Operations Research Letters 10 (1991) 285–290
North-Holland
July 1991

Copositive-plus Lemke algorithm solves polymatrix games

Douglas A. Miller and Steven W. Zucker

Computer Vision and Robotics Laboratory, McGill Research Center for Intelligent Machines, McGill University, Montréal, Québec, Canada H3A 2A7

Received March 1990
Revised September 1990

A problem in visual labeling and artificial neural networks, equivalent to finding Nash equilibria for polymatrix n-person games, may be solved by the copositive-plus Lemke algorithm. Analysis suggests efficiency improves Howson's recursive method by $O(n^{-1})$ and is same order as Eaves' L_2-formulation. Method extends to any dynamical system $p' = Rp + c$, $Ap \le q$, $p \ge 0$.

linear complementarity problem; polymatrix games; relaxation labeling; artificial neural networks; dynamical systems

1. Introduction

In [13] Lemke and Howson showed how to find Nash equilibria [15] for bimatrix games by formulating the problem as an instance of what came to be known as the linear complementarity problem (LCP). Lemke later described this more general problem in [12], and gave a complementary pivoting procedure using a negative artificial column for solving certain special cases, in particular, where the defining matrix M was, as Cottle and Dantzig [1] later referred to it, copositive-plus.

However, it was not apparent in [1] or [12] how to fit bimatrix games into the copositive-plus category, and the solution of bimatrix games depended on a special starting pivoting sequence which did not involve an artificial column. Eaves [2] showed that a slight reformulation of the bimatrix games problem fit into his more general class of L-class LCP's, for which he showed Lemke's method with artificial column terminated successfully. However, Eaves' formulation of bimatrix games was not copositive-plus.

Shortly afterward Howson [7] described an algorithm, suggested by a more general algorithm of Wilson [16], for *polymatrix* n-person games, that is, n-person games where payoffs to player i in pure strategies are of the form

$$s_i(\lambda_1, \dots, \lambda_n) = \sum_j r_{ij}(\lambda_i, \lambda_j),$$

where $r_{ij}(\lambda_i, \lambda_j)$ is i's payoff from j given respective strategies λ_i and λ_j, and where $r_{ij}(\lambda_i, \lambda_j) = 0$ for $i = j$. This implies a payoff to i in mixed strategies of the form

$$\sum_{\lambda_i, j, \lambda_j} p_i(\lambda_i) r_{ij}(\lambda_i, \lambda_j) p_j(\lambda_j), \tag{1}$$

where $p_i(\lambda_i)$ is the probability player i uses strategy λ_i, $i = 1, \dots, m$. (For simplicity and without loss of generality we shall assume each player has exactly m pure strategies, though in general there can be any finite number.) Note for two players we have the usual bimatrix game.

Howson's algorithm was recursive, and amounted to solving $n - 1$ LCP subproblems of sizes $k(m + 1)$, for $k = 2, \dots, n$.

Subsequently, Eaves [4] formulated a more general form of the polymatrix game problem within his L_2-class of LCP's solvable by Lemke's algorithm. Eaves' method required solving just one LCP of size $(m + 2)n + 1$.

Similarly, the present paper shows how to reduce the polymatrix game equilibria problem to a copositive-plus LCP of size $(m+1)n$, solvable, like Eaves' formulation, with a single application of Lemke's algorithm.

If we assume an LCP of K rows requires $O(K)$ pivots [14, p. 162], and each pivot $O(K^2)$ operations, then the comparison in arithmetic operations of Howson's recursive method to the copositive-plus and L_2-class methods is $O(\Sigma_{k=2}^{n} m^3 k^3)$ or $O(m^3 n^4)$ versus $O(m^3 n^3)$.

Even for modest size n this difference would appear to be critical. For applications such as finding equilibria for dynamical systems of artificial neural networks in early vision processes [17], where n may be several orders of magnitude, Howson's recursive algorithm would appear quite impractical.

