On Iterated Dominance, Matrix Elimination, and Matched Paths

Felix Brandt[*] Felix Fischer[†] Markus Holzer[‡]

Abstract

We study computational problems that arise in the context of iterated dominance in anonymous games, and show that deciding whether a game can be solved by means of iterated weak dominance is NP-hard for anonymous games with three actions. For the case of two actions, this problem can be reformulated as a natural elimination problem on a matrix. While enigmatic by itself, the latter turns out to be a special case of matching along paths in a directed graph, which we show to be computationally hard in general but also use to identify tractable cases of matrix elimination. We further identify different classes of anonymous games where iterated dominance is in P and NP-complete, respectively.
Keywords: Algorithmic Game Theory, Computational Complexity, Iterated Dominance, Matching

1 Introduction

We study problems related to iterated dominance in anonymous and symmetric games. An *anonymous game* is characterized by the fact that players do not distinguish between other players in the game, *i.e.*, their payoff only depends on the numbers of other players playing the different actions, but not on their identities. A *symmetric* game additionally has identical payoff functions for all players. A *strategy* of a player is a probability distribution over his actions. An action of a particular player in a game is said to be *weakly dominated* if there exists a strategy guaranteeing him at least the same payoff for any profile of actions of the other players, and strictly more payoff for some such action profile. The well-known solution concept of *iterated dominance* works by removing a dominated action and applying the same reasoning to the reduced game. A game is then called *solvable* by iterated dominance if there is a sequence of eliminations that leaves only one action for each player (Moulin, 1979). In general, the order in which actions are eliminated is important.

Related Work Deciding whether a game can be solved by iterated weak dominance is NP-complete already for games with two players and two different payoffs (Gilboa et al., 1993; Conitzer and Sandholm, 2005). Apart from work by Brandt et al. (2006), we are not aware of any complexity results for restricted classes of games. The complexity of Nash equilibria in anonymous and symmetric games, on the other hand, has been studied extensively by several authors. Symmetric games are guaranteed to possess a symmetric equilibrium, *i.e.*, one where all players play the same strategy. Such an equilibrium can be found efficiently if the number of actions is not too large compared to the number of players (Papadimitriou and Roughgarden, 2005). Anonymous games have recently been shown to admit the efficient computation of approximate Nash equilibria—by a factor depending on the Lipschitz constant of the payoff function and on the square of the number of actions—, and a PTAS for the case of two actions (Daskalakis and Papadimitriou, 2007). Brandt et al. (2007) show that the pure equilibrium problem in anonymous games is tractable if the number

[*]Institut für Informatik, Universität München, 80538 München, Germany, email: `brandtf@tcs.ifi.lmu.de`
[†]Institut für Informatik, Universität München, 80538 München, Germany, email: `fischerf@tcs.ifi.lmu.de`
[‡]Institut für Informatik, Technische Universität München, 85748 Garching, Germany, email: `holzer@in.tum.de`

of actions is a constant, and complete for NP or PLS, respectively, if the number of actions grows in the number of players.

Results and Paper Structure We begin by introducing the relevant game-theoretic concepts. In Section 3 we then show that iterated dominance solvability is NP-hard for symmetric games with a growing number of actions, and tractable for symmetric games with a constant number of actions. The only remaining case, anonymous games with a constant number of actions, is then studied for the remainder of the paper. When restricted to two actions, it can be reformulated as a natural elimination problem on matrices, which we do in Section 4. While the complexity of this problem remains open, we point out connections to a matching problem on paths of a directed graph in Section 5. The latter problem, which may be of independent interest, is intractable in general but allows us to obtain efficient algorithms for restricted versions of matrix elimination. In Section 6 we finally use the matching formulation to show NP-hardness of iterated dominance in anonymous games with three actions.

2 Preliminaries

An accepted way to model situations of strategic interaction is by means of a *normal-form game* (see, *e.g.*, Luce and Raiffa, 1957).

Definition 1 (normal-form game) *A* game in normal-form *is a tuple* $\Gamma = (N, (A_i)_{i\in N}, (p_i)_{i\in N})$ *where N is a set of* players *and for each player $i \in N$, A_i is a nonempty set of* actions *available to player i, and $p_i : (\bigtimes_{i\in N} A_i) \to \mathbb{R}$ is a function mapping each action profile of the game (*i.e.*, combination of actions) to a real-valued* payoff *for player i.*

A combination of actions $s \in \bigtimes_{i\in N} A_i$ is also called a profile of *pure strategies*. This concept can be generalized to *mixed strategy profiles* $s \in S = \bigtimes_{i\in N} S_i$, by letting players randomize over their actions. We have S_i denote the set of probability distributions over player i's actions, or *mixed strategies* available to player i. We further write $n = |N|$ for the number of players in a game, s_i for the ith element of profile s, and s_{-i} for the vector of all elements but s_i.

A central aspect of symmetries in games is the inability to distinguish between other players. Following Daskalakis and Papadimitriou (2007), the most general class of games with this property will be called *anonymous*. Four different classes of games are then obtained by considering two additional characteristics: *identical payoff functions* for all players and the ability to *distinguish oneself* from the other players. The games obtained by adding the former property will be called *symmetric*, and presence of the latter will be indicated by the prefix "*self*".[1] For the obvious reason we only talk about games where the set of actions is the same for all players and write $A = A_1 = \cdots = A_n$ and $k = |A|$, respectively, to denote this set and its cardinality. We arrive at the following definition.

Definition 2 (symmetries) *Let $\Gamma = (N, (A_i)_{i\in N}, (p_i)_{i\in N})$ be a normal-form game, A a set of actions such that $A_i = A$ for all $i \in N$. For any permutation $\pi : N \to N$ of the set of players, let $\pi' : A^N \to A^N$ be the permutation of the set of action profiles given by $\pi'((a_1, a_2, \ldots, a_n)) = (a_{\pi(1)}, \ldots, a_{\pi(n)})$. Γ is called*

- anonymous *if $p_i(s) = p_i(\pi'(s))$ for all $s \in A^N$, $i \in N$ and all π with $\pi(i) = i$,*
- symmetric *if $p_i(s) = p_j(\pi'(s))$ for all $s \in A^N$, $i, j \in N$ and all π with $\pi(j) = i$,*
- self-anonymous *if $p_i(s) = p_i(\pi'(s))$ for all $s \in A^N$, $i \in N$, and*

[1]This terminology differs from the one used in our previous work (Brandt et al., 2007).

- self-symmetric *if* $p_i(s) = p_j(\pi'(s))$ *for all* $s \in A^N$, $i, j \in N$.

The class of self-symmetric games equals the intersection of symmetric and self-anonymous games, and both of these are strictly contained in the class of anonymous games. In the above definition, π' is an automorphism on the set of action profiles that preserves the number of players who play a particular action. Thus, an intuitive and convenient way to describe an anonymous game is in terms of the equivalence classes induced by π', corresponding to the different values of $\#(s) = (\#(a, s))_{a \in A}$ for $s \in A^N$. This description requires only space polynomial in the number of players when the number of actions is bounded.

A well-known method for simplifying strategic games is the removal of actions that are weakly dominated by some strategy of the same player, *i.e.*, playing the former is never better than playing the latter, while in some situation it is strictly worse. The removal of one or more dominated actions from the game may render additional actions dominated, which may then iteratively be removed.

Definition 3 (iterated dominance) *Let* $\Gamma = (N, (A_i)_{i \in N}, (p_i)_{i \in N})$. *An action* $d_i \in A_i$ *is said to be* (weakly) dominated *by strategy* $s_i \in S_i$ *if for all* $b \in \bigtimes_{i \in N} A_i$, $p_i(b_{-i}, d_i) \leq \sum_{a_i \in A_i} s_i(a_i) p_i(b_{-i}, a_i)$ *and for at least one* $\hat{b} \in \bigtimes_{i \in N} A_i$, $p_i(\hat{b}_{-i}, d_i) < \sum_{a_i \in A_i} s_i(a_i) p_i(\hat{b}_{-i}, a_i)$.

A sequence $a_1, a_2, \dots, a_k$ *of actions* $a_j \in \cup_{i \in N} A_i$ *is called iterated (weak) dominance if for all* $j \leq k$, a_j *is weakly dominated in the game* $(N, (A_i^j)_{i \in N}, (p_i)_{i \in N})$ *where* $A_i^j = A_i \setminus \{ a_\ell : \ell < j \}$.

In contrast to iterated *strict* dominance, the result of this process depends on the order in which actions are removed, since the elimination of an action may render actions of another player undominated (see, *e.g.*, Apt, 2004). If at the end of the process only a single action remains for each player, we say that the game has been *solved*. In the following, we call iterated dominance solvability (IDS) the computational problem of deciding, for a given game Γ, whether there exists a sequence of eliminations of length $\sum_{i \in N}(|A_i| - 1)$. Iterated dominance eliminability (IDE) is given an action $a \in A_i$ of some player $i \in N$ and asks whether it is possible to eliminate a.

3 Complexity of Iterated Dominance

Intuitively, a large number of actions nullifies the computational advantage obtained from symmetries by allowing for a distinction of the players by means of the *actions they play*. Brandt et al. (2007) show that this renders the search for pure Nash equilibria NP-hard. We can derive an analogous result for iterated dominance in self-symmetric games, hardness for the other classes follows by inclusion. For games with a polynomial number of players and efficiently computable payoff functions, the problems are NP-complete. Detailed proofs of all results are deferred to the appendix.

