Iterated Weaker-than-Weak Dominance

Shih-Fen Cheng
Singapore Management University
School of Information Systems
80 Stamford Rd
Singapore 178902
sfcheng@smu.edu.sg

Michael P. Wellman
University of Michigan
Computer Science & Engineering
2260 Hayward St
Ann Arbor, MI 48109-2121 USA
wellman@umich.edu

Abstract

We introduce a weakening of standard game-theoretic dominance conditions, called δ-*dominance*, which enables more aggressive pruning of candidate strategies at the cost of solution accuracy. Equilibria of a game obtained by eliminating a δ-dominated strategy are guaranteed to be *approximate* equilibria of the original game, with degree of approximation bounded by the dominance parameter, δ. We can apply elimination of δ-dominated strategies iteratively, but the δ for which a strategy may be eliminated depends on prior eliminations. We discuss implications of this order independence, and propose greedy heuristics for determining a sequence of eliminations to reduce the game as far as possible while keeping down costs. A case study analysis of an empirical 2-player game serves to illustrate the technique, and demonstrate the utility of weaker-than-weak dominance pruning.

1 Introduction

Analysis of games can often be simplified by pruning agents' strategy sets. For instance, a strategy is *strictly dominated* iff there exists a mixture (randomization) over the remaining strategies that achieves strictly higher payoff regardless of the strategies played by other agents. Elimination of strictly dominated strategies is a venerated idea, established at the dawn of game theory as a sound way to remove unworthy strategies from consideration [Gale et al., 1950, Luce and Raiffa, 1957]. In particular, the dominated strategy cannot be part of any Nash equilibrium (NE). Moreover, the elimination conserves solutions in that any NE of the pruned game is also an NE of the original.

Weak dominance relaxes strict dominance by allowing that the dominating strategy achieves payoffs only equally as great. Although weakly dominated strategies may appear in NE of the full game, it remains the case that NE of the pruned game are NE of the original as well. Additional refinements and variants of dominance are possible, for example based on rationalizability, or minimal sets of strategies closed under rational behavior [Benisch et al., 2006].

Elimination of a dominated strategy for one player may enable new dominance relations for others, as it removes cases for which the defining inequality must hold. Therefore, we generally invoke dominance pruning iteratively, until no further reduction is possible. This requires some care in the case of weak dominance, since the set of surviving strategies is *order dependent*, that is, may differ based on the order of eliminations [Gilboa et al., 1990, Myerson, 1991]. Whether a strategy is eliminable through *some* sequence of removals of weakly dominated strategies is a computationally hard problem, in general [Conitzer and Sandholm, 2005]. In contrast, iterated strict dominance is order *in*dependent.

In this paper we investigate a further weakening of weak dominance, which enables more aggressive pruning by allowing that the "dominated" strategy actually be superior to the "dominating" by up to a fixed value δ in some contexts. Such δ-dominated strategies may participate in NE of the original game, and NE of the pruned game are not necessarily NE of the original. However, any NE after pruning does correspond to an *approximate* NE of the original game.

Iterated application of δ-dominance is likewise order dependent. The order of removals dictates not only eliminability, but also the degree of approximation that can be guaranteed for solutions to the reduced game. We explore alternative elimination policies, focusing on greedy elimination based on local assessment of δ.

We illustrate the techniques by applying them to a two-player 27-strategy symmetric game derived from simulation data. Our case study demonstrates the potential utility of the weaker dominance condition, as it reduces the game substantially with little loss in solution accuracy.

2 Preliminaries

A finite normal form game is formally expressed as $[I, \{S_i\}, \{u_i(s)\}]$, where I refers to the set of players and $m = |I|$ is the number of players. S_i is a finite set of pure strategies available to player $i \in I$. Let $S = S_1 \times \cdots \times S_m$ be the space of *joint strategies*. Finally, $u_i : S \to \mathbb{R}$ gives the payoff to player i when players jointly play $s = (s_1, \ldots, s_m)$, with each $s_j \in S_j$.

It is often convenient to refer to a strategy of player i separately from that of the remaining players. To accomodate this, we use s_{-i} to denote the joint strategy of all players other than i.