Comparing the present method to Eaves' [4], the most obvious difference would seem to be the sparsity of the LU decomposition of the basis columns. Since the LCP formulation we shall describe requires adding a constant to a submatrix of the original matrix, the present method would have a distinct disadvantage if this were to adversely affect the sparsity of the LU terms. However, it can be shown (Appendix) that it suffices to compute the LU terms of a matrix identical or very close to the original sparse matrix, in addition to solving a trivial vector equation. Thus, pivoting efficiency would not seem to be a significant difference between the two methods.

Finally we show copositive-plus LCP's also solve any linear dynamical system $p' = Rp$, $Ap \leq q$, $p \geq 0$ in the following sense: either an equilibrium of the system is computed, it is demonstrated the constraint set is infeasible, or an infinite ray representing an unbounded integral curves is computed. This implies a copositive-plus version of Eaves' L_2 computational results for generalized polymatrix games with joint constraints and convex quadratic penalties [4], and for general quadratic programming [3].

Clearly, one result of the present work is to extend the already considerable importance attached in [1] and [12] to copositive-plus LCP's.

Another result is to exhibit a strong connection between game theory and certain problems in artificial intelligence. Indeed our approach to polymatrix games was inspired by the equivalent *relaxation labeling* problem [8]. The latter was formulated in order to find easy continuous approximate solutions for NP-hard discrete problems in visual labeling such as the well-known blocks world problem [10]. This work was similar in many respects to Hopfield and Tank's later efforts to find approximate solutions to the traveling salesman problem using artificial neural networks [6]. More recently, artificial neural networks modeling early visual processes in the brain have been formulated as relaxation labeling problems [17].

The solution method described in [8] for relaxation labeling is of course applicable to polymatrix games. This method amounts to following an integral curve of a constrained linear dynamical system defining the polymatrix game. In the case of symmetric payoffs (e.g. [17]) it reduces to a gradient projection algorithm, the function maximized being the sum of all payoffs. However, for nonsymmetric payoffs such as zero sum games convergence may be poor or nonexistent. On the other hand this primal method describes the dynamical behavior of a very broad class of artificial neural systems. How it can be complemented by dual methods such as those treated here would seem a fruitful question for future research.

2. Relaxation labeling

In their paper [8], Hummel and Zucker pose a mathematical programming problem of importance to visual processing. We shall refer to this as the relaxation labeling problem and denote it (P). It may be stated:

(P) Given a set of n objects, each object i having m possible labels indexed by λ_i, and each object/label pair i, λ_i, j, λ_j having a mutual support $r_{ij}(\lambda_i, \lambda_j) \in \mathbb{R}$, find a set of weighted labelings $\{p_i(\lambda_i)\}_{i,\lambda_i}$ such that:

$$\sum_{\lambda_i} p_i(\lambda_i) = 1 \quad \text{for all } i, \qquad p_i(\lambda_i) \geq 0 \quad \text{for all } i, \lambda_i \tag{2}$$

and such that

$$\sum_{i,\lambda_i,j,\lambda_j} r_{ij}(\lambda_i, \lambda_j) p_j(\lambda_j) d_i(\lambda_i) > 0$$

has no solution for $d_i(\lambda_i)$, $1 \le i \le n$, $1 \le \lambda_i \le m$, where

$$\sum_{\lambda_i} d_i(\lambda_i) = 0 \text{ for all } i,$$

$$\text{if } p_i(\lambda_i) = 0, \text{ then } d_i(\lambda_i) \ge 0 \quad \text{for all } i, \lambda_i.$$

The motivation for this problem is more apparent in looking at an equivalent problem ($\hat{\mathrm{P}}$) [8, Theorem, 4.1, p. 273]:

($\hat{\mathbf{P}}$) Find a set of labels such that the weighted sum of supports of all labels at object i by all other labels, that is, the sum

$$\sum_{\lambda_i} v_i(\lambda_i) \sum_{j,\lambda_j} r_{ij}(\lambda_i, \lambda_j) p_j(\lambda_j) \tag{3}$$

is maximized over all possible labels $v_i(\lambda_i)$ by setting $v_i(\lambda_i) = p_i(\lambda_i)$.