Theorem 1 *IDS and IDE are NP-hard for all four classes of anonymous games, even if the number of actions grows logarithmically in the number of players, if only dominance by pure strategies is considered, and if there are only two different payoffs.*

In the case of symmetric games, iterated dominance becomes tractable when the number of actions is bounded by a constant.

Theorem 2 *For symmetric games with a constant number of actions, IDS and IDE can be decided in polynomial time.*

In light of these two results, only one interesting class remains, namely anonymous games with a constant number of actions. To gain a better understanding of the problem, we restrict ourselves even further to games with only two actions. It turns out that iterated dominance can reformulated as a natural elimination problem on matrices. The latter problem will be the topic of the following section.

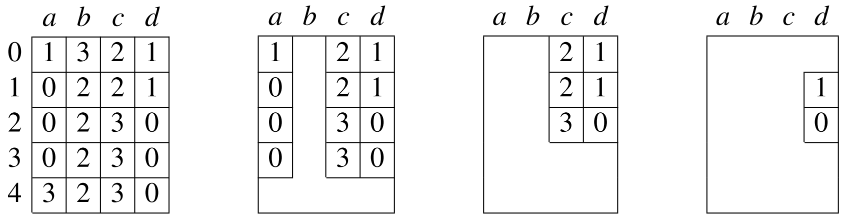

Figure 1: A matrix and a sequence of eliminations

4 A Matrix Elimination Problem

Consider an $m \times n$ matrix X with entries from the natural numbers. We call a column c of X *increasing* for an interval I over the rows of X if the entries in c are monotonically increasing in I, with a strict increase somewhere in this interval. Analogously, we call c *decreasing* for I if its entries are monotonically decreasing in I, with a strict decrease somewhere in this interval. We then say that c is *active* for I if it is either increasing or decreasing for this interval. Now consider a process that starts with X and successively eliminates pairs of a row and a column. Rows will only be eliminated from the top or bottom, such that the remaining rows always form an interval over the rows of X. A column will only be eliminated if it is active for the remaining rows. Elimination of an increasing column is accompanied by elimination of the top row. Similarly, a decreasing column and the bottom row are eliminated at the same time. The process ends when no active columns remain. In this paper we study two computational problems. *Matrix elimination* asks whether for a given matrix there exists a sequence of such eliminations of length $\min(m-1, n)$, *i.e.*, one that eliminates all columns of the matrix or all rows but one, depending on the dimensions of the original matrix. *Eliminability of a column* asks whether a particular column can be eliminated at some point during the elimination process.

More formally, the matrix elimination process can be described by a pair of sequences of equal length, where the first sequence consists of column indices of X and the second sequence of elements of $\{0, 1\}$, corresponding to elimination of the top or bottom row, respectively. The first sequence will contain every column index at most once. The ith element of the second sequence will be 0 or 1, respectively, if the column corresponding to the ith element of the first sequence is increasing or decreasing in the interval described by the number of 0s and 1s in the second sequence up to element $i-1$.

Consider for example the sequence of matrices shown in Figure 1, obtained by starting with the 5×4 matrix on the left and successively eliminating columns b, a, c, and d. In this particular example, the process ends when all rows and columns of the matrix have been eliminated. Of course, this does not always have to be the case. Again consider the matrix on the left of Figure 1, with all entries in the second row from the bottom replaced by 2. It is easy to see that in this case no column will be active after the first elimination step, and elimination cannot continue. Since column b was the only active column in the first place, eliminating just this one column is in fact all that can be done. A related phenomenon can be observed if we instead replace the top entry in the leftmost column by 0, and take a closer look at the matrix obtained after one elimination. While we could continue eliminating at this point, it is already obvious that we will not obtain a sequence of length 4. The reason is that one of the columns not eliminated so far, namely the leftmost one, contains the same value in every row. This column cannot become active anymore, and, as a consequence, will never be eliminated.

Let us define the problem more formally. For a set A, $v \in A^n$, and $a \in A$, denote by $\#(a, v) = |\{\, \ell \leq n : v_\ell = a \,\}|$ the *commutative image* of a and v, and write $v_{\ldots k} = (c_1, c_2, \ldots, c_k)$ for the prefix of v of length $k \leq n$. Further denote $[n] = \{1, 2, \ldots, n\}$ and $[n]_0 = \{0, 1, \ldots, n\}$.

Definition 4 (elimination sequence) *Let $X \in \mathbb{N}^{m \times n}$ be a matrix. A column $k \in [n]$ of X is called* increasing *in an interval* $[i, j] \subseteq [m]$ *if the sequence* $x_{ik}, x_{i+1,k}, \dots, x_{jk}$ *is monotonically increasing and* $x_{ik} < x_{jk}$*, and* decreasing *in* $[i, j] \subseteq [m]$ *if* $x_{ik}, x_{i+1,k}, \dots, x_{jk}$ *is monotonically decreasing and* $x_{ik} > x_{jk}$.

Then, an elimination sequence *of length k for X is a pair* (c, r) *such that* $c \in [m]^k$*,* $r \in \{0, 1\}^k$*, and for all* i, j *with* $1 \leq i < j \leq k$*,* $c_i \neq c_j$ *and*

- $r_i = 0$ *and column* c_i *is increasing in* $[\#(0, r_{\dots i-1}) + 1, m - \#(1, r_{\dots i-1})]$*, or*
- $r_i = 1$ *and column* c_i *is decreasing in* $[\#(0, r_{\dots i-1}) + 1, m - \#(1, r_{\dots i-1})]$.

A column will be called *active* in an interval if it is either increasing or decreasing in this interval. What really matters are not the actual matrix entries x_{ij}, but rather the difference between successive entries x_{ij} and $x_{i+1,j}$. A more intuitive way to look at the problem may thus be in terms of a different matrix with the number of rows reduced by one, and entries describing the relative size of x_{ij} and $x_{i+1,j}$, *e.g.*, arrows pointing upward and downward, respectively, depending on whether $x_{ij} > x_{i+1,j}$ or $x_{ij} < x_{i+1,j}$, and empty cells if $x_{ij} = x_{i+1,j}$. According to this representation, a column can be deleted if it contains at least one arrow, and if all arrows in this column point in the same direction. The corresponding row to be deleted is the one at the base of the arrows.

We call matrix elimination (ME) the following computational problem: given a matrix $X \in \mathbb{N}^{m \times n}$, does X have an elimination sequence of length $\min(m - 1, n)$? The problem of eliminability of a column (CE) is given $k \in [n]$ and $d \in \{0, 1\}$ and asks whether there exists an elimination sequence (c, r) such that for some i, $c_i = k$ and $r_i = d$.

Let us formally establish the relationship between IDS and ME, and between IDE and CE.

Lemma 1 *IDS and IDE in anonymous games with two actions per player are polynomial time many-one equivalent to ME and CE, respectively, restricted to instances with* $m = n + 1$.

If we look for a way to solve a game as above with particular actions remaining for the different players, the problem becomes equivalent to MED and thus tractable.

Without restrictions on m and n, ME and CE are equivalent. We prove this statement by showing equivalence to the problem of deciding whether there exists an elimination sequence eliminating certain numbers of rows from the top and bottom of the matrix. Several other questions, like the one of an elimination sequence of a certain length, are equivalent as well.

Lemma 2 *CE and ME are equivalent under disjunctive truth-table reductions.*

If we restrict the problem to the case $m > n$, CE is at least as hard than ME in the sense that the latter can be reduced to the former while there is no obvious reduction in the other direction. In general, the case of ME where $m > n$ appears easier than the one where $m \leq n$. In the former *every* column has to appear somewhere in the elimination sequence, while in the latter the set of columns effectively needs to be partitioned into two sets of sizes m and $n - m$, respectively, of columns to be deleted and columns to be discarded right away.

A natural way of obtaining restricted versions of ME is to consider special classes of matrices, like matrices with entries in $\{0, 1\}$ or with a bounded number of maximal intervals in which a particular column is increasing or decreasing. One such restriction is to require that all columns are increasing or decreasing in $[1, m]$. It is not too hard to show that this makes the problem tractable irrespective of the dimensions of the matrix. We will formally state this result in the following section and prove it as a corollary of a more general result. Unfortunately, tractability of this restricted case does not tell us a lot about the complexity of ME in general. The latter obviously becomes almost trivial if the order of elimination for the columns is known, *i.e.*, if we ask for a specific vector $c \in [n]^k$ whether there exists a vector $r \in \{0, 1\}^k$ such that (c, r) is an elimination sequence. This observation directly implies membership in NP. More interestingly, deciding whether for a given $r \in \{0,1\}^k$ there exist $c \in [n]^k$ such that (c, r) is an elimination sequence is also tractable. The reason is the specific "life cycle" of a column. Consider a matrix X, two intervals $I, J \subseteq [m]$ over the rows of X such that $J \subseteq I$, and a column $c \in [n]$ that is active in both I and J. Then, c must also be active for any interval K such that $J \subseteq K \subseteq I$, and c must either be increasing for all three intervals, or decreasing for all three intervals. Thus, r determines for every $i \in [k]$ a set of possible values for c_i, and leaves us with a matching problem in a bipartite graph with edges in $[n] \times [k]$. A simple greedy algorithm is sufficient to solve this problem in polynomial time. Closer inspection reveals that it can in fact be decomposed into two independent matching problems on convex bipartite graphs, for which the best known upper bound is NC^2 (Glover, 1967). As we will see in the following section, yet another way to make the problem tractable is to provide a set of k pairs (c_j, r_j) that have to appear in corresponding places in the sequences of rows and columns, while leaving open the ordering of these pairs.