Let $\Sigma(R)$ be the set of all probability distributions (mixtures) over a given set R. A mixture $\sigma_i \in \Sigma(S_i)$ is called a *mixed strategy* for player i. The payoff $u_i(\sigma)$ of a *mixed strategy profile*, $\sigma \in \Sigma(S)$, is given by the expectation of $u_i(s)$ with respect to σ.

A configuration where all agents play strategies that are best responses to the others constitutes a *Nash equilibrium*.

Definition 1 *A strategy profile σ constitutes a* Nash equilibrium *(NE) of game $[I, \{S_i\}, \{u_i(s)\}]$ iff for every $i \in I$, $\sigma'_i \in \Sigma(S_i)$, $u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma'_i, \sigma_{-i})$.*

We also define an *approximate* version.

Definition 2 *A strategy profile σ constitutes an* ϵ-Nash equilibrium *(ϵ-NE) of game $[I, \{S_i\}, \{u_i(s)\}]$ iff for every $i \in I$, $\sigma'_i \in \Sigma(S_i)$, $u_i(\sigma_i, \sigma_{-i}) + \epsilon \geq u_i(\sigma'_i, \sigma_{-i})$.*

In this study we devote particular attention to games that exhibit symmetry with respect to payoffs.

Definition 3 *A game $[I, \{S_i\}, \{u_i(s)\}]$ is symmetric iff $\forall i, j \in I$, (a) $S_i = S_j$ and (b) $u_i(s_i, s_{-i}) = u_j(s_j, s_{-j})$ whenever $s_i = s_j$ and $s_{-i} = s_{-j}$*

That is, the agents in symmetric games are strategically identical, since all elements of their strategic behavior are the same.

3 δ-Dominance

We start by defining our weaker-than-weak dominance condition.

Definition 4 *Strategy $s_i^d \in S_i$ is δ-dominated iff there exists $\sigma_i^D \in \Sigma(S_i \setminus \{s_i^d\})$ such that:*

$$\delta + u_i(\sigma_i^D, s_{-i}) > u_i(s_i^d, s_{-i}), \forall s_{-i} \in S_{-i}. \quad (1)$$

In other words, s_i^d is δ-dominated if we can find a mixed strategy (on the set of pure strategies excluding s_i^d) that, when compensated by δ, outperforms s_i^d against all pure opponent profiles. Notice that unlike the standard conditions, in considering whether strategy s_i^d is dominated, we must exclude it from the domain of potential dominators. Otherwise, s_i^d would be δ-dominated by itself.

Suppose s_i^d is δ-dominated in game Γ. As noted above, if $\delta > 0$, s_i^d may well appear with positive probability in NE profiles. We may nevertheless choose to eliminate s_i^d, obtaining a new game $\Gamma' = \Gamma \setminus s_i^d$, which is identical to Γ except that $S'_i = S_i \setminus \{s_i^d\}$, and the payoff functions apply only on the reduced joint-strategy space. Although Γ' does not necessarily conserve solutions, we can in fact relate its solutions to approximate solutions of Γ.[1]

Proposition 1 *Let Γ be the original game and let s_i^d be δ-dominated in Γ. If σ is an ϵ-NE in $\Gamma \setminus s_i^d$, then it is a $(\delta+\epsilon)$-NE in Γ.*

Note that with $\epsilon = 0$, the proposition states that exact NE of $\Gamma \setminus s_i^d$ are δ-NE of Γ, where δ is the compensation needed to support dominance.

We may also eliminate δ-dominated strategies in an iterative manner.

[1]The proofs of Proposition 1 and subsequent results are omitted due to space limitations.

Proposition 2 *Let $\Gamma^0, \ldots, \Gamma^n$ be a series of games, with Γ^0 the original game, and $\Gamma^{j+1} = \Gamma^j \setminus t^j$. Further, suppose the eliminated strategy t^j is δ_j-dominated in Γ^j. Then, if σ is an ϵ-NE in Γ^n, it is also a $(\sum_{j=0}^{n-1} \delta_j + \epsilon)$-NE in Γ^0.*

The result follows straightforwardly by induction on Proposition 1.

4 Identifying δ-Dominated Strategies

Definition 4 characterizes the condition for δ-dominance of a single strategy. It is often expedient to eliminate many strategies at once, hence we extend the definition to cover δ-dominance of a subset of strategies.

