for closed-loop equilibrium of high-order linear–quadratic dynamic games

Süheyla Özyıldırım*

Faculty of Business Administration, Bilkent University, Bilkent 06533, Ankara, Turkey

Received 1 March 2000; accepted 24 March 2000

Abstract

In this paper, we implement an adaptive search algorithm, *genetic algorithm* to derive closed-loop Nash equilibria for linear–quadratic dynamic games. The computation of these equilibria is quite difficult to deal with analytically and numerically. Our strategy is to search over all time-invariant strategies depending only on the current value of the state. Also provided are some evidences which show the success of the algorithm. © 2000 IMACS. Published by Elsevier Science B.V. All rights reserved.

Keywords: Genetic algorithm; Closed-loop equilibria; Linear feedback rule

1. Introduction

Many developments in game and control theory in the last few decades have caused an increasing interest in using non-zero sum dynamic games for modeling several problems in the area of engineering, mathematics, biology, economics, management science, and political science. One of the basic questions that arises in these models deals with the information of the players during the game. It is this difference in information sets that gives rise to the different solution concepts. In an information-theoretic sense, one solution concept, open-loop corresponds to the receipt of no information during the play while closed-loop (feedback Nash) represents full information. In the latter one cannot just predict the decision rule of the other player and go ahead taking that as a given, one has to take into account the effect that one's own decisions will have on the other player's decision in the future. In reality, neither player can precommit to a time path of future actions. Thus, at each stage of the game, both players reoptimize in light of what happened in the previous stages. Therefore, all new information has to be considered at each stage. It is well-known that the Nash equilibrium of n-person, non-zero sum, linear differential game with quadratic cost function can be expressed in term of the solution of coupled generalized

---

* Tel.: +90-312-2901899; fax: +90-312-2664958.
E-mail address: suheyla@bilkent.edu.tr (S. Özyıldırım).

PII: S0378-4754(00)00155-5

Riccati-type matrix differential equations. For high-order games, the numerical determination of the solution of the non-linear coupled equations is very difficult, and sometimes it is impossible to obtain unique Nash solution [5,7,8,12,13]. However, no-memory restriction on admissible strategies enables alternative solution techniques such as control theory to derive the equilibrium for dynamic games. Cohen and Michel [4] applied control theory to find stable closed-loop Nash equilibrium of ‘appropriately specified’ dynamic game. The aim of this paper is to study application of genetic algorithm (GA) to search feedback Nash solution for high-order linear–quadratic difference games [1] using standard control theory techniques. In general, the task of designing and implementing algorithms for the solution of optimal control problems is difficult one, but the numerical approximation for more than single controller in the problem is even more difficult to handle. Hence, we use both the optimization and the learning property of the GA to solve the problem of multiple criteria.

2. Description of the problem

For presentational simplicity, we will consider two-player dynamic game of the following general form:

$$x_{it+1} = A_i x_{it} + B_i u_t + C_i v_t, \quad i = 1, 2 \tag{1}$$

where u and v are respective control vectors of Players 1 and 2 and A_i, B_i and C_i are matrices of appropriate dimensions. The cost function is given by

$$J_i = \frac{1}{2}\sum_{t=0}^{N} \left(x_{it+1}^{\mathrm{T}} Q_{it+1} x_{it+1} + u_t^{\mathrm{T}} R_{it} u_t + v_t^{\mathrm{T}} S_{it} v_t\right) \tag{2}$$

where the superscript ‘T’ stands for transpose and Q_i, R_i and S_i are symmetric positive definite matrices of appropriate dimensions.