But what if nothing about c and r is known? Despite the fact that we can only eliminate the top or bottom row of the matrix in each step, this still amounts to an exponential number of possible sequences. The current bound for matching in convex bipartite graphs means that there is not much hope for constructing an algorithm that determines r nondeterministically and computes a matching on the fly. We can nevertheless use the above reasoning to recast the problem in the more general framework of *matching on paths*. For this, we will identify intervals and pairs of intervals over the rows of X by vertices and edges of a directed graph G, and will then label each edge (I, J) by the identifiers of the columns of X that take I to J. An elimination sequence of length k for X then corresponds to a path of length k in G which starts at the vertex corresponding to the interval $[1, m]$, such that there exists a matching of size k between the edges on this path and the columns of X. In particular, by fixing a particular path, we obtain the bipartite matching problem described above. A more detailed discussion of this problem is the topic of the following section. We first study the problem as such, and return to matrix elimination toward the end of the section.

5 Matched Paths

Let us formally define the matching problem described in the previous section. This problem generalizes the well-studied class of matching problems between two disjoint sets, or bipartite matching problems, by requiring that the elements of one of the two sets form a certain sub-structure of a combinatorial structure. This problem is particularly interesting from a computational perspective if identifying the underlying combinatorial structure can be done in polynomial time, as for paths like in our case, or for spanning trees.

Definition 5 (matching, matched path) *Let X be a set, Σ an alphabet, and $\sigma : X \to 2^\Sigma$ a labeling function assigning sets of labels to elements of X. Then, a* matching *of σ is a total function $f : X \to \Sigma$ such that for all $x, y \in X$, $f(x) \in \sigma(x)$ and $f(y) \neq f(x)$ if $y \neq x$.*

Let $G = (V, E)$ be a directed graph, Σ an alphabet, and $\sigma : E \to 2^\Sigma$ a labeling function for edges of G. Then, a matched path *of length k in G is a sequence $e_1, e_2, \ldots, e_k$ such that*

- *for all i, $1 \leq i < k$, $e_i \in E$ and there exist $u, v, w \in V$ such that $e_i = (u, v)$ and $e_{i+1} = (v, w)$, and*
- *the restriction of σ to $\{ e_i : 1 \leq i \leq k \}$ has a matching.*

We call matched path (MP) the following computational problem: given the explicit representation of a directed graph G with corresponding labeling function σ and an integer k, does there exist a matched path of length k in G? Variants of this problem can be obtained by asking for a matching that contains a certain set of labels, or a matched path between a particular pair of vertices. These variants have an interesting interpretation in terms of sequencing with resources and multi-dimensional constraints on the utilization of these resources: every resource can be used in certain states corresponding to vertices of a directed graph, and their use causes transitions between states. The goal then is to find a sequence that uses a specific set or a certain number of resources, or one that reaches a certain state.

In the context of this paper, we are particularly interested in instances of MP corresponding to instances of ME. We will see later on that the graphs of such instances are layered grid graphs (see, *e.g.*, Allender et al., 2006), and that the labeling function satisfies a certain convexity property. But let us look at the general problem for a bit longer. Greenlaw et al. (1995) consider the related *labeled graph accessibility problem*, which, given a directed graph G with a single label attached to each edge, asks whether there exists a path such that the concatenation of the labels along the path is a member of a context free language L given as part of the input. This problem is P-complete in general and LOGCFL-complete if G is acyclic. A matching, however, corresponds to a partial permutation of the members of the alphabet, and Ellul et al. (2004) have shown that the number of nonterminal symbols of any context-free grammar in Chomsky normal form for the permutation language over Σ grows faster than any polynomial in the size of Σ. It should thus not come as a surprise if the problem becomes harder when we ask for a matching. Indeed, MP bears some resemblance to the NP-complete problem *forbidden pairs* of finding a path in a directed or undirected graph if certain pairs of nodes or edges may not be used together (Gabow et al., 1976). Instead of trying to reduce forbidden pairs to MP, however, we show NP-hardness of a restricted version of MP using a slightly more complicated construction. We will then be able to build on this construction in Section 6.

In the following we restrict our attention to the case where G is a layered grid graph.

Definition 6 (layered grid graph) *A directed graph $G = (V, E)$ is an $m \times n$* grid graph *if $V = [m]_0 \times [n]_0$. An edge $(u, v) \in E$ is called* south edge *if for some i, j, $u = (i, j)$ and $v = (i + 1, j)$, and* east edge *if for some i, j, $u = (i, j)$ and $v = (i, j + 1)$. A grid graph is called* layered *if it contains only south and east edges.*

Theorem 3 *MP is NP-complete. Hardness holds even if G is a layered grid graph, and if for every $e \in E$, $|\sigma(e)| = 1$, and for every $\lambda \in \Sigma$, $|\{ e \in E : \lambda \in \sigma(e) \}| \leq 2$.*

Let us now return to matrix elimination. In light of Theorem 3, an efficient algorithm for ME would have to exploit additional structure of MP instances induced by instances of ME. It turns out that this structure is indeed quite restricted in that edges carrying a particular label satisfy a “directed” convexity condition: if a particular label λ appears on two edges $e = (u, v)$ and $e' = (u', v')$, then λ must appear on *all* south edges or on all east edges that lie on a path from u to v', but not both. In particular, if there is such a path, it cannot be that one of e and e' is a south edge and the other is an east edge. This fact is illustrated in Figure 2, which shows the labeled graph for the ME instance of Figure 1, as well as a matched path corresponding to an elimination sequence of maximum length.

Let us formally define the above property, along with a second property which requires the set of edges carrying a particular label to form a weakly connected subgraph of G. We henceforth concentrate on *complete* layered grid graphs, *i.e.*, ones that contain all south and all east edges.

Definition 7 (directed convexity, connectedness) *Let $G = (V, E)$ be a complete layered grid graph. A labeling function $\sigma : E \to 2^{\Sigma}$ for G is called directed convex if for every label $\lambda \in \Sigma$ and for every set of three edges e_1, e_2, e_3, $e_i = (u_i, v_i)$, such that u_2 is reachable from u_1, u_3 is reachable from u_2, and $\lambda \in \sigma(e_1) \cap \sigma(e_3)$, it holds that e_1 and e_3 have the same direction and $\lambda \in \sigma(e_2)$ if and only if e_2 has the same direction as well. A labeling function σ is called* connected *if for every $\lambda \in \Sigma$ and every pair of edges $e_1, e_2 \in E$ such that $\lambda \in \sigma(e_1) \cap \sigma(e_2)$ there exists $(u, v) \in E$ such that $\lambda \in \sigma(u, v)$ and both e_1 and e_2 are reachable from u.*

It is not too hard to see that instances corresponding to ME have a directed convex labeling function. Connectedness is related to a restricted version of ME which we term *matrix elimination with given directions* (MED): given a matrix X, a labeling function σ, and a total function $d : [n] \to \{0, 1\}$, does there exist

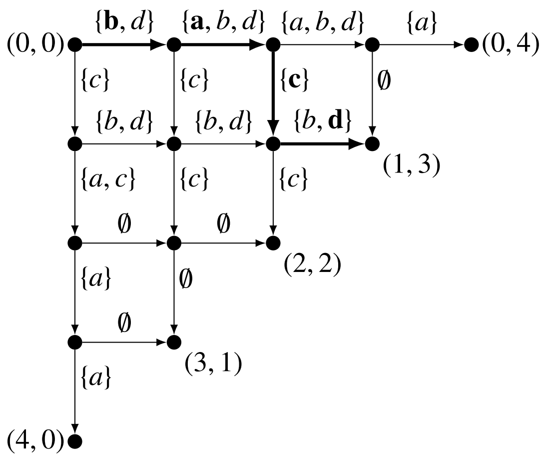

Figure 2: Labeled graph for the matrix elimination instance of Figure 1. A matched path and its matching are shown in bold.

an elimination sequence (c, r) with directions given by d, *i.e.*, one such that for all i, j satisfying $d(i) = j$ there is some $\ell \in \mathbb{N}$ for which $c_i = i$ and $r_i = j$.

Lemma 3 *ME is polynomial time many-one reducible to MP restricted to layered grid graphs and directed convex labeling functions. MED is polynomial time many-one equivalent to MP restricted to layered grid graphs and directed convex and connected labeling functions.*

Label a in the instance of Figure 2 serves as an example that the labeling function of an instance of MP corresponding to one of ME does not have to be connected, and it even appears on both east edges and south edges. On the other hand, MP can be solved in polynomial time if restricted to instances that do satisfy connectedness in addition to directed convexity. This also means that we can decide in polynomial time whether there exists an elimination sequence with a specific direction of elimination for every column of a matrix.

Theorem 4 *Let G be a layered grid graph, σ a directed convex and connected labeling function for G. Then MP for G, σ and $k = |\Sigma|$ is in P.*

Clearly, matrix elimination when all columns are active from the beginning is a special case of this theorem. With some additional work, we can derive a better upper bound.

