Control
and Cybernetics
VOL. 7 (1978) No. 1

Numerical solution of N-person non-zero-sum differential games

by

BOLESŁAW TOŁWIŃSKI
Polish Academy of Sciences
Systems Research Institute, Warszawa

The paper is concerned with numerical methods for finding Nash and Stackelberg equilibria of N-person, non-zero-sum differential games. An algorithm for open-loop solutions is proposed and its convergence is proved for a wide class of differential game problems.

1. Introduction

In the N-person non-zero-sum differential game the i-th player chooses a control u_i ($u_i(t) \in R^{m_i}$, $t \in [t_0, t_f]$) trying to minimize a cost functional

$$J_i = \int_{t_0}^{t_f} L_i(x, u_1, \ldots, u_N, t)\, dt + K_i(x(t_f)) \tag{1}$$

subject to the n-dimensional state equation

$$\dot{x} = f(x, u_1, \ldots, u_N, t), \quad x(t_0) = x_0, \tag{2}$$

and possibly subject to various inequality or equality constraints on the state and/or control variables (these will be not considered in this paper). This problem, which includes the optimal control problem ($N=1$) and the two-person differential game ($N=2$, $J_1 = -J_2$) as special cases, requires a more sophisticated approach to the concept of the optimal solution, when $N \geqslant 2$ and the goals of players are not in total conflict [2, 17, 20, 21]. What one means by the optimal solution depends on the information structure of the game. If a player knows only his own cost functional ignoring those of other players, the min-max strategy v_i, where for all admissible $(u_1, \ldots, u_N)$

$$\max_{\substack{u_j \\ j \neq i}} J_i(u_1, \ldots, v_i, \ldots, u_N) \leqslant \max_{\substack{u_j \\ j \neq i}} J_i(u_1, \ldots, u_i, \ldots, u_N) \tag{3}$$

may be appropriate. For games with a more complete information structure the min-max solution becomes unsatisfactory and other concepts of the optimal solution are needed. In the context of noncooperative, non-zero-sum games two such concepts are considered: Nash equilibrium solution and Stackelberg equilibrium solution, the latter for games with biased information structure.

Assume, that each player knows all functionals J_i, $i=1, \ldots, N$. A natural equilibrium point for such a game is a point $u^*=(u_1^*, \ldots, u_N^*)$ satisfying

$$J_i(u^*) \leqslant J_i(u_1^*, \ldots, u_{i-1}^*, u_i, u_{i+1}^*, \ldots, u_N^*) \tag{4}$$

for all admissible u_i and $i=1, \ldots, N$.

The point u^* is called a Nash equilibrium solution for the game (1)–(2) [21].

Consider now a two player ($N=2$) non-zero-sum game, where the first player knows only his own cost functional, but the other one knows both J_1 and J_2. Assume furthermore, that the second player announces his strategy to his rival before the latter makes his own decision. Let T be a mapping transforming the set of all admissible strategies of player 2 into the set of all admissible strategies of player 1 in such a way, that for any admissible u_2

$$J_1(Tu_2, u_2) \leqslant J_1(u_1, u_2) \tag{5a}$$

for all admissible u_1.

Clearly, Tu_2 is an optimal response of the player 1 to the strategy u_2 of the player 2. (We will assume, that for all u_2 such a response exists and is unique). Now, let u_{2s} be a control satisfying

$$J_2(Tu_{2s}, u_{2s}) \leqslant J_2(Tu_2, u_2) \tag{6}$$

for all admissible u_2.

Then the point (u_{1s}, u_{2s}), where $u_{1s}=Tu_{2s}$, is called a Stackelberg solution of the game (1)–(2) with player 2 as a leader and player 1 as a follower [2, 18, 19].

The concept of the Stackelberg solution can be generalized for the case of $N \geqslant 2$ players, if these players are divided into two groups: $A=\{1, \ldots, N_1\}$—followers, and $B=\{N_1+1, \ldots, N\}$—leaders, where $1 \leqslant N_1 < N$ [17]. We assume, that each follower knows only functionals $J_1, \ldots, J_{N_1}$, but each leader knows all functionals $J_1, \ldots, J_N$ and, that leaders announce their strategies to followers before these can make their own choice. Now, $T=(T_1, \ldots, T_{N_1})$ will be a mapping with the property, that for any admissible $v=(u_{N_1}, \ldots, u_N)$

$$J_i(Tv; v) \leqslant J_i(T_1 v, \ldots, T_{i-1} v, u_i, T_{i+1} v, \ldots, T_{N_1} v; v) \tag{5b}$$

$i=1, \ldots, N_1$, for all admissible u_i.

It follows, that $Tv=(T_1 v, \ldots, T_{N_1} v)$ is an "optimal response" of the followers to the strategies $v=(u_{N_1}, \ldots, u_N)$ of the leaders. Note, that in this case the "optimal response" means a Nash solution of the N_1—person game, played by the followers after the leaders have announced their strategies. Let us assume, that for all v there exists a unique point Tv. Obviously, the mapping T reduces the original game to a N-N_1—person game played by leaders, where cost functionals are of the form:

$$\bar{J}_i(v) = J_i(Tv; v), \quad i=N_1+1, \ldots, N. \tag{7}$$

Let v_s be a Nash point of this game. Then the strategies (u_s, v_s), where $u_s = Tv_s$, are called a Stackelberg solution of the game (1)–(2) with group B as Nash leaders and group A as Nash followers.