Definition 5 *The set of strategies $T \subset S_i$ is δ-dominated iff there exists, for each $t \in T$, a mixed strategy $\sigma_i^t \in \Sigma(S_i \setminus T)$ such that:*

$$\delta + u_i(\sigma_i^t, s_{-i}) > u_i(t, s_{-i}), \forall s_{-i} \in S_{-i}. \quad (2)$$

Propositions 1 and 2 can be straightforwardly generalized to eliminations of subsets of strategies for a particular player.

It is well known that standard dominance criteria can be evaluated through linear programming [Myerson, 1991]. The same is true for δ-dominance, and moreover we can employ such programs to identify the minimal δ for which the dominance relation holds. The problem below characterizes the minimum δ such that the set of strategies T is δ-dominated. The problem for dominating a single strategy is a special case.

$$\begin{aligned}
&\min \quad \delta \qquad\qquad (3)\\
&\text{s.t. } \forall\, t \in T\\
&\quad \delta + \sum_{s \in S_i \setminus T} x_t(s) u_i(s, s_{-i}) > u_i(t, s_{-i}),\ \forall\, s_{-i} \in S_{-i}\\
&\quad \sum_{s \in S_i \setminus T} x_t(s) = 1\\
&\quad 0 \leq x_t(s) \leq 1,\ \forall\, s \in S_i \setminus T
\end{aligned}$$

Problem (3) is not quite linear, due to the strict inequality in the first constraint. We can approximate it with a linear constraint by introducing a small predefined constant, τ. The result is the linear program LP-A(S, T).

$$\begin{aligned}
&\min \quad \delta\\
&\text{s.t. } \forall\, t \in T\\
&\quad \delta + \sum_{s \in S_i \setminus T} x_t(s) u_i(s, s_{-i}) \geq u_i(t, s_{-i}),\ \forall s_{-i} \in S_{-i}\\
&\quad \sum_{s \in S_i \setminus T} x_t(s) \leq 1 - \tau\\
&\quad 0 \leq x_t(s) \leq 1,\ \forall\, s \in S_i \setminus T
\end{aligned}$$

5 Controlling Iterated δ-Dominance

By Proposition 1, every time we eliminate a δ-dominated strategy, we add δ to the potential error in solutions to the pruned game. In deciding what to eliminate, we are generally interested in obtaining the greatest reduction in size for the least cost in accuracy. We can pose the problem, for example, as minimizing the total error to achieve a given reduction, or maximizing the reduction subject to a given error tolerance.

In either case, we can view iterated elimination as operating in a state space, where nodes correspond to sets of remaining strategies, and transitions to elimination of one or more strategies. The cost of a transition from node $S = (S_i, S_{-i})$ to $(S_i \setminus T, S_{-i})$ is the δ minimizing LP-A(S, T). We can formulate the overall problem as search from the original strategy space. However, the exponential number of nodes and exponential number of transitions from any given node render any straightforward exhaustive approach infeasible.

As indicated above, the problem is complicated by the order dependence of strategy eliminations. Eliminating a strategy from player i generally expands the set of δ-dominated strategies for the others, though it may shrink its own δ-dominated set. We can formalize this as follows. Let $\delta(t, \Gamma)$ denote the minimum δ such that strategy t is δ-dominated in Γ.[2]

Proposition 3 *Let* $t'_i \in S_i$.

1. $\delta(t, \Gamma \setminus t'_i) \leq \delta(t, \Gamma)$ *for all* $t \in S_j$, $j \neq i$, *and*
2. $\delta(t, \Gamma \setminus t'_i) \geq \delta(t, \Gamma)$ *for all* $t \in S_i \setminus t'_i$.

Because eliminating a strategy may decrease the cost of some future eliminations and increase others, understanding the implications of a pruning operation apparently requires some lookahead.

Our choice at each point is what set of strategies to eliminate, which includes the question of how many to eliminate at one time. For example, suppose $\delta(t_i^1, \Gamma) = \delta_1$, and $\delta(t_i^2, \Gamma \setminus t_i^1) = \delta_2$. In general, it can be shown that $\delta_2 \leq \delta(\{t_i^1, t_i^2\}, \Gamma) \leq \delta_1 + \delta_2$. In many instances, the cost of eliminating both strategies will be far less than the upper bound, which is the value that would be obtained by sequentially eliminating the singletons. However, since the number of candidate elimination sets of size k is exponential in k, we will typically not be able to evaluate the cost of all such candidates. Instead, we investigate heuristic approaches that consider only sets up to a fixed size for elimination in any single iteration step.