Corollary 1 *Let $X \in \mathbb{N}^{m \times n}$ be a matrix every column of which is active in $[1, m]$. Then ME for X is in L.*

The complexity of ME remains open, and additional insights will be necessary to solve this question. The proof of Theorem 4 hinges on connectedness of the labeling function, and the case $m \leq n$ will probably add additional complications. On the other hand, directed convexity of the labeling function corresponding to an ME instance means that we cannot use a construction similar to the one used in the proof of Theorem 3 to show NP-hardness of MP.

Open Problem 1 *Let G be a layered grid-graph, σ a directed convex labeling function for G. Can MP for G and σ be decided in polynomial time? Is the case $k = |\Sigma|$ easier than the general case?*

6 Self-Anonymous Games With a Constant Number of Actions

It is natural to ask what happens for games with more than two actions, and whether there still exists a nice interpretation in terms of row and column eliminations in a matrix or matrix-like structure. It turns out there is such an interpretation, but its formulation is rather complicated. Consider a self-anonymous game with n players and k actions for each player. As before, the payoff of a particular player i only depends on the number of players, including himself, that play each of the different actions. For a particular player we thus have payoff values for each tuple $(j_1, j_2, \ldots, j_k)$ with $\sum_{\ell=1}^{k} j_\ell = n$. These can be represented as entries in a discrete simplex of dimension $k-1$. When writing down the payoffs of all players, one obtains a structure $X = (x_{i,j_1 \iddots j_k})_{i \in N, \sum_{\ell=1}^{k} j_\ell = n}$ where $x_{i,j_1 j_2 \iddots j_k} \in \mathbb{R}$ denotes the payoff of player $i \in N$ if for each ℓ, j_ℓ players play action a_ℓ. This structure has the aforementioned simplices as columns and resembles a triangular prism for the case $k = 3$.

Restricting our attention to dominance by pure strategies, action $a_\ell \in A$ weakly dominates action $a_m \in A$ for player $i \in N$ if i can never decrease his payoff by playing a_ℓ instead of a_m, no matter which actions the other players play, and if the payoff strictly increases for at least one combination of actions played by the other players. This corresponds to the values in the ith column of X being *increasing* from a_m to a_ℓ, *i.e.*, weakly increasing with a strict increase at some position $(j_1, j_2, \ldots, j_k)$. If m players have eliminated action a_ℓ, tuples with $j_\ell > n - m$ are no longer reachable, corresponding to a cut along the ℓth 0-face of the simplex. Eliminations of a particular action have the same effect on the payoff simplex of every single player and thus correspond to cuts along the respective edge of the prism in the case $k = 3$. Given a vector $d = (d_i)_{1 \leq i \leq k}$, $1 \leq d_i \leq n$, we will write $X(d)$ to denote the structure obtained by performing, for each i, d_i eliminations of action a_i from X, *i.e.*, $X(d) = (x_{i,j_1 j_2 \iddots j_k})_{j_i \leq n - d_i}$.

We are now ready to give a new formulation of solvability of a self-anonymous game using iterated dominance by pure strategies.

Fact 1 *Let $\Gamma = (N, (A_i)_{i \in N}, (p_i)_{i \in N})$ be a self-anonymous game, and let X be defined by $x_{i,j_1 j_2 \iddots j_k} = p_i(j_1, j_2, \ldots, j_k)$ for all $i \in N$ and $j_1, j_2, \ldots, j_k \in \mathbb{N}_0$ such that $\sum_{\ell=1}^{k} j_\ell = n$. Then, Γ is solvable using iterated dominance by pure strategies if there exists a pair (c, r) of sequences $c \in N^{(k-1)n}$ and $r \in A^{(k-1)n}$ such that*

(i) $|\{\, 1 \leq i \leq (k-1)n : c_i = j \,\}| = k - 1$ *for all* $j \in N$,

(ii) $c_i = c_j$ *and* $r_i = r_j$ *implies* $i = j$ *for all* $1 \leq i, j \leq (k-1)n$, *and*

(iii) for each i, $1 \leq i \leq (k-1)n$, there exists some $k^* \in A$ such that, for all $j < i$, $c_j \neq c_i$ or $r_j \neq r^*$, and c_i is increasing from r_i to r^* in $X(r_1, r_2, \ldots, r_{i-1})$.

That is, a game is solvable if there exists a sequence of $(k-1)n$ eliminations of an action by a player such that (i) every player deletes exactly $k-1$ times, (ii) no player deletes the same action twice, and (iii) every action is deleted using some other action that has not itself been deleted.

The left hand side of Figure 3 shows the payoffs of a particular player in a self-anonymous game with $n = 3$ and $k = 3$. Compared to matrix elimination as introduced in Definition 4 and illustrated in Figure 1, we notice an interesting shift. Curiously, this shift has nothing to do with the added possibility of dominance by mixed strategies in games with more than two actions. Rather, a particular action $a \in A$ may now be eliminated by either one of several other actions in $A \setminus \{a\}$, and the situations where a a can be eliminated no longer form a convex set. Recalling the proof of Theorem 3, our strategy becomes clear: try to construct a layered grid graph with labels for which the existence of a matched path is NP-hard to decide, and which

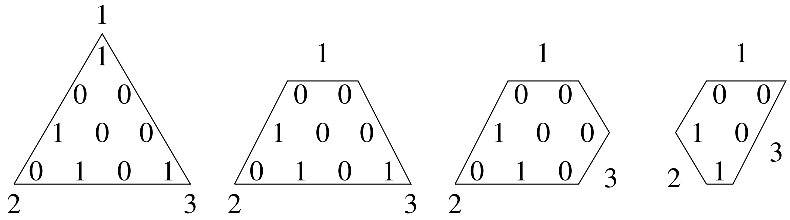

Figure 3: Payoffs of a particular player in a self-anonymous game with $n = 3$ and $k = 3$. Initially all actions are pairwise undominated. If one of the other players eliminates action 1, action 3 weakly dominates action 1. Action 1 then becomes undominated if some player deletes action 3, and dominated by action 2 if one more player deletes action 3, and some player deletes action 2.

is induced by a self-anonymous game with three actions for each player. It turns out that this is indeed possible.

Theorem 5 *IDS and IDE are NP-complete. Hardness holds even for self-anonymous games with three actions and only two different payoffs.*

Acknowledgements

This material is based upon work supported by the Deutsche Forschungsgemeinschaft under grants BR 2312/3-1 and BR 2312/3-2. We thank Hermann Gruber, Paul Harrenstein, Tim Roughgarden, and Michael Tautschnig for helpful discussions, and apologize to Edith Hemaspaandra for spoiling the sunset at White Sands.

References

E. Allender, D. A. Mix Barrington, T. Chakraborty, S. Datta, and S. Roy. Grid graph reachability problems. In *Proc. of 21st Annual IEEE Conference on Computational Complexity (CCC)*, pages 299–313, 2006.

K. R. Apt. Uniform proofs of order independence for various strategy elimination procedures. *Contributions to Theoretical Economics*, 4(1), 2004.

F. Brandt, F. Fischer, and Y. Shoham. On strictly competitive multi-player games. In Y. Gil and R. Mooney, editors, *Proc. of 21st AAAI Conference*, pages 605–612. AAAI Press, 2006.

F. Brandt, F. Fischer, and M. Holzer. Symmetries and the complexity of pure Nash equilibrium. In W. Thomas and P. Weil, editors, *Proc. of 24th STACS*, volume 4393 of *LNCS*, pages 212–223. Springer, 2007.

V. Conitzer and T. Sandholm. Complexiy of (iterated) dominance. In *Proc. of 6th ACM-EC Conference*, pages 88–97. ACM Press, 2005.

C. Daskalakis and C. H. Papadimitriou. Computing equilibria in anonymous games. In *Proc. of 48th FOCS Symposium*. IEEE Press, 2007.

K. Ellul, B. Krawetz, J. Shallit, and M.-W. Wang. Regular expressions: new results and open problems. *Journal of Automata, Languages and Combinatorics*, 9(2–3):233–256, 2004.

H. N. Gabow, S. N. Maheshwari, and L.J. Osterweil. On two problems in the generation of program test paths. *IEEE Transactions on Software Engineering*, 2(3):227–231, 1976.

I. Gilboa, E. Kalai, and E. Zemel. The complexity of eliminating dominated strategies. *Mathematics of Operations Research*, 18(3):553–565, 1993.

F. Glover. Maximum matching in convex bipartite graphs. *Naval Research Logistics Quarterly*, 14:313–316, 1967.

R. Greenlaw, H. J. Hoover, and W. L. Ruzzo. *Limits to Parallel Computation*. Oxford University Press, 1995.

R. D. Luce and H. Raiffa. *Games and Decisions: Introduction and Critical Survey*. John Wiley & Sons Inc., 1957.

H. Moulin. Dominance solvable voting schemes. *Econometrica*, 47:1337–1351, 1979.

C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

C. H. Papadimitriou and T. Roughgarden. Computing equilibria in multi-player games. In *Proc. of 16th SODA*, pages 82–91. SIAM, 2005.

I. Parberry. On the computational complexity of optimal sorting network verification. In *Proc. of Conference on Parallel Architectures and Languages Europe (PARLE)*, volume 505 of *LNCS*, pages 252–269. Springer, 1991.