We will consider first the case of open-loop Nash equilibrium. Let $\{u\}_{-t}$ denote the sequence of moves before and after, but not including period t: $u_0, u_1, \dots, u_{t-1}, u_{t+1}, u_{t+2}, \dots, u_N$. An open-loop Nash equilibrium is a sequence $\{u^*\}_0^N$ with the property that:

$$\text{for all } t,\ u_t^* \text{ minimizes } J_i \text{ subject to Eq.(1) and given } \{u^*\}_{-t} \text{ and } \{v^*\}_0^N. \tag{3}$$

In the ‘closed-loop’ version of Nash equilibrium, we assume that Player i plays a *rule* (or strategy) θ_{it}, which maps $(x_{it}, x_{it-1}, \dots)$ to u_t, rather than just a move u_t. As before, define the sequence $\{\theta\}_{-t}$ as $\theta_{i0}, \theta_{i1}, \dots, \theta_{it-1}, \theta_{it+1}, \theta_{it+2}, \dots, \theta_{iN}$, and

$$\begin{aligned}&\{\theta_i^*\}_0^N \text{ is a closed-loop Nash equilibrium if and only if for all } t,\ u_t\\ &\quad = \theta_{it}^*(x_{it}, x_{it-1}, \dots) \text{ minimizes } J_i \text{ subject to Eq. (1) and given } \{\theta_i^*\}_{-t} \text{ and } \{\theta_j^*\}_0^N.\end{aligned} \tag{4}$$

In general, there will be many such Nash equilibria, some of which are not very desirable. In such circumstances, nature of the equilibrium is refined to include only Nash perfect ones. A strategy sequence

---

[1] In continuous time problems (differential games) numerical computation of the feedback Nash solution requires inevitably the use of discretization (or numerical approximation) techniques.

$\{\theta_i\}_0^N$ is said to be perfect equilibrium if for any history the problem from 0 to t, strategies $\{\theta_i\}_0^N$ constitute a Nash equilibrium in the sub-game perfect from 0 to N. We now define time consistency as:

$\{\theta_i\}_0^N$ is time consistent if and only if $\{\theta_i\}_0^N$ is a Nash perfect equilibrium.

Unfortunately, even the perfectness concept does not eliminate the problem of a multiplicity of equilibria. To narrow the search, we will use the simplest case 'memoryless strategies' in which u_t is a function of *current* state vector, x_{it} (see [14] for the justification of the 'memoryless strategies' of this type):

$u_t = \theta(x_{it})$ minimizes J_i subject to Eq. (1) and to the restriction that $u_s = \theta(x_s)$ for all $s \neq t$.

Since the game is linear quadratic, it is natural to consider *only* solutions of the following form:

$$u_t = \theta_t x_{it}$$

where θ_t is the linear feedback rule to be determined in equilibrium. Under these assumptions, Player i's problem becomes

$$\min_{\{\theta_i\}_0^N} J_i = \frac{1}{2}\sum_{t=0}^{N}(x_{it+1}^{\mathrm{T}} Q_{it+1} x_{it+1} + u_t^{\mathrm{T}} R_{it} u_t + v_t^{\mathrm{T}} S_{it} v_t) \tag{5}$$

subject to

$$x_{it+1} = A_i x_{it} + B_i u_t + C_i v_t, \tag{6}$$

$$u_t = \theta_{it} x_{it} \text{ given } x_{i0}. \tag{7}$$

By the description of the model as above, the control theory (via minimum principle) can be applied to compute the equilibrium.[2] One should not minimize the Player 1's cost function subject to the dynamic optimality equation of the Player 2. Instead, one should take as given the equilibrium feedback rule of the Player 2 and postulate that it is independent of the Player 1's actions.

3. Solution procedure

A prime example of direct application of optimal control theory in dynamic game theory is the derivation of conditions for open-loop Nash equilibria in differential games. The discrete time counterpart of the minimum principle is likewise applicable to open-loop Nash equilibria of multi-stage (discrete time) games. In both cases, each player faces a standard optimal problem, which is arrived at by fixing the other players' policies as some arbitrary functions. In principle, the necessary and/or sufficient conditions for open-loop Nash equilibria can be obtained by listing down the conditions required by each optimal control problem (via minimum principle) and then requiring that these all be satisfied simultaneously (see [1,2]). Because of the couplings that exists between these various conditions, each one corresponding to the optimal control problem faced by one player, solving analytically for the Nash equilibria of our game poses a formidable task.