Comparing (3) to (1), and changing the terms 'label' to 'strategy', and 'object' to 'player', ($\hat{\mathrm{P}}$) becomes precisely te problem of finding a mixed strategy equilibrium for a polymatrix n-person game.

3. The copositive-plus LCP

The LCP problem is that of finding real m-vectors z, w such that:

$$-Mz + w = q, \quad w \ge 0,\ z \ge 0,\ z^{\mathrm{T}} w = 0, \tag{4}$$

where M is a given $m \times m$ matrix, and q is a given real m-vector.

Lemke [12] described a complementary pivoting procedure for solving any LCP using a negative artificial column to start. However, in general, cases where the pivoting process terminated in a nonpositive column constituted a failure. Only for certain specific classes of matrices M the procedure was shown to be failproof. (This was not so surprising in retrospect, since it was later shown that the general LCP problem is NP-complete.) The focus of the LCP problem therefore became in [1] and [12] to identify matrices M for which Lemke's algorithm terminated successfully.

An important class of such matrices M (it included convex quadratic programs) was identified by Lemke in [12], and referred to in [1] as *copositive-plus*. Such matrices are characterized by:

$$u^{\mathrm{T}} M u \ge 0 \quad \text{for all } u \ge 0, \tag{5a}$$

$$\text{if } u^{\mathrm{T}} M u = 0 \text{ and } u \ge 0, \text{ then } (M + M^{\mathrm{T}}) u = 0. \tag{5b}$$

Lemke [12] showed that for M copositive-plus, termination of his algorithm in a ray (nonpositive column) implies no solution to the LCP exists. Otherwise the algorithm must terminate in a solution.

4. Polymatrix games as copositive-plus LCP's

We now transform (P) into a dual problem (D) which is in copositive-plus LCP form. We use Farkas' lemma in a manner similar to Kuhn and Tucker (cf. [11], p. 486]), except that we replace the gradient of an objective function with the vector field of a linear dynamical system.

Observe, in view of (2), by the addition of a constant k to each term $r_{ij}(\lambda_i, \lambda_j)$ in (3) we change the value of each sum (3) by exactly nk, regardless of the values of p and v. It follows that by choice of sufficiently small k we may without loss of generality assume

$$r_{ij}(\lambda_i, \lambda_j) < 0 \quad \text{for all } i, j, \lambda_i, \lambda_j.$$

Note all $r_{ii}(\lambda_i, \lambda_i')$ terms are negative too, all having the same value k. In effect each $m \times m$ matrix $[r_{ii}(\lambda_i, \lambda_i')]$ contributes a constant k to i's payoff.

Note also that in view of the negativity of the $r_{ij}(\lambda_i, \lambda_j)$ and (3), the equalities in (2) may be relaxed in the positive direction without affecting the equilibria. Thus (2) may be replaced by

$$\begin{aligned} -\sum_{\lambda_i} p_i(\lambda_i) &\le -1 \quad \text{for all } i, \\ p_i(\lambda_i) &\ge 0 \quad \text{for all } i, \lambda_i. \end{aligned} \tag{6}$$

Let p^{T} be the real mn-vector $(p_1(1), \ldots, p_1(m), \ldots, p_n(1), \ldots, p_n(m))$, let q^{T} be the n-vector $(-1, \ldots, -1)$, and let A be the $n \times mn$ matrix

$$\begin{bmatrix} -1 & \cdots & -1 & \cdots & 0 & \cdots & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & \cdots & 0 & \cdots & -1 & \cdots & -1 \end{bmatrix}.$$