A Proof of Theorem 1

Proof: Recall that circuit satisfiability (CSAT), *i.e.*, deciding whether a Boolean circuit has a satisfying assignment, is NP-complete (see, *e.g.*, Papadimitriou, 1994). We provide a reduction from CSAT to IDS and IDE for self-symmetric games. Hardness for the other types of symmetries follows by inclusion. For a particular circuit C with inputs $M = \{1, 2, \ldots, m\}$, we define a game Γ with $n \geq m$ players and actions $A = \{ a_j^0, a_j^1 : j \in M \} \cup \{a^0, a^1\}$. An action profile s of Γ where $\#(a_j^0, s) + \#(a_j^1, s) = 1$ for all $j \in M$, *i.e.*, one where exactly one action of each pair a_j^0, a_j^1 is played, directly corresponds to an assignment c of C, the jth bit c_j of this assignment being 1 if and only if a_i^1 is played. Observe that in this case the auxiliary actions a^0 and a^1 have to be played by exactly $n - m$ players. We can thus also identify action profiles of Γ that correspond to a *satisfying* assignment of C. Now define the (common) payoff function p by letting $p(s) = 1$ if $\#(a^0, s) + \#(a^1, s) = n$ or if s corresponds to a satisfying assignment of C and $\#(a^1, s) = n - m$, and $p(s) = 0$ otherwise. Since the payoff function is the same for all players, and the payoff only depends on the number of players playing each of the different actions, Γ is self-symmetric. Furthermore, both a^0 and a^1 weakly dominate a_j^0 and a_j^1 for all $j \in M$.

Assume that C has a satisfying assignment. In this case a^1 weakly dominates a^0, and the elimination of a^0 by all players, followed by the elimination of a_j^0 and a_j^1 for all $j \in M$, again by all players, leaves a^1 as the sole action for each player. If on the other hand there is no satisfying assignment of C, then the payoff for a player playing a^0 or a^1, respectively, is identical under every action profile for the other players, so neither of them can be eliminated *via* iterated dominance. Thus, a^0 is eliminable for any player, and Γ is solvable *via* iterated dominance with action a^1 remaining for each player, if and only if C has a satisfying assignment. The transformation from C to Γ essentially works by writing down a Boolean circuit that computes p_i. Observing that this can be done in time polynomial in the size of C if $n \leq 2^{|A|}$ completes the proof. □

B Proof of Theorem 2

Proof: Since all players have identical payoff functions, a state of iterated dominance elimination can be represented as a vector that counts, for each set $C \subseteq A$, the number of players that have eliminated exactly the actions in C. This vector has constant dimension if the number of actions is constant. The value of each entry is bounded by n, so the number of different vectors is polynomial in n and thus in the size of the game. The elimination process can then be described as a graph that has the above vectors as vertices and a directed edge between two such vectors if the second one can be obtained from the first by adding 1 to some component, and if the action corresponding to this component can indeed be eliminated in the state described by the first vector. For dominance by mixed strategies, this neighborhood relation can be computed in polynomial time *via* linear programming. This reduces the computational problems related to iterated dominance to reachability problems in a directed graph, which in turn can be decided in nondeterministic logarithmic space and thus in polynomial time. For IDS, we need to find a directed path from (the vertex corresponding to) the zero vector to some vector with sum $n(k-1)$ or to some vector describing a particular state of elimination, respectively. For IDE, we need to find a path where the respective action is deleted while traversing the final edge. □

C Proof of Lemma 1

Proof: Brandt et al. (2007) observe that an anonymous game with two actions can be transformed into a self-anonymous game while preserving pure Nash equilibria. The same is true for (iterated) dominance when there are only two different payoffs, so it suffices to prove the equivalences for self-anonymous games. Consider a self-anonymous game $\Gamma = (N, (A_i)_{i\in N}, (p_i)_{i\in N})$ such that for all $i \in N$, $A_i = \{0, 1\}$, and assume w.l.o.g. that for all $i \in N$ and all $s \in \{0, 1\}^N$, $p_i(s) \in \mathbb{N}$. Since in games with two actions it suffices to consider dominance by pure strategies, we can otherwise construct a game with payoffs from the natural numbers that is equivalent w.r.t. iterated dominance. Now write down the payoffs of Γ in an $(|N| + 1) \times |N|$ matrix X such that the jth column contains the payoffs of player $j \in N$ for the different numbers of players playing action 1, *i.e.*, $x_{ij} = p_j(s)$ where $\#(1, s) = i$. Then, the jth column of X is increasing in an interval $[k_0, k_1]$ if and only if action 1 dominates action 0 for player j given that at least $k_0 - 1$ and at most $k_1 - 1$ other players play action 1. Analogously, the jth column is decreasing in such an interval if action 0 dominates action 1 under the same conditions. If player j eliminates action 0 or 1, respectively, this decreases the number of players that can still play the respective action, corresponding to the deletion of the top or bottom row of X, respectively. Furthermore, since every player has only two actions, the corresponding column of the matrix can be ignored as soon as one of them has been deleted. Observing that the above does not impose any restrictions on the resulting matrix apart from its dimensions, equivalence of the corresponding problems follows. □

D Proof of Lemma 2

Proof: We provide reductions between both CE and ME and the problem of *matrix elimination up to an interval* (IE): given a matrix X and two numbers k_0 and k_1, does there exist an elimination sequence (c, r) of X such that $\#(0, r) = k_0$ and $\#(1, r) = k_1$?

To reduce ME to IE, observe that X is a "yes" instance of ME if and only if X and some interval of length $\max(1, m - n)$ form a "yes" instance of IE. Analogously, to reduce CE to IE, X and (i, d) for some $i \in [n]$ and $d \in \{0, 1\}$ form a "yes" instance of CE if there is an interval over the rows of X in which column i is active in the direction corresponding to d and which together with X forms a "yes" instance of IE.

$$
\begin{array}{c|ccc|ccc|ccc|c}
 & & & & \multicolumn{3}{c|}{n+\left\lfloor\frac{m-(i+j)}{2}\right\rfloor} & \multicolumn{3}{c|}{n+\left\lceil\frac{m-(i+j)}{2}\right\rceil} & \\
\hline
 & x_{11} & \cdots & x_{1n} & 0 & \cdots & 0 & 1 & \cdots & 1 & \\
 & \vdots & \ddots & \vdots & \vdots & & \vdots & \vdots & & \vdots & \\
 & x_{i1} & \cdots & x_{in} & 0 & \cdots & 0 & 1 & \cdots & 1 & \\
\hline
 & \vdots & & \vdots & 1 & \cdots & 1 & 0 & \cdots & 0 & \\
n & x_{i1} & \cdots & x_{in} & \vdots & & \vdots & \vdots & & \vdots & n+\left\lceil\frac{m-(i+j)}{2}\right\rceil \\
\cline{1-4}
 & x_{i+1,1} & \cdots & x_{i+1,n} & & & & & & & \\
 & \vdots & & \vdots & 1 & \cdots & 1 & 0 & \cdots & 0 & \\
\cline{5-11}
 & & & & 0 & \cdots & 0 & 1 & \cdots & 1 & \\
 & x_{m-j-1,1} & \cdots & x_{m-j-1,n} & \vdots & & \vdots & \vdots & & \vdots & n+\left\lfloor\frac{m-(i+j)}{2}\right\rfloor \\
\cline{1-4}
n & x_{m-j,1} & \cdots & x_{m-j,n} & & & & & & & \\
 & \vdots & & \vdots & 0 & \cdots & 0 & 1 & \cdots & 1 & \\
\hline
 & x_{m-j,1} & \cdots & x_{m-j,n} & 1 & \cdots & 1 & 0 & \cdots & 0 & \\
 & \vdots & \ddots & \vdots & \vdots & & \vdots & \vdots & & \vdots & \\
 & x_{m1} & \cdots & x_{mn} & 1 & \cdots & 1 & 0 & \cdots & 0 & \\
\hline
\end{array}
$$

Figure 4: Matrix Y used in the proof of Lemma 2

For a reduction of IE to either ME or CE, let $X \in \mathbb{N}^{m\times n}$ and consider the $(m + 2n) \times (3n + m - (i + j))$ matrix Y shown in Figure 4. We claim that a column with index greater than n, and the entire matrix, can be eliminated if and only if X has an elimination sequence (c, r) satisfying $\#(0, r) = i$ and $\#(0, r) = j$.

For the direction from left to right, assume that (c, r) is an elimination sequence for X as above and define (c', r') by

$$
c'_k = \begin{cases} c_k & \text{if } 1 \le k \le i + j, \\ n + k - (i + j) & \text{if } i + j < k \le m + 2n, \text{ and} \end{cases}
$$

$$
r'_k = \begin{cases} r_k & \text{if } 1 \le k \le i + j, \\ 1 & \text{if } i + j < k \le n + \lceil\frac{m+(i+j)}{2}\rceil, \\ 0 & \text{if } n + \lceil\frac{m+(i+j)}{2}\rceil < k \le m + 2n. \end{cases}
$$

It is easily verified that (c', r') is an elimination sequence of length $m + 2n$ for Y, *i.e.*, one that eliminates Y entirely.

For the direction from left to right, consider an elimination sequence (c', r') of length $m + 2n$ for Y. Define ℓ to be the smallest index k for which $c'_k > n$, and let $I = [\#(0, r_{\ldots\ell-1}) + 1, m - \#(1, r_{\ldots\ell-1})]$. Clearly, $\ell > i + m - j$. Now define a sequence c that contains the first i elements c'_k of c' for which $r'_k = 0$, and the first j elements c'_k for which $r'_k = 1$, in the same order in which they appear in c'. Define r to be sequence of corresponding elements of r'. Then, (c, r) is an elimination sequence for Y, because the set of active columns is the same for I and $[i, m - j]$, and also for all intervals in between. Furthermore, c only contains columns with index at most n. Thus (c, r) is also an elimination sequence for X, and the number of rows eliminated from the top and bottom is exactly as required. $\square$

E Proof of Theorem 3

Proof: Membership in NP is immediate. We can simply guess a sequence of edges of the required length as well as an assignment of labels to these edges, and verify in polynomial time that we have in fact obtained a path and a matching on this path.