In this paper we are mainly concerned with computational aspects of the differential game theory. The development of computational techniques for obtaining various equilibria of non-zero-sum differential games is crucial for their potential applications in techniques and economics. The problem of obtaining the min-max solution is equivalent to solving a zero-sum, two-person differential game and it can be done by some of the methods proposed in Refs. [1, 15, 24]. So we have to focus our attention on the numerical techniques for obtaining the Nash and Stackelberg equilibria of N-person, non-zero-sum differential games. This subject has been investigated by a number of authors, but is far from being exhausted. A solution for linear-quadratic games (functions L_i, K_i are quadratic and f is linear) has been worked out by Starr and Ho [21], Foley and Schmitendorf [4] (Nash equilibrium) and Simaan and Cruz [17, 18] (Stackelberg equilibrium). Heuristic algorithms for a Nash solution of general, non-linear-quadratic games has been proposed by Starr [20], Sage [16], Holt and Mukundan [5], Pau [13, 14] and Mukundan and Elsner [11]. From these the Holt-Mukundan procedure seems to be the most relevant for practical computations. It has however a drawback of being very sensitive to the choice of an initial reference solution and may easily fail to converge, even for the relatively simple problems. In Section 3 of this paper we present an algorithm, which for a wide class of differential games does not have this drawback, while preserving the good performance of the Holt-Mukundan procedure near optimum.

It is important to note, that except for the case of Nash solution of linear-quadratic games, all methods mentioned above are concerned only with open-loop equilibria. As it has been demonstrated in [19–22] closed-loop equilibrium strategies are, in general, extremely difficult to compute and until now there exist no methods for such computations. Some special closed-loop structures, which are relatively easy to compute has been presented in [11] and [16].

2. Necessary conditions for Nash and Stackelberg equilibria

An important fact in the theory of non-zero-sum differential games is, that open-loop and closed-loop Nash, as well as Stackelberg, solutions are, in general, different and may lead to entirely different trajectories and costs [2, 21]. Unfortunately, the necessary conditions for a closed-loop solution include terms $(\partial/\partial x)\, u_j\,(x, t)$, what makes them virtually useless for deriving computational algorithms. One notable exception is the linear-quadratic game, where necessary conditions lead to the system of matrix differential equations of Riccati type [4, 21]. The algorithm presented in the next Section is concerned with a general, non-linear, open-loop game and is derived from the following relations:

(A) Necessary conditions for an open-loop Nash equilibrium solution

Define a Hamiltonian function for each player as

$$H_i(x, u_1, \dots, u_N, p_i, t) = L_i + p_i' f, \quad i=1, \dots, N \tag{8}$$

(The sign prim denotes the transposition of a vector or a matrix).
For an open-loop Nash solution one has[1])

$$\dot{x} = f(x, u_1, \dots, u_N, t), \tag{9}$$

$$x(t_0) = x_0, \tag{10}$$

$$\dot{p}_i' = -\partial H_i / \partial x, \quad i=1, \dots, N, \tag{11}$$

$$p_i'(t_f) = (\partial / \partial x(t_f))\, K_i(x(t_f)), \quad i=1, \dots, N, \tag{12}$$

$$\partial H_i / \partial u_i = 0, \quad i=1, \dots, N. \tag{13}$$

(B) Necessary conditions for an open-loop Stackelberg equilibrium solution

Define a Hamiltonian function for each player from group A as

$$H_i^f(x, u_1, \dots, u_N, p_i, t) = L_i + p_i' f, \quad i=1, \dots, N_1 \tag{14}$$

and for each player from the group B as

$$H_j^l(x, p_1, \dots, p_{N_1}, u_1, \dots, u_N, \beta_{1j}, \dots, \beta_{N_1 j}, \lambda_j, \gamma_{1j}, \dots, \gamma_{N_1 j}) =$$

$$= L_j + \lambda_j' f + \sum_{i=1}^{N} [\gamma_{ij}' (-\partial H_i^f / \partial x)' + \beta_{ij}' (\partial H_i^f / \partial u_i)'] \quad j = N_1 + 1, \dots, N. \tag{15}$$