---

[2] see Appendix 3 in [3] for the proof of uniqueness of the linear strategies being the feedback Nash equilibrium in appropriately specified games as above.

One search technique that has been successfully applied to such complex problems is the genetic algorithm. Genetic algorithm is a globally robust search mechanism which combines a Darwinian survival-of-the-fittest strategy to eliminate unfit characteristics and uses random information exchange, with exploitation of the knowledge contained in the previous solutions. Grefenstette [9], Michalewicz and Krawczyk [15] and Krishnakumar and Goldberg [11] used GA to optimize control problems with a single controller. Özyıldırım [16,17] extented GA to solve open-loop difference games of finite horizon. In this paper we implement GA to dynamic games for closed-loop equilibria. The 'appropriately specification' of the games by the introduction of linear feedback rule as in Eq. (7), the solution procedure for open-loop equilibria would also be applicable for the derivation of the feedback Nash ones.

3.1. Genetic algorithm

Genetic algorithm initiated by Holland [10] and further extended by DeJong [6] is best viewed in terms of optimizing a sequential decision process involving uncertainty in the form of lack of a priori knowledge, noisy feedback and time varying payoff function. It is a highly parallel mathematical algorithm that transforms a set of (population) individual mathematical objects (typically fixed length character strings patterned after chromosome strings), each with an associated fitness value, into a new population (i.e. the next generation) using operations patterned after Darwinian principles of reproduction and survival of the fittest after naturally occurring genetic operations.

A GA performs a multi-directional search by maintaining a population of individuals, $P(t) = \{x_1, \dots, x_n\}$ where $x_i = \{x_{i1}, \dots, x_{iT}\}$; each individual, x_i represents a potential solution vector to the problem at hand. An objective function (fitness) plays the role of an environment to discriminate between 'fit' and 'unfit' solutions. The population experiences a simulated evolution: at each generation the relatively 'fit' solutions *reproduce* while the relatively 'unfit' solutions die. During a single reproductive cycle fit individuals are selected to form a pool of candidates some of which undergo *crossover* and *mutation* in order to generate a new population.

Crossover combines the features of two parent chromosomes to form two similar offsprings by swapping corresponding segments of the parents. The intuition behind the applicability of the crossover operator is the information exchange between different potential solutions. Mutation arbitrarily alters one or more genes of a selected chromosome by a random change with a probability equal to the mutation rate *pmut*. The mutation operator introduces additional variability into the population. After some number of generations, the program converges. The best individuals represent the optimum solutions.[3]

3.2. Genetic algorithm for non-cooperative closed-loop dynamic games

Since the closed-loop two-person[4] Nash equilibria can be obtained as the joint solution to two optimal control problems, we will use two parallel GAs to optimize this control system. In the closed-loop Nash game, each player takes as given the feedback rule of the other and these closed-loop rules, $u = \theta_1 x$, $v = \theta_2 x$, are observed in linear feedback form. The substitution of these rules into objective function enables us to apply the algorithm devised for open-loop equilibrium for feedback Nash.

---

[3] For further details, see [9,11,15].

[4] The generalization for n-person game is immediate. The solution procedure described in this section is also valid for games with $n > 2$ players.

```
procedure Player1 GA;
begin
  initialize P1(0)
  randomly initialize
  shared memory;
  synchronize;
  evaluate P1(0);
  t = 1;
  repeat
    select P1(t) from P1(t-1);
    copy best to shared memory;
    synchronize;
    crossover and mutate P1(t);
    evaluate P1(t);
    t=t+1;
  until(termination condition);
end;
```

```
procedure Player2 GA;
begin
  initialize P2(0);
  randomly initialize
  shared memory;
  synchronize;
  evaluate P2(0);
  t = 1;
  repeat
    select P2(t) from P2(t-1);
    copy best to shared memory;
    synchronize;
    crossover and mutate P2(t);
    evaluate P2(t);
    t=t+1;
  until(termination condition);
end;
```

Fig. 1. Parallely implemented genetic algorithm for dynamic games where $Pi(t)$ denotes population of solution for Player i at generation t.