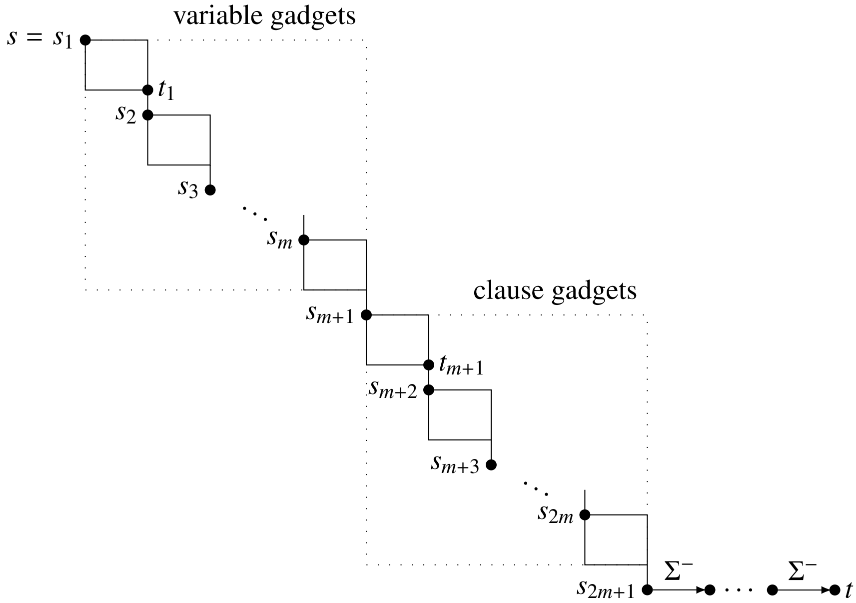

Figure 5: Overall structure of the layered grid graph G used in the proof of Theorem 3

For *hardness*, we provide a reduction from the NP-complete problem *balanced one-in-three 3SAT* (B3SAT) (Parberry, 1991) to MP with the above restrictions. A B3SAT instance over a set U of variables is given by a set $C \subseteq U^3$ of clauses of length three such that every variable occurs in *exactly* three clauses, *i.e.*, $|\{ (x_1, x_2, x_3) \in C : x_i = x \}| = 3$ for all $x \in U$. An instance is said to be *satisfiable* if there exists an assignment to the variables such that *exactly* one element of each clause is *true*, *i.e.*, a set $S \subseteq V$ such that $|\{ i : x_i \in S \}| = 1$ for all $(x_1, x_2, x_3) \in C$. It is easily verified that $|U| = |C|$ for every instance of B3SAT, and $|U| = 3|S|$ for every assignment S satisfying C (in particular, satisfiable instances must have $|U|$ divisible by three). Given a particular B3SAT instance C, we construct an MP instance consisting of a complete layered grid graph $G = (V, E)$ and a labeling function $\sigma : E \to 2^\Sigma$ such that a path between two designated nodes s and t of G has a matching if and only if C is satisfiable. For the moment, we will put aside the restrictions that all sets of labels are singletons and every label occurs on at most two different edges. This allows us to prove hardness for a labeled graph with special structure, which will then also be used in the proof of Theorem 5. At the end of the current proof, we will see that the construction can easily be modified to meet the above requirements for σ.

Now let $m = |U|$, and define G as a complete $6m \times 5m$ layered grid graph. Figure 5 illustrates the overall structure of the labeling function σ. From s to t, G is composed of gadgets for each of the variables of C, gadgets for the clauses, and a final path of $2m$ east edges. We write s_i and t_i, $1 \leq i \leq 2m$, for the initial and final node of the ith of these gadgets. Before we take a closer look at both types of gadgets, let us define the set Σ of labels available for labeling edges of G. For every variable x_i of C, $1 \leq i \leq m$, we have six labels λ_{ij}, $1 \leq j \leq 6$, appearing on east edges only. Labels λ^v_{ij}, $j \in \{1, 2\}$, and λ^c_{ij}, $j \in \{1, 2, 3\}$, on the other hand, are exclusive to south edges. The labeling function σ is such that labels on east edges appear on *every* east edge in the respective rows of the grid, and labels on south edges appear on *every* south edge in the respective columns. Furthermore, for each label, there are at most two sets of subsequent rows or columns where this label appears. Intuitively, the gadget for variable x_i lies at the intersection of columns carrying labels λ^v_{ij} and rows carrying labels λ_{ij}, while the gadget for clause c_i lies at the intersection of columns carrying labels λ_{ij} and rows carrying labels for the variables that appear in c_i.

Figures 6 and 7 illustrate the gadgets for variables and clauses of C. The labeling function is defined

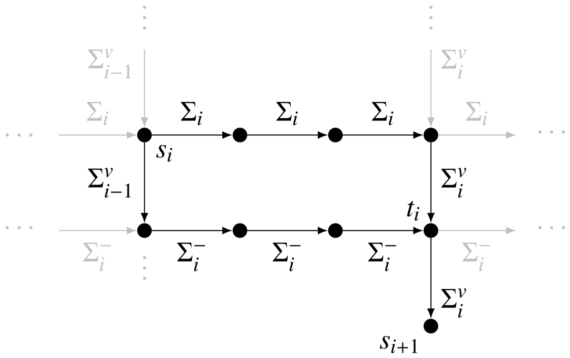

Figure 6: Gadget for variable x_i used in the proof of Theorem 3

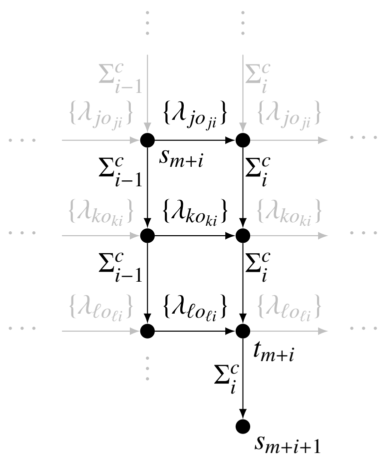

Figure 7: Gadget for clause $c_i = (x_j \vee x_k \vee x_\ell)$ used in the proof of Theorem 3. o_{ji} denotes the number of times variable x_j occurs in clauses up to and including c_i.

using the following subsets of Σ:

$$
\begin{array}{rcll}
\Sigma_i &=& \{\lambda_{i1}, \lambda_{i2}, \lambda_{i3}\} & \\
\Sigma^-_i &=& \{\lambda_{i4}, \lambda_{i5}, \lambda_{i6}\} & \\
\Sigma^- &=& \bigcup_{1 \le i \le m} \Sigma^-_i & \\
\Sigma^v_0 &=& \{\lambda^v_{11}\} & \\
\Sigma^v_i &=& \{\lambda^v_{i,1}, \lambda^v_{i,2}, \lambda^v_{i+1,1}\} & \text{for } 1 \le i \le m-1 \\
\Sigma^v_m &=& \Sigma^c_0 = \{\lambda^v_{m1}, \lambda^v_{m2}, \lambda^c_{11}, \lambda^c_{12}\} & \\
\Sigma^c_i &=& \{\lambda^c_{i1}, \lambda^c_{i2}, \lambda^c_{i3}, \lambda^c_{i+1,1}, \lambda^c_{i+1,2}\} & \text{for } 1 \le i \le m-1 \\
\Sigma^c_m &=& \{\lambda^c_{m1}, \lambda^c_{m2}, \lambda^c_{m3}\} &
\end{array}
$$

Labels in Σ_i and Σ^-_i correspond to a positive and negative assignment of the ith variable, respectively. Sets Σ^v_i and Σ^c_i contain auxiliary labels for the ith variable gadget and the ith clause gadget. Note that while labels in sets Σ^-_i are marked as "negative," assigning them to edges in the variable gadget actually sets variable x_i to *true*, because the selection of labels from the corresponding set Σ_i will have to take place in the respective clause gadgets. Returning to Figure 5, the final path of east edges from s_{2m+1} to t has length $2m$, and each of the edges carries all "negative" variable labels λ_{ij} for $1 \le i \le m$ and $j \in \{4, 5, 6\}$. It is readily appreciated that G and σ can be constructed from C in polynomial time.

Two properties of G and σ will be useful in the following. First, every path from s to t traverses exactly $6m$ east edges and $5m$ south edges, which equals the overall number of labels for both directions. Secondly, a matched path from s to t must traverse every edge (t_i, s_{i+1}) for $1 \leq 1 \leq 2m$. To see the latter, assume for contradiction that there is an edge $(v, v') \neq (t_i, s_{i+1})$ on the path such that t_i is reachable from v but not from v'. If v is to the west from t_i, *i.e.*, (v, v') is a south edge, then the number of south edges on the path up to v' exceeds the number of labels available for these. If v is to the north from t_i, *i.e.*, (v, v') is an east edge, then the number of labels for south edges that do not appear on any edge reachable from v' exceeds the number of south edges on the path to v'. In both cases, the number of edges differs from the number of labels available for these edges, and the path cannot have a matching.