For an open-loop Stackelberg solution one has [17]

$$\dot{x} = f(x, u_1, \dots, u_N, t), \tag{16}$$

$$x(t_0) = x_0, \tag{17}$$

$$\dot{p}_i' = -\partial H_i^f / \partial x, \quad i=1, \dots, N_1, \tag{18}$$

$$p_i'(t_f) = (\partial / \partial x(t_f))\, K_i(x(t_f)), \quad i=1, \dots, N_1, \tag{19}$$

$$\partial H_i^f / \partial u_i = 0, \quad i=1, \dots, N_1, \tag{20}$$

$$\dot{\lambda}_j' = -\partial H_j^l / \partial x, \quad j = N_1 + 1, \dots, N, \tag{21}$$

$$\lambda_j'(t_f) = (\partial / \partial x(t_f))\, K_j(x(t_f)) - \sum_{i=1}^{N} \gamma_{ij} (\partial^2 / \partial x^2(t_f))\, K_i(x(t_f))$$

$$j = N_1 + 1, \dots, N, \tag{22}$$

$$\dot{\gamma}_{ij}' = -\partial H_j^l / \partial p_i, \quad i=1, \dots, N_1;\ j = N_1 + 1, \dots, N, \tag{23}$$

[1]) Assuming, that the solution sought is in the interior of the set of admissible controls.

$$\gamma_{ij}(t_0)=0,\ i=1,\ldots,N_1;\ j=N_1+1,\ldots,N, \tag{24}$$

$$\partial H_j^1/\partial u_i=0,\ i=1,\ldots,N_1;\ j=N_1+1,\ldots,N, \tag{25}$$

$$\partial H_j^1/\partial u_j=0,\ j=N_1+1,\ldots,N. \tag{26}$$

3. Numerical algorithms

Our purpose is to define a computational procedure generating a sequence of controls and trajectories converging to a desirable point, i.e. to a point satisfying necessary conditions (9)–(13) or (16)–(26). One possibility is to use the so called direct approach. Consider first the Nash problem and assume, that controls u_i can be obtained from the equations (13) as functions of time, t state x and co-state variables $p_1, p_2, \ldots, p_N$. By substituting these functions for u_i in (9) and (11) one arrives to a two-point-boundary-value problem (TPBVP) of the following form:

$$\dot{x}=\bar{f}(x,p,t),\ x(t_0)=x_0, \tag{27}$$

$$\dot{p}_i'=(\partial/\partial x)\,\bar{H}_i(x,p,t),\ p_i'(t_f)=(\partial/\partial x(t_f))\,K_i(x(t_f)),\ i=1,\ldots,N, \tag{28}$$

where $p=(p_1,\ldots,p_N)$

$$\bar{f}(x,p,t)=f(x,u_1(x,p,t),\ldots,u_N(x,p,t)), \tag{29}$$

$$\bar{H}_i(x,p,t)=H_i(x,u_1(x,p,t),\ldots,u_N(x,p,t),p_i,t). \tag{30}$$

The problem (27)–(28) can be solved by some of the particular numerical algorithms developped for the TPBVP [17, 23].

The same method can be used for finding a Stackelberg solution. Assume, that controls u_i, $i=1,\ldots,N$, and Lagrange multipliers β_{ij}, $i=1,\ldots,N_1$, $j=N_1+1,\ldots,N$, can be obtained from equations (20), (25) and (26) as functions of time, state and co-state variables p, λ, and γ. By substituting these functions for u_i and β_{ij} in (16), (18), (21) and (23) one obtains:

$$\dot{x}=\bar{f}(x,p,\lambda,\gamma,t), \tag{31}$$

$$\dot{p}_i'=(-\partial/\partial x)\,\bar{H}_i^f(x,p,\ \ ,\ \ ,t),\ i=1,\ldots,N_1) \tag{32}$$

$$\dot{\lambda}_j'=(-\partial/\partial x)\,\bar{H}_j^1(x,p,\ \ ,\ \ ,t),\ j=N_1+1,\ldots,N, \tag{33}$$

$$\dot{\gamma}_{ij}'=(-\partial/\partial p_i)\,\bar{H}_i^1(x,p,\ \ ,\ \ ,t),\ i=1,\ldots,N_1;\ j=N_1+1\ \ldots,N, \tag{34}$$

with boundary conditions (17), (19), (22) and (24).

The direct method has been successfully used for solving two-person, zero-sum differential games [1, 15]. Clearly, it can be also used to solve non-zero-sum, open-loop games with small number of players. The applicability of the method to the more complex games is limited by the fact, that the dimensionality of problems (27)–(28) and (31)–(34) grows rapidly with n and N and the efficiency, as measured in computer space and time required for computations, of algorithms solving TPBVP is for high dimensional systems rather poor. It is worth to note also, that the convergence of such algorithms depends to great extent on the choice of the initial reference solution and often it may be difficult to make a good guess.