Then the system (6) may be expressed as

$$Ap \leq q, \quad p \geq 0. \tag{7}$$

If also we let R be the $mn \times mn$ matrix

$$\begin{bmatrix} [r_{11}(\lambda_1, \lambda_1')] & \cdots & [r_{1n}(\lambda_1, \lambda_n)] \\ \vdots & \ddots & \vdots \\ [r_{n1}(\lambda_n, \lambda_1)] & \cdots & [r_{nn}(\lambda_n, \lambda_n')] \end{bmatrix},$$

then $\bar{p}$ solves (P) if and only if $\bar{p}$ satisfies (7) and there is no $d \in \mathbb{R}^{mn}$ where

$$\begin{aligned} &-d^{\mathrm{T}} R\bar{p} < 0, \\ &\text{if } A_{i\cdot}\, p = q_i, \text{ then } A_{i\cdot}\, d \leq 0, \\ &\text{if } \bar{p}_i(\lambda_i) = 0, \text{ then } -d_i(\lambda_i) \leq 0. \end{aligned} \tag{8}$$

We use Farkas' lemma [5]. This states that given a real matrix B and corresponding vector b, exactly one of the following systems has a solution:

$$-b^{\mathrm{T}}\mu < 0, \quad B\mu \leq 0,$$

or

$$B^{\mathrm{T}}\nu = b, \quad \nu \geq 0.$$

It follows from Farkas' lemma and (8) that $\bar{p}$ is a solution to (P) if and only if $\bar{p}$ satisfies (7) and the system

$$\begin{aligned} &\begin{bmatrix} A^{\mathrm{T}} & -I_{mn} \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix} = Rp, \quad y, u \geq 0, \\ &\text{if } p_i(\lambda_i) > 0, \text{ then } u_i(\lambda_i) = 0 \\ &\quad \text{for all } i, \lambda_i, \\ &\text{if } A_{i\cdot}\, p < q_i, \text{ then } y_i = 0 \quad \text{for all } i, \end{aligned} \tag{9}$$

has a solution for y, u. We shall refer to the problem defined by (7) and (9) as (D). Letting $v = q - Ap$, we may rewrite (9) in the LCP form (4):

(D) Find p, y, u, v such that:

$$\begin{bmatrix} R & -A^{\mathrm{T}} \\ A & 0 \end{bmatrix} \begin{bmatrix} I_{mn} & 0 \\ 0 & I_n \end{bmatrix} \begin{bmatrix} p \\ y \\ u \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ q \end{bmatrix}, \tag{10}$$

$$p, y, u, v \geq 0, \quad p^{\mathrm{T}}u + y^{\mathrm{T}}v = 0.$$

It is trivial since $-R > 0$ that $-R$ satisfies the copositive-plus condition (5), and also trivial that a zero matrix is copositive-plus, hence as Cottle and Dantzig [1] note, this implies the matrix

$$M = \begin{bmatrix} -R & A^{\mathrm{T}} \\ -A & 0 \end{bmatrix}$$

is copositive-plus for any values of A. Since it is known [15] from the Brouwer fixed point theorem that (P) and hence (D) have a solution, we have proved

Proposition. *Lemke's complementary pivoting method with artificial column applied to the copositive-plus matrix M terminates in a solution to* (P).

Notice that (10) is in the same form as Howson's formulation [7, p. 314], except that R is negative.

5. Extensions to dynamical systems

In this section we show the copositive-plus Lemke algorithm will either: find an equilibrium, find a ray which is an unbounded integral curve, or demonstrate infeasibility, for any constrained dynamical system (11) below. This shows that Eaves' results on L_2-class computability for extensions of polymatrix games [4], and general quadratic programs [3] may be replicated in terms of copositive-plus computability. In so doing we shall describe a strong geometric link between the two methods.

Consider the constrained dynamical system

$$\begin{aligned} &p' = Rp + c, \\ &Ap \leq q, \quad p \geq 0. \end{aligned} \tag{11}$$

We have shown how to compute an equilibrium for this system for special forms of R, A, q, c. (Here $-c$ replaces 0 in the upper right side of (10).) But to be able to render R negative by adding a negative constant without altering the equilibria of (11), it is only necessary that

$$Ae = \rho, \tag{12}$$

where e is a vector of 1's and ρ is fixed. So assuming (12) is at least implicitly in (11) we may add any other constraints we wish, and choose any values for R and c. For instance if for each i, $[r_{ii}(\lambda_i, \lambda_i')]$ is a negative semi-definite matrix, and the constraint set in (11) satisfies (2), an equilibrium of (11) will correspond to that of a more general polymatrix game.