Now assume that there exists a satisfying assignment for C. We construct a path from s to t *via* all s_i and t_i, $1 \leq 1 \leq 2m$, as well as a matching for this path of size $|\Sigma|$. For vertices s_i and t_i, $1 \leq i \leq m$, *i.e.*, the gadget for variable x_i, we select the path labeled with elements of Σ_i if $x_i = true$, and the path labeled with elements of Σ_i^- otherwise. For nodes s_i and t_i, $m < i \leq 2m$, *i.e.*, the gadget for clause c_i, we select the (unique) path labeled with λ_{jk} for some k such that $x_j = true$. In both cases, we arbitrarily assign one of the available labels to each edge. By this, "positive" labels $\lambda \in \Sigma_i$ corresponding to variable x_i are assigned to edges in clause and variable gadgets, respectively, depending on whether or not $x_i = true$. Every "positive" label is used exactly once on the path from s to t_{2m}, and none of the "negative" labels is used more than once. Since a satisfying assignment must set exactly $m/3$ variables to *true*, and since, by construction of G, $2m$ of the "negative" labels are not assigned to any edge on a labeled path from from s to t_{2m}, arbitrarily assigning these labels to the edges on the path from t_{2m} to t yields a matching for the path from s to t.

Conversely assume that there is a matched path from s to t. As observed above, this path must traverse s_i and t_i for all $1 \leq i \leq 2m$. Furthermore, by construction of G, the "positive" labels for a particular variable x_i either all have to be assigned to edges in the gadget x_i, or to edges in the gadgets of the clauses where x_i appears, but not both. It is then easily verified that setting a variable to *true* if and only if the corresponding "positive" labels are assigned to edges in clause gadgets yields a satisfying assignment. Thus, some path from s to t in G has a matching if and only if C is satisfiable.

It remains to be shown that the above construction can be simplified such that every edge can be labeled with exactly one label and every label appears on at most two different edges. For this, we first remove all edges that cannot be part of a path from s that has a matching, *i.e.*, those that are not part of any gadget. Then, for every set of labels defined above, the number of edges labeled with this set *within a particular gadget* equals the cardinality of the set, and we can assign a different singleton to each of these edges. The path from s_{2m+1} to t requires some additional attention. We know that, at the time we have found a path from s to s_{2m+1} that does not use any of the labels more than once, exactly $2m$ labels in Σ^- have not yet been assigned to an edge, but we do not know which. To ensure that the remaining labels can be chosen in an arbitrary order, we replace the path starting at s_{2m+1} by $2m/3$ additional gadgets of the form shown in Figure 8, which use $2m^2/3$ additional labels λ_{ji}^e for $1 \leq j \leq m$ and $1 \leq i \leq 2m/3$. It is easily verified that the modified labeling function satisfies the desired constraints. □

F Proof of Lemma 3

Proof: First consider the reduction from ME to MP. For a matrix $X \in \mathbb{N}^{m \times n}$, define a layered grid graph $G = (V, E)$ with $V = [m]_0 \times [n]_0$ and a labeling function $\sigma : E \mapsto 2^{[n]}$ such that for all $\lambda \in [n]$, $\lambda \in \sigma(e)$ if for some $i, j \in \mathbb{N}$, $e = ((i, j), (i+1, j))$ and column ℓ of X is increasing in $[i+1, m-j]$, or $e = ((i, j), (i, j+1))$ and column ℓ of X is decreasing in $[i+1, m-j]$. Now consider $k \in \mathbb{N}$, $c \in [n]^k$, and $r \in \{0, 1\}^k$. Let $p = e_1, e_2, \ldots, e_k$ be a path in G such that $e_1 = ((0, 0), v)$ for some $v \in V$, and e_i is a south edge if and only if $r_i = 0$. Further define a function $f : E \to [n]$ by letting $f(e_i) = c_i$ for all $i \leq k$. It is not too hard to see that

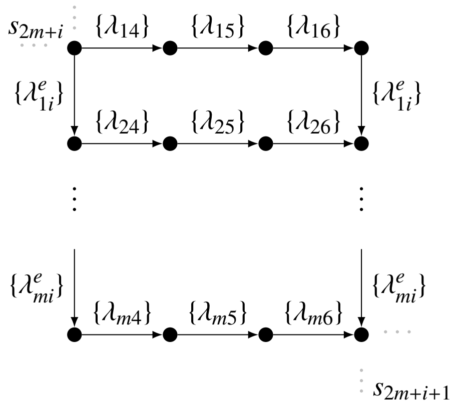

Figure 8: Gadget for exhausting the remaining labels, used in the Proof of Theorem 3

(c, r) is an elimination sequence of X if and only if f is a matching for the restriction of σ to the edges on p.

For directed convexity of σ, consider $e_1, e_2, e_3 \in E$, $e_i = (u_i, v_i)$, such that u_2 is reachable from u_1 and u_3 is reachable from u_2. For $\ell = 1, 2, 3$, define an interval $I_\ell = [i + 1, m - j]$ for $i, j \in \mathbb{N}$ such that $e_\ell = ((i, j), v)$ for some $v \in V$. Further consider $\lambda \in \sigma(e_1) \cap \sigma(e_3)$. By definition of σ, column λ of X must be active in both I_1 and I_3. Since $I_3 \subseteq I_1$, λ must either be increasing in both of them, or decreasing in both of them. Furthermore, since $I_3 \subseteq I_2$ and $I_2 \subseteq I_1$, column λ must also be increasing or decreasing in I_2, respectively.

For MED, consider a total function $d : [n] \to \{0, 1\}$ and define $\sigma' : E \mapsto 2^{[n]}$ such that for all $e \in E$ and $\lambda \in [n]$, $\lambda \in \sigma'(e)$ if $\lambda \in \sigma(e)$ and if either e is a south edge and $f(\lambda) = 0$ or e is an east edge and $f(\lambda) = 1$. It is not hard to see that σ' is directed convex and connected. On the other hand consider a layered grid graph $G = (V, E)$ and a directed convex and connected labeling function σ. Then, for every $\lambda \in \Sigma$, there exists a unique pair of vertices $u, v \in V$ such that $\lambda \in \sigma(e)$ for exactly those south edges or exactly those east edges e that are reachable from u but not from v. It is now possible to define a matrix X with a column for λ that is active exactly in every interval I such that $I \subseteq [i, j]$ and $I \cap [i', j'] \neq \emptyset$, and increasing if $f(\lambda) = 0$ and decreasing if $f(\lambda) = 1$. By the same reasoning as above, elimination sequences of X correspond to matched paths of G and σ with initial vertex $(0, 0)$. □

G Proof of Theorem 4

Proof: It suffices to show how to decide whether there exists a matched path from $s = (0, 0)$ to a particular vertex $t = (k_s, k_e)$ such that $k_s + k_e = k$. Different values for t can then be checked sequentially.

Given a path p from some vertex $v_1 \in V$ to t, we define two labeling functions $\sigma^p_s : [k_s] \to \Sigma$ and $\sigma^p_e : [k_e] \to \Sigma$, one for south edges and one for east edges of paths from s to t. We will argue that a pair of matchings for σ^p_s and σ^p_e can easily be combined into a matching for p, while nonexistence of a matching for either of the two implies that a large set of paths in G cannot be *matched* paths. The latter will ultimately provide us with a succinct certificate that a particular pair of a graph G and a labeling function σ does *not* have a matched path of length k.

More formally, consider a complete layered grid graph $G = (V, E)$ and a labeling function $\sigma : E \to 2^\Sigma$.

For a path $p = e_\ell, e_{\ell+1}, \dots, e_k$, define σ_s^p and σ_e^p such that for every $\lambda \in \Sigma$,

$$\begin{aligned}
\lambda \in \sigma_s^p(i) \quad & \text{if there exists a path } e'_1, e'_2, \dots, e'_k \text{ with } e'_i = e_i \text{ for all } i \geq \ell, \\
& \text{and } j \in [k],\, i' \in [k_e] \text{ such that} \\
& e_j = ((i-1, i'), (i, i')) \text{ and } \lambda \in \sigma(e_j), \text{ and} \\
\lambda \in \sigma_e^p(i) \quad & \text{if there exists a path } e'_1, e'_2, \dots, e'_k \text{ with } e'_i = e_i \text{ for all } i \geq \ell, \\
& \text{and } j \in [k],\, i' \in [k_s] \text{ such that} \\
& e_j = ((i', i-1), (i', i)) \text{ and } \lambda \in \sigma(e_j).
\end{aligned}$$

In other words, σ_s^p and σ_e^p provide an "optimistic" version of the matching problems obtained by restricting σ to a path in G that contains p as a sub-path, by allowing a certain label to be matched to the ith south edge or east edge of these paths, respectively, if it appears on the ith edge in the respective direction of *some* such path. It follows from directed convexity and connectedness of σ that for every path p, σ_s^p and σ_e^p are convex functions and $\{\, a \in \sigma_s^p(i) : i \in [k_s] \,\} \cap \{\, a \in \sigma_e^p(i) : i \in [k_e] \,\} = \emptyset$. We can further assume w.l.o.g. that for every p, σ_s^p and σ_e^p have images of size k_s and k_e, respectively.

Now let p be a path from s to t. By definition, there is a one-to-one correspondence between σ_s^p and the restriction of σ to south edges of p, and also between σ_e^p and the restriction of σ to east edges of p. Any pair of matchings for σ_s^p and σ_e^p thus directly corresponds to a matching for the restriction of σ to p, and existence of the former implies that p is a matched path.