Another possibility for defining a computational procedure solving a differential game is to use the indirect approach. As before, consider first the Nash problem. The system of equations (9)–(13) is equivalent to the following optimal control problem:

Minimize the functional

$$J(u)=(1/2)\int_{t_0}^{t_f}\left(\sum_{i=1}^{N}\|(\partial/\partial u_i)\, H_i(x, u_1, \ldots, uN, p_i, t)\|^2\right) dt \tag{35}$$

subject to the state equations:

$$\dot{x}=f(x, u_1, \ldots, u_N, t),\ \ x(t_0)=x_0, \tag{36}$$

$$\dot{p}_i'=-(\partial/\partial x)\, H_i,\ \ p_i(t_f)=(\partial/\partial x(t_f))\, K_i(x(t_f)),\ \ i=1, \ldots, N. \tag{37}$$

The Hamiltonian of this problem has the form:

$$H(x, p, u, \eta, \delta, t)=(1/2)\sum_{i=1}^{N}\|(\partial/\partial u_i)\, H_i\|^2+\eta' f-\sum_{i=1}^{N}\delta_i'[(\partial/\partial x)\, H_i]' \tag{38}$$

where

$$\dot{\eta}'=-(\partial/\partial x)\, H,\ \ \eta(t_f)=-\sum_{i=1}^{N}\delta_i'(t_f)\,(\partial^2/\partial x^2(t_f))\, K_i(x(t_f)) \tag{39}$$

$$\dot{\delta}_i'=-(\partial/\partial p_i)\, H=-(\partial/\partial u_i\, H_i)\cdot(\partial/\partial u_i)\, f+\delta_i'(\partial/\partial x) f,\ \ \delta_i(t_0)=0,\ \ i=1, \ldots, N. \tag{40}$$

The problem (35)–(37) can be solved, at least in theory, by an arbitrary optimization scheme developed for the optimal control problems. It seems however, that for obtaining an efficient numerical procedure the particular form of the functional (35) should be taken into consideration. Consider a simple algorithm proposed by Holt and Mukundan [5, 11]. It proceeds as follows. An initial set of controls $u_i(t)$, $t\in[t_0, t_f]$, $i=1, \ldots, N$, is guessed. The state equations (36) are integrated forward in time, obtaining $x(t_f)$, which in turn permits the integration of the co-state equations (37). The new controls are calculated by solving the equations (13), or, which is equivalent, by minimizing

$$(1/2)\sum_{i=1}^{N}\|(\partial/\partial u_i)\, H_i\|^2,\ \ t\in[t_0, t_f] \tag{41}$$

and the procedure is restarted. In some methods solving an optimal control problems (the so-called min-H methods) the next approximation to an optimal solution is obtained by minimizing the Hamiltonian of the problem in u (the control variable), using values of the state and co-state variables, which have been computed for the current value of the control u [6, 8, 10]. The Holt-Mukundan procedure may be looked at as the approximation to such a method applied to solve the problem

(35)–(37). It follows from that fact, that in some cases, for a near-optimal trajectory x, the minimization of the Hamiltonian (38) can be approximated by the minimization of the summation term (41) (Note, that if x is desirable for the differential game and, as a consequence, optimal for the optimal control problem, then for all $t \in [t_0, t_f]$ $u(t)$, which minimizes the expression (38) minimizes also the summation term (41) and vice-versa).

Obviously, unlike any gradient method for solving the minimization problem (35)–(37), the Holt-Mukundan procedure permits to avoid the integration of the equations (39) and (40). Computational experiments show, that in some cases the procedure converges in a small number of iterations. As an example consider a two-person game with linear dynamics and non-quadratic cost functionals defined in [11]:

Example 1

$$J_i = (1/2) \int_0^1 [x_i^2 + 0{,}1\,(u_i^2 + u_i^4)]\, dt, \quad i = 1, 2$$

$$\dot{x}_1 = x_2 + u_1, \quad x_1(0) = 1$$

$$\dot{x}_2 = -2x_1 - 0{,}1 x_2 + u_2, \quad x_2(0) = 10.$$

It is easy to verify, that starting from controls $u_1^0 = u_2^0 = 0$ (which are not in a direct neighbourhood of the solution), after two iterations of the Holt-Mukundan procedure one obtains an approximation to the Nash solution of the game, for which the value of the functional (35) is less than 10^{-6}. From the other side, it can be shown, that the procedure may fail to converge even for rather simple linear-quadratic games, no-matter how close an initial reference solution will be to the actual one. Consider the following

Example 2

$$J_1 = (1/2) \int_0^1 u^2(t)\, dt + x(1)\, y(1)$$

$$J_2 = (1/2) \int_0^1 v^2(t)\, dt - x(1)\, y(1)$$

$$\dot{x} = u, \quad \dot{y} = v, \quad x(0) = y(0) = 0,$$

where u is the control of the first player, v is the control of the second player, $(x, y)'$ is the state vector and $u(t)$, $v(t)$, $x(t)$ and $y(t)$ are scalar variables defined for $t \in [0, 1]$.

The Hamiltonians of the problem are defined as:

$$H_1 = (1/2)\, u^2 + p_1 u + p_2 v$$

and

$$H_2=(1/2)\,v^2+q_1\,u+q_2 v,$$

where $(p_1, p_2)'$ and $(q_1, q_2)'$ are co-state vectors.
One has:

$$\dot{p}_1=\dot{p}_2=\dot{q}_1=\dot{q}_2=0,\ p_1(1)=y(1),\ p_2(1)=x(1),\ q_1(1)=-y(1)$$

and $q_2(1)=-x(1)$.