Alternatively, if R is symmetric, then an equilibrium of (11) will be a Kuhn–Tucker point for a general quadratic program.

If the constraint set in (11) has no nonzero homogeneous solution (no infinite rays), then by adding slack variables, making the appropriate positive diagonal transformation D on the resulting affine space, and writing equalities as two inequalities, we may always transform (11) into a problem in which (12) is satisfied.

Suppose now we make no assumption on (11). To eliminate any infinite rays we can add an artificial constraint $e^{T}p \le \delta$, where δ is big. As with the lexicographic rule, we pivot as though δ did have a value, except in this case δ is arbitrarily large relative to any other values.

Putting this new problem in the form (10), observe (cf. [14, p. 170]) that it is equivalent to Eaves' pivoting rule for his L_2 formulation [2]. That is, if we stopped here and applied Lemke's algorithm, we would be using Eaves' method. Thus, the δ-constraint enters strongly into both methods.

We can now transform the δ-version of (11) into copositive-plus form. Since this problem is bounded, by fixed point arguments [9] it is either infeasible or has an equilibrium z, and this will be determined by Lemke's algorithm. It if is infeasible, to is (11). Otherwise, if the dual multiplier for nonnegativity constraint of the slack variable of the δ-constraint is zero, then $D^{-1}z$ (minus the slack variables) is an equilibrium of (11). Otherwise there is an integral curve of (11) which follows an infinite ray for arbitrarily large values of δ. If R is symmetric, then (11) defines an unbounded quadratic program.

Appendix: Maintaining LU sparsity

Given an LU decomposition of a nonsingular matrix, we seek an efficient solution for

$$[LU+K]x=b, \tag{13}$$

where

$$K=[\kappa \quad \cdots \quad \kappa \quad 0 \quad \cdots \quad 0],$$

κ is a nonzero column vector, and $LU+K$ is nonsingular.

Notice Kx is in the space spanned by κ, so we may rewrite (13) as

$$LUx=\rho\kappa=b, \tag{14}$$

where ρ is a scalar. Solving (14) for x and substituting in (13) we obtain

$$KU^{-1}L^{-1}b=\rho(\kappa+KU^{-1}L^{-1}\kappa). \tag{15}$$

Since K is nonzero, the left side of (15) may be nonzero for some b, hence $\kappa+KU^{-1}L^{-1}\kappa$ must be nonzero and we can trivially solve (15) for ρ.

It remains how to insure nonsingularity. This is no problem to start, since at that point $K=0$. Furthermore, if at some iteration we have a nonsingular basis A_s such that A_s+K_s is also nonsingular, and if A_r+K_r is the next basis chosen, then the standard pivoting rules insure A_r+K_r is nonsingular. The problem, then, is how to proceed if A_r is singular, that is, if the element of A_r corresponding to the pivot element of A_r+K_r is zero.

One way is to perturb a single $r_{ij}(\lambda_i, \lambda_j)$ in the column ϕ of A_r being pivoting into in order that the pivot row of $A_s^{-1}\phi$ becomes nonzero. That this is always possible follows from the fact that, by the nonsingularity of A_s and A_r+K_r, $A_s^{-1}(\phi+\kappa)$ must be nonzero in the pivot row.

Of course eventually we might build up a large number of these perturbations, but by periodically removing those no longer needed, this number can never exceed the number of rows in the problem.

Addendum proof: Adding an implicit constraint

We wish to show that any bounded equilibrium problem (11) is equivalent to a problem in the same form which includes the constraint (12).