The algorithm stars with the identification of the fitness function for each player. The performance measures of each party is obtained by substituting constraint (6) including the feedback rules (7) into a loss function (5). Then the search over optimal feedback strategies $\{\theta_i^*\}_0^N$ terminates until the following inequalities are satisfied for all players:

$$\begin{aligned} J_1\left(\theta_{10}^*, \dots, \theta_{1N}^*, \theta_{20}^*, \dots, \theta_{2N}^*\right) &\leq J_1\left(\theta_{10}, \dots, \theta_{1N}, \theta_{20}^*, \dots, \theta_{2N}^*\right) \quad \forall \{\theta_1\}_0^N \\ J_2\left(\theta_{10}^*, \dots, \theta_{1N}^*, \theta_{20}^*, \dots, \theta_{2N}^*\right) &\leq J_2\left(\theta_{10}^*, \dots, \theta_{1N}^*, \theta_{20}, \dots, \theta_{2N}\right) \quad \forall \{\theta_2\}_0^N \end{aligned}$$

In this setting, there are two artifically intelligent players (controllers) who update their strategies through GA and a referee, or a fictive player, who administers the parallel implementation of the algorithm and acts as an intermediary for the exchange of best responses. This fictive player (*shared memory*) has no decisive role but provides the best strategies in each iteration to the requested parties *synchronously*. In making his decisions, each player has certain expectations as to what the other players will do. These expectations are shaped through the information received from the shared memory in each iteration.

The Fig. 1 shows the general outline of the algorithm we use for the two-player dynamic game.

In the above algorithm, each side waits for the presence of the previous best structure of the other side in the synchronize statement.

In each step of this algorithm, two GAs are solved. In order to reduce the time complexity, the two GAs are solved for one generation while continuously sharing the best responses. This approach has the advantage that while reducing the time complexity it ensures that the convergence is to the global extremum.

4. Numerical experiments

We have tested the success of the algorithm using three sample difference games studied previously. In this section, we present the results of the genetic algorithm for the closed-loop solutions of these games.

For all tests, the population size is fixed at 50, the runs were made for 200 000 generations. For each test we have made five random runs and reported the average results, it is important to note, however, that the standard deviations of such runs were almost negligibly small. The crossover rate is 0.6. Mutation rate is 0.1. We also used *elitist* selection strategy to stipulate that the structure with the best performance always survives intact into the next generation. Besides, these elitist strategies are the ones sent to the shared memory for information exchanges.

Example 1. A simple numerical example from Kydland [12] for Player $i = 1, 2$ is as follows:

$$\max_{y_{i1}, y_{i2}} \sum_{t=1}^{2} (1 - x_{1t} - x_{2t}) x_{it} - \frac{1}{2} y_{it}^2$$

subject to

$$x_{it} = x_{it-1} + y_{it}, \text{ given } x_{10} = x_{20} = 0.1.$$

Since the objective function has constant term, the feedback rule has also constant. In addition, each player has its own state variable, the estimated feedback rule for Player i will be linear function of its own and other player's state variable. Thus, the problem of the Player i is reformulated as

$$\max_{y_{i1}, \theta_{i2}} \sum_{t=1}^{2} (1 - x_{1t} - x_{2t}) x_{it} - \frac{1}{2} y_{it}^2$$

subject to

$$y_{i2} = \theta_{i0} + \theta_{i1} x_{11} + \theta_{i2} x_{21}$$

$$x_{it} = x_{it-1} + y_{it}, \quad x_{i0} = 0.1.$$

The solutions are as follows:

Player	Kydland		Genetic algorithm			
	y_{i1}	y_{i2}	y_{i1}	θ_{i0}	θ_{i1}	θ_{i2}
1	0.1927	0.0305	0.1927	0.25	−0.625	−0.125
2	0.1927	0.0305	0.1927	0.25	−0.125	−0.625

When we calculate $y_{i2} = \theta_{i0} + \theta_{i1} x_{12} + \theta_{i2} x_{22}$ for $i = 1, 2$, observe that Kydland's solutions for $y_{12} = y_{22} = 0.030475$ will be obtained for both players[5]. This experiment is the most general case in the sense that feedback rule having constant term and more than one state variable.