Consequently

$$\partial H_1/\partial u=u+y\,(1)$$

and

$$\partial H_2/\partial v=v-x\,(1).$$

Let x^k and y^k be the trajectories computed for a reference solution (u^k, v^k). By applying the method of Holt and Mukundan one obtains, that for all $t\in[t_0, t_f]$

$$(u^{k+1}(t), v^{k+1}(t))'=\text{const.}=\left(-y^k(1), x^k(1)\right)'.$$

Now, let c be any constant number and take $u^o(t)=v^o(t)\equiv c$. Obviously, one has:

$$\begin{bmatrix}u^1\\ v^1\end{bmatrix}=\begin{bmatrix}-c\\ c\end{bmatrix},\ \begin{bmatrix}u^2\\ v^2\end{bmatrix}=\begin{bmatrix}-c\\ -c\end{bmatrix},\ \begin{bmatrix}u^3\\ v^3\end{bmatrix}=\begin{bmatrix}c\\ -c\end{bmatrix},\ \begin{bmatrix}u^4\\ v^4\end{bmatrix}=\begin{bmatrix}c\\ c\end{bmatrix}$$

i.e.

$$\begin{bmatrix}u^4\\ v^4\end{bmatrix}=\begin{bmatrix}u^o\\ v^o\end{bmatrix}.$$

It follows, that the procedure fails to converge, no-matter how close an initial reference solution will be to the unique Nash point $(u^*, v^*)=(0, 0)$ of the game (excluding the trivial case, when $(u^o, v^o)=(u^*, v^*)$).

If for a given reference solution the Holt-Mukundan procedure does not converge, one can integrate the equations (39) and (40) and use a gradient method to minimize the functional (35). By means of such a method it is possible to obtain a required approximation to a Nash point, or, if only a small number of iterations (one for example) will be performed—a new reference solution for the Holt-Mukundan procedure. It can be argued, that in many cases the latter approach will be computationally more efficient. A simple algorithm using this approach is defined below.

Algorithm

Step 1. Select a real number $0<\varepsilon<1$ and an initial reference solution $u^o=(u_1^o, ..., u_N^o)\in G$, where G is a set of admissible controls.

Step 2. Set $u=u^o$ and $k=0$.

Step 3. Compute x by integrating (36) forward in time.

Step 4. Compute $p=(p_1, ..., p_N)$ by integrating (37) backward in time.

Step 5. Compute $J=J(u)$ using (38).
Step 6. If $J=0$ stop (a desirable point is found); else go to Step 7.
Step 7. For all $t\in[t_0, t_f]$ compute

$$\bar{u}(t)=\arg\min \sum_{i=1}^{N} \|(\partial/\partial u)\, H_i(x, u, p, t)\|^2.$$

If $\bar{u}\in\tilde{G}$ and

$$\int_{t_0}^{t_f}\left(\sum_{i=1}^{N} \|(\partial/\partial u_i)\, H_i(x, \bar{u}, p_i, t)\|^2\right) dt=0$$

then go to Step 8; else go to Step 12.
Step 8. Compute x from the equation

$$\dot{\bar{x}}=f(\bar{x}, u, t),\ \ \bar{x}(t_0)=x_0.$$

Step 9. Compute $\bar{p}=(\bar{p}_1, ..., \bar{p}_N)$ from the equations

$$\dot{\bar{p}}_i'=-(\partial/\partial x)\, H_i(\bar{x}, \bar{u}, \bar{p}_i, t),\ \ \bar{p}_i'(t_f)=(\partial/\partial x(t_f))\, K_i(x(t_f)),$$

$i=1, ..., N$.
Step 10. Compute $\bar{J}=J(\bar{u})$.
Step 11. If $\bar{J}-J\leqslant -\varepsilon J$ then go to Step 19; else go to Step 12.
Step 12. Compute δ_i, $i=1, ..., N$, by integrating (40) forward in time.
Step 13. Compute η by integrating (39) backward in time.
Step 14. Compute $\nabla J(u)=\partial H/\partial u)$.
Step 15. If $\|\nabla J(u)\|=0$ stop (The algorithm converges to a stationary point of J, which is not the desirable point of the game (1)–(2)); else go to Step 16.
Step 16. Compute

$$\alpha_0=\arg\min_{\alpha} J(u-\alpha\nabla J(u)).$$

Step 17. Set $\bar{u}=u-\alpha_0\nabla J(u)$.
Step 18. Compute $\bar{x}, \bar{p}$ and $\bar{J}$ as in Steps 8–10.
Step 19. Set $k=k+1$ and $u^k=\bar{u}$.
Step 20. Set $u=\bar{u}$, $x=\bar{x}$, $p=\bar{p}$, $J=\bar{J}$ and go to Step 6.