First add slack variables to put the problem in the form

$$\hat{p}'=\hat{R}\hat{p}+\hat{c},$$
$$\hat{A}\hat{p}=q, \quad \hat{p}\ge 0.$$

We seek a diagonal matrix D such that De lies orthogonal to the constraint set $\hat{A}\hat{p}=0$. Consider the inequalities

$$\hat{A}^{T}y\ge e. \tag{16}$$

If (16) has no solution for y, then by duality theory the constraint set in (11) has a nonzero homogeneous solution, which we are assuming is not the case. If $\bar{y}$ solves (16), define D by $De=\hat{A}^{T}\bar{y}$, and consider the dynamical system

$$\begin{aligned} z' &= \hat{R}D^{-1}z+\hat{c}, \\ \hat{A}D^{-1}z &= q, \quad z\ge 0. \end{aligned} \tag{17}$$

Clearly $\bar{z}$ is an equilibrium of (17) if and only if $D^{-1}\bar{z}$ is an equilibrium of (11). Furthermore by construction of D there exists a ρ such that

if $\hat{A}D^{-1}z=q$, then $\rho=e^{T}DD^{-1}z=e^{T}z$.

Finally we may put (17) in the form (11) by writing each equality constraint as two inequalities. This completes the transformation.

Acknowledgements

The authors thank Ilan Adler for valuable discussion of an early draft of this paper. This work was supported by grants from NSERC and AFOSR.

References

[1] R.W. Cottle and G.B. Dantzig, "Complementary pivot theory of mathematical programming", in: G.B. Dantzig and A.F. Veinott Jr. (eds.), *Mathematics of the Decision Sciences* Part I, American Mathematical Society, Providence, RI, 1968, 115–136.

[2] B.C. Eaves, "The linear complementarity problem", *Management Sci.* **17**, 612–634 (1971).

[3] B.C. Eaves, "On quadratic programming", *Management Sci.* **17**, 698–711 (1971).

[4] B.C. Eaves, "Polymatrix games with joint constraints", *SIAM J. Appl. Math.* **24**, 418–423 (1973).

[5] J. Farkas, "Theorie der einfachen Ungleichungen", *J. Reine Angew. Math.* **124**, 1–27 (1902).

[6] J.J. Hopfield and D.W. Tank, "'Neural' computation of decisions in optimization problems", *Biol. Cybern.* **52**, 141–152 (1985).

[7] J.T. Howson Jr., "Equilibria of polymatrix games", *Management Sci.* **18**, 312–318 (1972).

[8] R.A. Hummel and S.W. Zucker, "On the foundations of relaxation labeling processes", *IEEE Pattern Anal. Machine Intell.* **5**, 267–287 (1983).

[9] D. Kinderlehrer and G. Stampacchia, *An Introduction to Variational Inequalities and Their Applications*, Academic Press, New York, 1980.

[10] L.M. Kirousis and C.H. Papadimitriou, "The complexity of recognizing polyhedral scenes", *J. Comput. System Sci.* **37**, 14–38 (1988).

[11] H.W. Kuhn and A.W. Tucker, "Nonlinear programming", in: J. Neyman (ed.), *Second Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, 1951, 481–492.

[12] C.E. Lemke, "Bimatrix equilibrium points and mathematical programming", *Management Sci.* **11**, 681–689 (1965).

[13] C.E. Lemke and J.T. Howson Jr., "Equilibrium points of bimatrix games", *SIAM J. Appl. Math.* **12**, 413–423 (1964).

[14] K.G. Murty, *Linear Complementarity, Linear and Nonlinear Programming*, Heldermann Verlag, Berlin, 1988.

[15] J.F. Nash, "Noncooperative games", *Ann. Math.* **54**, 286–295 (1951).

[16] R. Wilson, "Computing equilibria of *N*-person games", *SIAM J. Appl. Math.* **21**, 80–87 (1971).

[17] S.W. Zucker, A. Dobbins and L. Iverson, "Two stages of curve detection suggest two styles of visual computation", *Neural Comput.* **1**, 60–81 (1989).