On the other hand, consider a path $p = e_\ell, e_{\ell+1}, \dots, e_k$ and an edge $e \in E$ such that $e = (u, v)$ and $e_\ell = (v, w)$ for some $u, v, w \in V$. Denote $p' = e, e_\ell, e_{\ell+1}, \dots, e_n$, and assume that both σ_s^p and σ_e^p have a matching while $\sigma_s^{p'}$ or $\sigma_e^{p'}$ does not.

First consider the case where e is an east edge, and where the function that does not have a matching is $\sigma_e^{p'}$. Let $i, j \in \mathbb{N}$ such that $u = (i, j)$. By definition, σ_e^p and $\sigma_e^{p'}$ only differ w.r.t. labels λ such that $\lambda \in \sigma_e^p(j')$ if and only if $j' < j$. Since $\sigma_e^{p'}$ is a convex function that does not have a matching, and since the image of $\sigma_e^{p'}$ has size k_e, there has to be some interval in $[k_e]$ the size of which is strictly larger, and some interval the size of which is strictly smaller than the number of labels $\sigma_e^{p'}$ assigns exclusively to elements of this interval. Furthermore, every matching f of σ_e^p must satisfy $f(j) \notin \sigma(e)$, since a matching with $f(j) \in \sigma(e)$ would also be a matching for $\sigma_e^{p'}$. This means that there actually must exist an interval I of the second type such that $I \subseteq [1, j-1]$. Now consider any path p'' from a vertex u' south of u to t, *i.e.*, a vertex $u' = (i', j)$ such that $i' > i$. Clearly, the number of labels appearing exclusively in I cannot be smaller for $\sigma_e^{p''}$ than it is for $\sigma_e^{p'}$. This means that $\sigma_e^{p''}$ does not have a matching, and thus that no matched path of G and σ can traverse u'.

Now assume that the function that does not have a matching is $\sigma_s^{p'}$. This again means that there has to be an interval such that the number of different labels assigned by $\sigma_s^{p'}$ to elements of this interval is strictly smaller than the length of this interval. Since σ_s^p has a matching, since the restrictions of σ_s^p and $\sigma_s^{p'}$ to $[i, k_s]$ are identical, and by convexity of $\sigma_s^{p'}$, there has to exist an interval I with this property such that $I \subseteq [0, i-1]$. Now consider any path p'' from a vertex u' west of u to t, *i.e.*, a vertex $u' = (i, j')$ such that $j' < j$. By definition, for any $\lambda \in \Sigma$, $\lambda \in \sigma_e^{p''}$ only if $\lambda \in \sigma_e^{p'}$, such that the number of different labels assigned by $\sigma_s^{p''}$ to elements of I is strictly smaller than $|I|$. Thus $\sigma_s^{p''}$ does not have a matching, and thus no matched path of G and σ can traverse u'.

If e is a south edge, then by symmetrical arguments either no matched path of G and σ can traverse any vertex north of u, or no such path can traverse any vertex east of u.

Now consider an algorithm which starts at t and tries to iteratively construct a path from s to t by traversing edges of G backwards. Given a path p_ℓ of length ℓ from a vertex v_ℓ to t, the algorithm selects $p_{\ell+1}$ to be a path of length $\ell + 1$ containing p_ℓ as a sub-path such that both $\sigma^s p_{\ell+1}$ and $\sigma^s p_{\ell+1}$ have a matching.

If the algorithm runs for k steps, we obtain a path p_k from s to t with this property, *i.e.*, a matched path. Assume on the other hand that for some ℓ no path satisfying the above requirements exists, and denote by P the set of paths obtainable by adding a predecessor of v_ℓ to p_ℓ (this set contains one or two paths depending on whether v_ℓ has one or two predecessors). Then, for every $p \in P$, one of σ_s^p and σ_e^p does not have a matching, and from the above reasoning we obtain a set of vertices such that no matched path can traverse any of these vertices. It is easily verified that the union of these sets for the different elements of P always forms a cut that separates s from t, implying that a matched path from s to t cannot exist. □

H Proof of Corollary 1

Proof: Consider the graph G and the labeling function σ corresponding to X. For any path p in G with final vertex t, consider the functions σ_s^p and σ_e^p defined in the proof of Theorem 4. Since every column of X is active in $[1, m]$, σ_s^p and σ_e^p are convex. Moreover, for every label $\lambda \in \Sigma$, $\lambda \in \sigma_s^p(1)$ or $\lambda \in \sigma_e^p(1)$. It is not too hard to see that for a path $p = v_k, v_{k+1}, \ldots, v_n$ with $v_n = t$, σ_s^p and σ_e^p have a matching if and only if there exists a path p' from $v_{k+1} = (i, j)$ to t such that $\sigma_s^{p'}$ and $\sigma_e^{p'}$ have matchings f_s and f_e, and if the *number* of labels both in $\sigma_s^p(i-1) \setminus (\cup_{k \geq i} f_s(k))$ and in $\sigma_s^p(i-1) \setminus (\cup_{k \geq j} f_e(k))$ are strictly positive. We can thus construct a path by moving backwards from t, and storing only a pointer to current source of the path and the numbers of labels that are currently active but have not been assigned to edges between the current source and t. This can clearly be done using only logarithmic space. □

I Proof of Theorem 5

Proof: Membership in NP is immediate. We can simply guess a sequence of eliminations and verify that all of them are valid and that they eventually leave only a single action for each player.

For *hardness* of IDS, recall the construction used in the proof of Theorem 3. Given a B3SAT instance C, we have constructed an MP instance consisting of a layered grid graph $G = (V, E)$ and a labeling function $\sigma : E \to 2^\Sigma$ such that a path between two designated nodes s and t has a matching if and only if C is satisfiable. We will now show that G and σ are induced by iterated dominance in a self-anonymous game Γ with $k = 3$ when only actions 1 and 2 of each player are considered. Observing that, given a matched path from s to t, all players in Γ can also eliminate action 3 at some vertex on the path without affecting the restriction of the labeling function to the remainder of the path effectively reduces B3SAT to IDS.

Given a particular grid graph G, a set Σ of labels, and a labeling function σ as defined in the proof of Theorem 3, we construct a game Γ with players $N = \Sigma$ and actions $A = \{1, 2, 3\}$. Action 1 is associated with east edges of G, action 2 is associated with south edges. Now consider a particular label $i \in \Sigma$. By construction of G, there exist two numbers k_1 and k_2 such that i appears exclusively on east edges (south edges, respectively) that can be reached from s by traversing exactly k_1 or k_2 south edges (east edges). In game Γ, this is modeled by a player that can eliminate action 1 (action 2) after exactly k_1 or k_2 players have eliminated action 2 (action 1). Since we only use payoffs 0 and 1, it follows from Lemma 1 of Conitzer and Sandholm (2005) that an action is dominated by a mixed strategy if and only if it is dominated by a pure strategy. We can thus concentrate exclusively on dominance by pure strategies.

The payoff structure for players of Γ is shown in Figure 9. Clearly, Γ can be constructed from G in polynomial time. In addition to the aforementioned properties regarding the elimination of action 1 or 2, it is easily verified that every player can also eliminate action 3 after k_2 eliminations of action 2 or 1, and that this has no effect whatsoever on the ability of other players to eliminate their actions. In other words, Γ actually induces a three-dimensional grid graph, where each layer in the third dimension is identical to G, and transitions between different layers may take place at vertices where some player has arrived at k_2. This means, however, that a matched path from s to t corresponds to a sequence of eliminations of actions 1 and 2

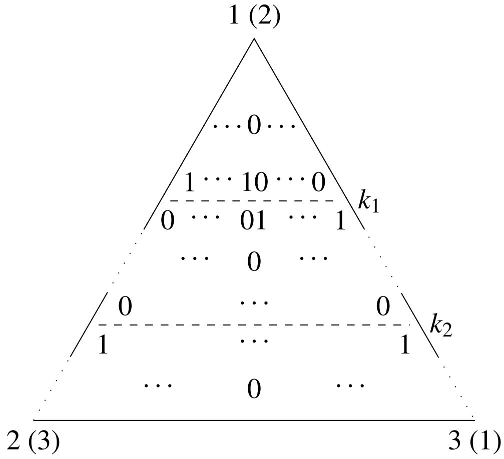

Figure 9: Payoff structure of a particular player of the self-anonymous game Γ used in the proof of Theorem 5. There are two types of players, eliminating action 2 and 1, respectively, actions of the second type are shown in parentheses. The player may eliminate action 2 (1, respectively) by action 3 after exactly k_1 players have eliminated action 1 (2), and by action 1 (2) after exactly k_2 players have eliminated action 1 (2).

in Γ, which can in turn be transformed into a sequence of eliminations that *solves* Γ by iterated dominance by letting each player eliminate action 3 at a certain well-defined point. On the other hand, the possible future transitions within a particular layer of the three-dimension grid graph do not depend on the layer, *i.e.*, a player may not gain the ability to eliminate actions 1 or 2 by first eliminating action 3. Hence, if there is no matched path from s to t, then some player of Γ will not be able to eliminate either action 1 or action 2, meaning that Γ is not solvable by iterated dominance.

Hardness of IDE can be obtained by adding an additional player that can only eliminate once the lowest level of the grid graph has been reached. □