To formulate the properties of the algorithm we shall need some assumptions about the set of admissible controls G and the functions f, L_i, K_i, which define the differential game (1)–(2). Define a set G as

$$G=\{u\colon [t_0, t_f]\to\Omega \mid u \text{ is continuous except at a countable number of points}\} \tag{42}$$

where Ω is a closed, bounded and convex subset of $R^m=R^{m_1}\times...\times R^{m_N}$.
Let $\tilde{G}$ be the set of equivalence classes of functions in G, which are equal almost everywhere. The algorithm seeks a desirable point $u^*\in\tilde{G}$, such, that for almost all $t\in[t_0, t_f]$ $u^*(t)\in \operatorname{int}\Omega$. All results given below and concerned with the convergence of the algorithm are valid only, if all points u^k generated by it belong to the set $\tilde{G}$. In practice, with an appropriate choice of Ω, this condition will be usually

satisfied. Alternatively, the gradient $\nabla J(u)$ as the direction of minimization in the step 16 may be replaced by its projection on the set $\tilde{G}$.

Denote

$$S=\{(x, u, t) \mid x \in R^n, \ u \in \Omega, \ t \in [t_0, t_f]\}. \tag{43}$$

We will use the following hypotheses:

(H1) $f: R^n \times R^m \times [t_0, t_f] \to R^n$,

$L_i: R^n \times R^m \times [t_0, t_f] \to R, \ i=1, \ldots, N$

and their partial derivatives of first, second and third order with respect to x and u exist and are continuous on S.

$K_i: R^n \to R, \ i=1, \ldots, N$

and its derivatives K_{ix}, K_{ixx}, and K_{ixxx} are continuous on R^n.

(H2) $f(x, u, t) \leqslant M(\|x\|+1), \ \forall (x, u, t) \in S$, for some $M<\infty$.

(H3) $f(x, u, t)=A(t)x+B(t)u$,

$(\partial/\partial x) L_i(x, u, t)=C_i(t)x+D_i(t)u, \ i=1, \ldots, N$,

for some matrix valued functions A, B, C_i and D_i.

$\|\partial L_i/\partial u_i\|^2, \ i=1, \ldots, N$, are convex in x and strictly convex in u, for $x \in R^n$ and $u \in \Omega$.

THEOREM 1. Let (H1) and (H2) be satisfied. Suppose $\{u^k\}$ is a sequence generated by the algorithm. Then, either the sequence is finite, in which case its last element is a stationary point of the functional (35), or it is infinite and

$$\lim_{k\to\infty} \|\nabla J(u^k)\|_2=0. \tag{44}$$

Proof. The sequance $\{u^k\}$ can be finite only, if for some $k=\hat{k}$

$$J(u^{\hat{k}})=0 \ \text{or} \ \|\nabla J(u^{\hat{k}})\|=0.$$

In both cases $u^{\hat{k}}$ is a stationary point of the functional J.

Suppose now, that the sequance $\{u^k\}$ is infinite and, that the inequality from the Step 11 of the algorithm has been satisfied s times in k iterations. Then one has

$$J(u^k) \leqslant (1-\varepsilon)^s J(u^o). \tag{45}$$

Obviously, if s grows to infinity with k, then

$$\lim_{k\to\infty} J(u^k)=0=\inf_u J(u), \ \text{what implies also} \ \lim_{k\to\infty} \|\nabla J(u^k)\|=0.$$

The boundness of s for all k means, that almost all elements of the sequence $\{u^k\}$ are obtained by the steepest descent method. It is a known result (see for example Ref. [3]), that a sequence obtained in this way has the property (44), if the gradient of the functional is Lipschitz-continuous and all u^k belong to a bounded set. As all $u^k \in \tilde{G}$, to complete the proof of the theorem it is enough to show, that the condition

$$\|J(u)-\nabla J(\bar{u})\|_2 \leqslant L\|u-u\|_2, \ \forall u, \bar{u} \in \tilde{G} \tag{46}$$

is satisfied for some $L<\infty$. This, however, follows from the hypotheses (H1) and (H2) and can be obtained by a repeated use of the Bellman-Gronwall and Schwartz inequalites. Note, that because of the assumption (H2) solutions of the equation (2) exist, are differentiable except at a countable number of points, and uniformly bounded for $u \in \tilde{G}$ [10]. The detailed proof of (46) is lengthy and in many aspects similar to the proofs given in Appendix A of Ref. [10], therefore it is omitted here.

THEOREM 2. Let (H1) and (H3) be satisfied. Then the sequence $\{u^k\}$ generated by the algorithm converges in L_2 metric to a point u^*, which is a unique minimum point of the functional (35) on the set $\tilde{G}$. If $J(u^*)=0$, then u^* is a desirable point for the differential game (1)–(2). If $J(u^*)>0$, then there exist no Nash solution of the game (1)–(2).

Proof. It follows from (H3), that the expression

$$\sum_{i=1}^{N} \|(\partial/\partial u_i)\, H_i(x, u, p_i, t)\|^2$$

is convex in $x \in R^n$, strictly convex in $u \in \Omega$ and, as the set is bounded, also uniformly convex in $u \in \Omega$. From it and from the linearity of the state equations (36) and (37) one gets [9], that the functional (35) is uniformly convex in $u \in \tilde{G}$. The uniformly convex and continuous functional J defined in the Hilbert space L_2 has a unique minimum point u^* on the convex, closed and bounded set $\tilde{G} \in L_2$ [25]. Furthermore, for the convex functional J and the bounded set $\tilde{G}$, it follows from the property (44) [3], that

$$\lim_{k\to\infty} J(u^k) = \min_{u \in G} J(u) = J(u^*)\,. \tag{47}$$

Finally, for the uniformly convex functional J and convex, closed and bounded set G, every sequence with property (47) converges in L_2 metric to the point u^* [25]. So, as (H3) implies also (H2), the proof of the theorem is completed.