Example 2. In a general linear–quadratic discrete time Stakelberg game by Vallèe et al. [19], the state variables evolve according to

$$x_{t+1} = 0.8x_t + 0.5u_t + 0.5v_t, \quad \text{given } x_1 = 10,$$

---

[5] In all of our experiments, instead of first period's rule, we search for the first period's control vector (or strategy). Given initial state values, this will decrease the dimension of the search space.

where u, v are, respectively, the control variables of leader (L) and the follower (F). The cost function for each player is:

$$J_{\mathrm{L}} = \min_{\{u\}_1^N} \frac{1}{2}\sum_{t=1}^{N}\left(x_{t+1}^2 + u_t^2 + \frac{1}{2}v_t^2\right)$$

$$J_{\mathrm{F}} = \min_{\{v\}_1^N} \frac{1}{2}\sum_{t=1}^{N}\left(\frac{1}{5}x_{t+1}^2 + \frac{1}{2}u_t^2 + v_t^2\right)$$

The problem of the leader becomes

$$J_{\mathrm{L}} = \min_{u_1,\{\theta\}_2^N} \frac{1}{2}\sum_{t=1}^{N}\left(x_{t+1}^2 + u_t^2 + \frac{1}{2}v_t^2\right)$$

subject to

$$u_t = \theta_{\mathrm{L}t} x_t$$

$$x_{t+1} = 0.8x_t + 0.5u_t + 0.5v_t,$$

We studied this game for two different periods, $N = 2, 3$:

N	Leader				Follower			
	u_1	θ_{L2}	θ_{L3}	J_{L}^*	v_1	θ_{F2}	θ_{F3}	J_{F}^*
2	−4.0682	−0.30770	–	31.472	−0.90140	−0.60150	–	9.655
3	−4.3093	−0.40682	−0.30769	32.505	−1.03931	−40.09014	−0.06154	10.718

The substitution of the optimal rules derived above as feedback form on the state variables, we can say that the procedure works successfully. Above results conform the printed (see [19]) discounted optimal functionals for leader, J_{L}^* and follower J_{F}^* in both periods.

Example 3. A standard two-country macro model under flexible exchange rates from Turnovsky et al. [18]:

$$\min_{m_t}\sum_{t=1}^{12}\rho^{t-1}\,[aY_t^2 + (1-a)(C_{t+1} - C_t)^2]$$

subject to

$$Y_t = \phi_1 m_t + \phi_3 m^* + \phi_3 s_t$$

$$C_{t+1} - C_t = \eta_1 m_t + \eta_2 m_t^* + \eta_3 s_t$$

$$s_{t+1} = cs_t + bm_t - bm_t^*, \quad \text{given}\, s_1 = 1,$$

where $*$ is used to imply other player's variables. Turnovsky et al. proves that for T-period dynamic games of the above problem is unique and linear function of current state variable, s

$$m_t = \theta_\tau s_t$$

$$m_t^* = \theta_\tau s_t \; \tau = T - t; \quad t = 1, 2, \ldots, T \quad \text{(for simulations, } T = 12\text{)}.$$

Using the parameters given in the paper: $\phi_1 = 0.637099$, $\phi_2 = 0.117618$, $\phi_3 = 0.519481$, $\eta_1 = -0.062436$, $\eta_2 = 0.628473$, $\eta_3 = 0.909091$, $a = 0.75$, $b = -1.35065$, $c = 2.298701$, $\rho = 0.9$, the optimal rules are as follows:

$$\begin{array}{ll} \theta_t = -\theta_t^* = 0.6847 & t = 1, \ldots, 8 \\ \theta_9 = -\theta_9^* = 0.6865 & \\ \theta_{10} = -\theta_{10}^* = 0.6873 & \\ \theta_{11} = -\theta_{11}^* = 0.6439 & \\ \theta_{12} = -\theta_{12}^* = -0.9036 & \end{array}$$

The above results are exactly the same solutions derived in the original paper [18].

