Improved Equilibrium Enumeration for Bimatrix Games

Extended abstract

June 30, 1998

Bernhard von Stengel

Institute for Theoretical Computer Science
ETH Zurich, CH-8092 Zurich
stengel@inf.ethz.ch

The enumeration of all equilibria of a bimatrix game is a classical algorithmic problem in game theory. As shown by Vorob'ev (1958), Kuhn (1961), and Mangasarian (1964), all equilibria can be represented as convex combinations of the vertices of certain polyhedra defined by the payoff matrices. Simplified by a projective transformation that eliminates the payoff variable, these polyhedra have the form

$$P_1 = \{\, x \in \mathbb{R}^M \mid x \geq \mathbf{0}_M,\ B^\top x \leq \mathbf{1}_N \},$$
$$P_2 = \{\, y \in \mathbb{R}^N \mid Ay \leq \mathbf{1}_M,\ y \geq \mathbf{0}_N \}\,.$$

The players' payoff matrices are A and B. These are w.l.o.g. positive, so that P_1 and P_2 are bounded and therefore polytopes. The disjoint sets M and N contain the row and column indices, respectively, which represent the pure strategies of the two players. The vectors $\mathbf{0}$ and $\mathbf{1}$ have all components equal to zero and one, respectively. The elements x in $P_1 - \{\mathbf{0}_M\}$ and y in $P_2 - \{\mathbf{0}_N\}$ represent the mixed strategies of the two players after normalization so that their components sum up to one. After this normalization, the right hand sides in $B^\top x \leq \mathbf{1}_N$ and $Ay \leq \mathbf{1}_M$ denote the expected payoff to the other player. If any of these inequalities is *binding* (holds as equality), the pure strategy corresponding to that inequality is optimal. In *equilibrium*, any pure strategy must be optimal or have probability zero. Hence, any Nash equilibrium of (A, B) corresponds to a *complementary* pair (x, y) in $P_1 \times P_2$ where any inequality in $M \cup N$ in the above definition of P_1 and P_2 is binding for x or for y.

It suffices to look at the *vertices* of the polytopes. Any equilibrium is the convex combination of a set of vertex pairs that are all complementary each other (Winkels, 1979; Jansen, 1981). We formulate this as follows.

Theorem. *Let V_1 and V_2 be the sets of vertices of P_1 and P_2, respectively, and let R be the set of completely labeled vertex pairs in $V_1 \times V_2 - \{(\mathbf{0}_M, \mathbf{0}_N)\}$. Then (x, y) represents a Nash equilibrium of (A, B) if and only if it belongs to the convex hull of some subset of R of the form $U_1 \times U_2$ where $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$.*

Here, the set R can be viewed as a bipartite graph with the completely labeled vertex pairs as edges. The subsets $U_1 \times U_2$ are *cliques* of this graph. The convex hulls of the maximal cliques of R are the *convex components* of Nash equilibria. (The *topological* equilibrium components are unions of non-disjoint convex components.) If the game is nondegenerate, all convex components are singletons.

Given a (possibly degenerate) bimatrix game (A, B), we want to output its convex equilibrium components. The straightforward approach is to enumerate the vertices of P_1 and P_2, find those that are complementary, which define the bipartite graph R in the Theorem, and then compute the maximal cliques of R.

Vertex enumeration is indeed superior to support enumeration, by an exponential factor. In a (square) $n \times n$ bimatrix game, for example, there are about 4^n possible equilibrium supports, even if the game is nondegenerate, but less than $O(2.6^n)$ vertices due to the Upper Bound Theorem for polytopes. Except for degenerate vertices, the algorithm by Avis and Fukuda (1992) enumerates the vertices of a polytope efficiently, that is, in running time that is polynomial in *input and output size*. Deciding if a game has a unique equilibrium is NP-hard (Gilboa and Zemel, 1989). Hence, an efficient *equilibrium enumeration* algorithm is not likely to exist since it would decide that question in polynomial time. Once the complementary vertices are found, however, the cliques of R can again be enumerated efficiently (Tsukiyama et al., 1976). Because of its simplicity, we have implemented a variant for bipartite graphs of the clique enumeration algorithm by Bron and Kerbosch (1973), which is very fast in practice.

Our main contribution is an algorithm that enumerates only the vertices of one polytope (the one of lower dimension, say P_1). It finds *directly* the complementary vertices of the other polytope, if these exist. For a given vertex v_1 of P_1, the complementary binding inequalities of P_2 define a face F of P_2. In many cases F is the empty face, if v_1 is not part of an equilibrium. For an equilibrium in a nondegenerate game, F is a single vertex of P_2. In degenerate games (where v_1 itself may or may not be a degenerate vertex), F may also be a higher-dimensional face of P_2.

We use the reverse-search pivoting algorithm by Avis and Fukuda for enumerating the vertices of P_1 (assuming $|M| \leq |N|$), with $\mathbf{0}_M$ as the nondegenerate starting vertex. For each pivoting step that generates a new vertex v_1 of P_1, we perform the complementary pivoting step on the affine *hyperplane* arrangement that has P_2 as a cell, and then check for feasibility. For generic payoffs, this pivoting step is possible, and immediately finds a vertex of the face F of P_2 that is complementary to v_1, or establishes that F is empty. If the pivoting step is not possible (since the pivot element is zero), we solve that problem in the standard way by a phase I simplex method. Once a vertex of F is found, the remaining vertices of F, now vertices of P_2, are again determined by the Avis–Fukuda algorithm. The output is then an incidence list of the bipartite graph R that is further processed for enumerating its maximal cliques.

Our method is particularly advantageous if the game has much fewer rows than columns (or vice versa) since the number of vertices of a polytope is in general exponential in the dimension, and polynomial in the number of defining inequalities. We use fraction-free exact arithmetic in order deal correctly with complementarity, which is a combinatorial property. Results about computational performance will be reported.

References

D. Avis and K. Fukuda (1992), A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. Discrete and Computational Geometry 8, 295–313.

C. Bron and J. Kerbosch (1973), Finding all cliques of an undirectred graph. Comm. ACM 16, 575–577.

I. Gilboa and E. Zemel (1989), Nash and correlated equilibria: some complexity considerations. Games and Economic Behavior 1, 80–93.

M. J. M. Jansen (1981), Maximal Nash subsets for bimatrix games. Naval Research Logistics Quarterly 28, 147–152.

H. W. Kuhn (1961), An algorithm for equilibrium points in bimatrix games. Proc. National Academy of Sciences of the U.S.A. 47, 1657–1662.

O. L. Mangasarian (1964), Equilibrium points in bimatrix games. Journal of the Society for Industrial and Applied Mathematics 12, 778–780.

S. Tsukiyama, H. Ozaki, H. Ariyoshi, and M. Ide (1976), An algorithm for generating all the maximal independent sets. Electronics and Communications in Japan 59-A:6, 1–8.

N. N. Vorob'ev (1958), Equilibrium points in bimatrix games. Theory of Probability and its Applications 3, 297–309.

H.-M. Winkels (1979), An algorithm to determine all equilibrium points of a bimatrix game. In: Game Theory and Related Topics, eds. O. Moeschlin and D. Pallaschke, North-Holland, Amsterdam, 137–148.