Note, that if the assumptions of the Theorem 2 are satisfied (as in the case of the linear-quadratic game, for example), then the algorithm finds a desirable point of the differential game (1)–(2), if such a point exists, or states explicitly its non-existence, and consequently, the non-existence of any Nash solution for a given game. It is worth to note also, that the results proved remain valid, if the conceptual rule for choosing the step length α_0, from the Step 16, is replaced by the easily implementable Armijo's rule [3]. With this and some other, rather obvious modifications an implementable version of the algorithm can be easily obtained.

Consider now the Stackelberg equilibrium solution. As before the system of necessary conditions is equivalent to an optimal control problem. Unfortunately this problem is in general case by far more difficult to solve numerically than the problem (35)–(37). It follows from the fact, that this time the state and co-state equations form usually TPBVP which can t be decomposed into separate one-point-boundary-value problems. Note however, that such decomposition is possible if $K_i(x)=0$ for all x and $i=1, ..., N$. In this case from the equations (16)–(26) one obtains the following problem

Minimize the functional

$$J(u,\beta)=\int_{t_0}^{t_f} S dt \tag{48}$$

where

$$S=(1/2)\left\{\sum_{i=1}^{N_1}\|\partial H_i^f/\partial u_i\|^2+\sum_{j=N_1+1}^{N}\|\partial H_j^l/\partial u_j\|^2+\sum_{i=1}^{N_1}\sum_{j=N_1+1}^{N}\|\partial H_j^l/\partial u_i\|^2\right\} \tag{49}$$

subject to state equations

$$\dot{x}=f,\ x(t_0)=x_0 \tag{50}$$

$$\dot{p}_i'=-\partial H_i^f/\partial x,\ p_i(t_f)=0 \tag{51}$$

$$\dot{\lambda}_j'=-\partial H_j^l/\partial x,\ \lambda_j(t_f)=0 \tag{52}$$

$$\gamma_{ij}'=-\partial H_j^l/\partial p_i,\ \gamma_{ij}(t_0)=0 \tag{53}$$

where $i=1,\dots,N_1$ and $j=N_1+1,\dots,N$.

The Hamiltonian of this problem has the form:

$$H=S+\eta_1' f-\sum_i(\eta_2^i)'(\partial H_i^f/\partial x)-\sum_j(\eta_3^j)'(\partial H_j^l/\partial x)+\sum_{i,j}(\eta_4^{ij})'(\partial H_j^l/\partial p_i) \tag{54}$$

where

$$\dot{\eta}_1'=-\partial H/\partial x,\ \eta_1(t_f)=0 \tag{55}$$

$$(\dot{\eta}_2^i)'=-\partial H/\partial p_i=-\partial S/\partial p_i+(\eta_2^i)' f_x+\sum_j(\eta_3^j)'[(-f_x)+\beta_{ij}f_{u_i}]\eta_2^i(t_0)=0 \tag{56}$$

$$(\eta_3^j)'=-\partial H/\partial\lambda_j=-\partial S/\partial\lambda_j+(\eta_3^j)' f_x,\ \eta_3^j(t_0)=0 \tag{57}$$

$$(\eta_4^{ij})'=-\partial H/\partial\gamma_{ij}=-\partial S/\partial\gamma_{ij}+(\eta_3^j)'(-\partial H_i^f/\partial x)-(\eta_4^{ij})' f_x\,\eta_4^{ij}(t_f)=0 \tag{58}$$

$i=1,\dots,N_1$ and $j=N_1+1,\dots,N$.

It is clear, that differential equations (50)–(53) and (55)–(58) can be integrated separately in the following order: (50), (51), (53), (52), (57), (58), (56) and (55). So, the problem (48)–(53) is analogous to the problem (35)–(37) and can be solved by the same algorithm with obvious modifications.

THEOREM 3. If to the hypothese (H1) one adds the condition that the fourth-order derivatives of f and L_i with respect to x and u exist and are continuous, then all results obtained for the Nash problem (Theorems 1 and 2) hold also for the Stackelberg problem with $K_i=0$.

Proof. Under the new stronger regularity assumptions the gradient of the functional (48) is Lipschitz-continuous, what implies the proposition of the Theorem 1. Furthermore, it is easy to check, that, if (H3) is satisfied, then the functional (48) is uniformly convex—The state equations (50)–(53) are linear and S is convex in $x, p_i, \lambda_j, \gamma_{ij}$ and strictly convex in $u\in\Omega$. This fact completes the proof.