5. Conclusion

The problem of minimizing quadratic form subject to linear constraints is nearly as old as mathematical physics itself [3]. In this paper, we introduce an alternative solution procedure for feedback Nash scenarios in dynamic games of linear–quadratic form. Our focus has been methodological, so we have experimented genetic algorithm for two-player multi-stage games. The generalization to n-player only requires implementation of n-parallel GA to derive the equilibria (see [16]).

A GA performs a multi-directional search by maintaining population of potential solutions and encouraging information formation and exchanges between directions. In game theoretic frameworks since we used both optimization and learning property of GA, this algorithm provided its value in the computation of closed-loop solutions which have various applications in different fields.

Acknowledgements

The author is extremely grateful to Professor Nedim Alemdar for extensive conversations and suggestions about the solutions of closed-loop equilibria.

References

[1] T. Başar, A counterexample in linear–quadratic games: existence of nonlinear Nash solutions, J. Optimization Theory Appl. 14 (1974) 25–43.
[2] T. Başar, Dynamic Games and Applications in Economics, Springer, Berlin, 1986.
[3] J. Casti, The linear–quadratic control problem: some recent results and outstanding problems, SIAM Rev. 22 (1980) 459–485.
[4] D. Cohen, P. Michel, How should control theory be used to calculate a time-consistent government policy? Rev. Econ. Studies 55 (1988) 263–274.

[5] J.B. Cruz, C.I. Chen, Series Nash solution of two person, nonzero-sum, linear–quadratic differential games, J. Optimization Theory Appl. 7 (1971) 240–257.
[6] K.A. DeJong, Adaptive system design: a genetic approach, IEEE Trans. Syst. Man Cybernetics SMC 10 (1980) 566–574.
[7] J.C. Engwerda, Asymptotic analysis of linear feedback Nash equilibria in nonzero-sum linear–quadratic differential games, J. Optimization Theory Appl. 101 (1999) 693–722.
[8] J.C. Engwerda, Feedback Nash equilibria in the scalar infinite horizon LQ-game, Automatica 36 (2000) 135–139.
[9] J.J. Grefenstette, Optimization of control parameters for genetic algorithm, IEEE Trans. Syst. Man Cybernetics SMC 16 (1986) 122–128.
[10] J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan, Ann Arbor, 1975.
[11] K. Krishnakumar, D.E. Goldberg, J. Guidance, Control system optimization using genetic algorithm, Control Dyn. 15 (1992) 735–738.
[12] F. Kydland, Noncooperative and dominant player solutions in discrete dynamic games, Int. Econ. Rev. 16 (1975) 321–335.
[13] D.L. Lukes, Equilibrium feedback control in linear games with quadratic costs, SIAM J. Control Optimization 9 (1971) 234–252.
[14] E. Maskin, J. Tirole, A theory of dynamic oligopoly. I, Econometrica 56 (1988) 549–569.
[15] Z. Michalewicz, Krawczyk, A modified genetic algorithm for optimal control problems, Comput. Math. Appl. 23 (1992) 83–94.
[16] S. Özyıldırım, Three-country trade relation: a discrete dynamic game approach, Comput. Math. Appl. 32 (1996) 43–56.
[17] S. Özyıldırım, Computing open-loop noncooperative solution in discrete dynamic games, J. Evolutionary Econ. 7 (1997) 23–40.
[18] S.J. Turnovsky, T. Başar, V. d'Orey, Dynamic strategic monetary policies and coordination in interdependent economies, Am. Econ. Rev. 78 (1988) 341–361.
[19] T.V. Vallèe, C.D. Deissenberg, T. Başar, Optimal open loop cheating in dynamic reversed linear–quadratic Stackelberg games, Ann. Operations Res. 88 (1999) 217–232.