4. Conclusion

The research in the field of non-zero-sum dynamic games is of considerable importance in view of their possible applications in economics (see Refs. [12, 13]). It seems, that the game framework is in many cases more suitable for economic modelling than the classical optimization approach, which does not reflect the multiplicity of objectives and inevitable conflicts arising in almost all economic problems. In this paper we have presented numerical techniques for obtaining open-loop Nash and Stackelberg equilibria. The algorithm defined in Section 3 solves variety of continuous, as well as discrete-time, dynamic game problems.

References

1. Anderson G. M.: A near optimal closed-loop solution method for non-singular zero-sum differential games. fi. *Opt. Theory a. Appl.* **13**, 3 (1974).
2. Chen C. I., Cruz J. B.: Stackelberg solution for two-person games with biased information patterns. *IEEE Trans. Autom. Contr.* **AC-17**, 6 (1972).
3. Daniel J. W.: The approximate minimization of functionals. Englewood Cliffs N. J. 1971.
4. Foley M. H., Schmitendorf W. E.: On a class of non-zero-sum differential games. *J. Opt. Theory a. Appl.* **7**, 5 (1971).
5. Holt D., Mukundan R.: A Nash algorithm for a class of non-zero-sum differential games. *Int. J. Syst. Sci.* **2**, 4 (1972).
6. Jacobson D. H., Mayne D. Q.: Differential dynamic programming. New York 1970.
7. Keller H. B.: Numerical methods for two point boundary value problems. Waltham, Mass. 1968.
8. Kryłow I. A., Czernouszko F. L.: Ałgoritm mietoda posledowatielnych pribliżenij dla zadacz optimalnogo uprawlenija. *Ż. wyczysl. matiem. i matiem. fiz.* **11**, 1 (1972).
9. Levitin E. S., Polak B. T.: Mietody minimizacji pri naliczi ograniczenij. *Ż. wyczysl. matiem. i matiem fiz.* **6**, 5 (1966).
10. Mayne D. Q., Polak E.: First-order strong variation algorithms for optimal control. *J. Opt. Theory a. Appl.* **16**, 3–4 (1975).
11. Mukundan R., Elsner W. B.: Linear feedback strategies in non-zero-sum differential games. *Int. J. Syst. Sci.* **6**, 6 (1975).
12. Myoken H.: Non-zero-sum differential games for the balance of payments adjustment in an open economy. *Int. J. Syst. Sci.* **6**, 6 (1975).
13. Pau L. F.: Differential game among sectors in a macroeconomy. IFAC/IFORS Int. Conf. on Dynamic Modelling a. Control of National Economies. Conf. publication No. 101, 1973.
14. Pau L. F.: Differential games and a Nash equilibrium searching algorithm. *SIAM J. Contr.* **13**, 4 (1975).
15. Quintana V. H., Davison E. J.: Two numerical techniques to solve differential game problems. *Int. J. Contr.* **16**, 3 (1972).
16. Sage A. P.: Gradient methods for optimization of differential game strategies. Annual Southwestern IEEE Conference Exhibition 22nd, New York 1970.
17. Simaan M., Cruz J. B.: A Stackelberg solution for games with many players. *IEEE Trans. Autom. Contr.* **AC-18**, 3 (1973).
18. Simaan M., Cruz J. B.: On the Stackelberg strategy in non-zero-sum games. *J. Opt. Theory a. Appl.* **11**, 5 (1973).
19. Simaan M., Cruz J. B.: Additional aspects of the Stackelberg strategy in non-zero-sum games. *Ibid.* **11**, 6 (1973).

20. STARR A. W.: Computation of Nash equilibria for nonlinear non-zero-sum games. Proc. I Int. Conf. on Theory a. Application of Differential Games, Amherst, Mass. 1969.
21. STARR A. W., HO Y. C.: Non-zero-sum differential games. *J. Opt. Theory a. Appl.* **3**, 3 (1969).
22. STARR A. W., HO Y. C.: Further properties of non-zero-sum differential games. *Ibid.* **3**, 4 (1969).
23. STOER J., BULIRSCH R.: Einführung in die numerische Mathematik II. Berlin 1973.
24. TABAC D.: Numerical solutions of differential game problems. *Int. J. Syst. Sci.* **6**, 6 (1975).
25. WEINBERG M. M.: Wariacjonnyj mietod i mietod nomotonnych opieratorow. Moskwa 1972.

Received, October 1976.

Metody numeryczne rozwiązywania wieloosobowych gier różniczkowych o sumie niezerowej

W pracy omówiono problemy związane z numerycznym wyznaczaniem punktów równowagi wieloosobowych gier różniczkowych o sumie niezerowej oraz przedstawiono pewien nowy algorytm obliczania punktów Nasha i Stackelberga. Zbieżność tego algorytmu udowodniono dla szerokiej klasy gier różniczkowych.

Вычислительные методы решения диффаренциальных игр *N*-лиц о ненулевой сумме

В статьи рассматривается проблемы связаны с вычислением точек равновесия фиффе-ренциальных игр *N*-лиц и ненулевой сумме. Представлен новый алгоритм вычисления точек Нэша и Стакельберга и доказаны теоремы о его сходимости для широкого класса игровых задач.