5.1 Greedy Elimination Algorithms

We propose iterative elimination algorithms that employ greedy heuristics for selecting strategies to prune for a given player i. Extending these to consider player choice as well is straightforward. The algorithms take as input a starting set of strategies, and an *error budget*, Δ, placing an upper bound on the cumulative error we will tolerate as a result of δ-dominance pruning.

Algorithm 1, GREEDY(S, Δ), computes $\delta(t_i, \Gamma)$ for each $t_i \in S_i$, and eliminates the strategy that is δ-dominated at minimal δ. The algorithm repeats this process one strategy at a time, until such a removal would exceed the cumulative error budget.

[2] Equivalently, $\delta(t, \Gamma)$ is the solution to LP-A$(S, \{t\})$ for S the strategy space of Γ. We also overload the notation to write $\delta(T, \Gamma)$ for the analogous function on strategy sets T.

Algorithm 1 Simple greedy heuristic. At each iteration, the strategy with least δ is pruned.

GREEDY(S, Δ)

$n \leftarrow 1,\ S^n \leftarrow S$
while $\Delta > 0$ **do**
  for $s \in S_i^n$ **do**
    $\delta(s) \leftarrow$ LP-A$(S^n, \{s\})$
  end for
  $t \leftarrow \arg\min_{s \in S_i^n} \delta(s)$
  $d \leftarrow \delta(t)$
  if $\Delta \geq d$ **then**
    $\Delta \leftarrow \Delta - d$
    $S_i^{n+1} \leftarrow S_i^n \setminus \{t\},\ S^{n+1} \leftarrow (S_i^{n+1}, S_{-i})$
    $n \leftarrow n + 1$
  else
    $\Delta \leftarrow 0$
  end if
end while
return S^n

Algorithm 2, GREEDY-K(S, Δ, k), is a simple extension that prunes k strategies in one iteration. We identify the k strategies with least δ *when considered individually*, and group them into a set **K**. We then employ LP-A$(S, \mathbf{K})$ to determine the cost incurred for pruning them at once. Since the set **K** is selected greedily, it will not necessarily be the largest possible set that can be pruned at this cost, nor the minimum-cost set of size k. Nevertheless, we adopt greedy selection to avoid the $\binom{|S_i|}{k}$ optimizations it would take to consider all the candidates.

5.2 Computing Tighter Error Bounds

We can reduce several players' strategy spaces by running Algorithm 2 sequentially. Let Γ be the original game, and let Γ' be the reduced game. Let $\{S_i\}$ and $\{S'_i\}$ be the set of all players' strategy spaces for Γ and Γ' respectively. For each player i, let Δ_i be the accumulated error actually used in GREEDY-K. The total error generated by these reductions, according to Proposition 2, is bounded by $\sum_i \Delta_i$. By taking into account the actual resulting game Γ', however, we can directly compute an error bound that is potentially tighter.

Let $\mathcal{N}$ be the set of all NE in Γ'. The overall error bound is the maximum over $\mathcal{N}$ of the maximal gain available to any player to unilaterally deviating to the original strategy space.

$$\epsilon = \max_{\sigma \in \mathcal{N}} \max_{i \in I} \max_{t \in T_i} \left[u_i(t, \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i}) \right], \qquad (4)$$

where $T_i = S_i \setminus S'_i$. To compute ϵ with (4), we must first find all NE for Γ'. However, computing all NE will generally not be feasible. Therefore, we seek a bound that avoids explicit reference to the set $\mathcal{N}$.

Since σ is an NE in Γ', we have that $u_i(\sigma_i, \sigma_{-i}) \geq u_i(x_i, \sigma_{-i})$, for all $x_i \in \Sigma(S'_i)$. With each $i \in I$, $t \in T_i$, we associate a mixed strategy x_i^t. Replacing σ_i by x_i^t in (4) can only increase the error bound. The resulting expression no longer involves i's equilibrium strategy. We can further relax the bound by replacing maximization wrt equilibrium mix-

Algorithm 2 Generalized greedy heuristic, with k strategies pruned in each iteration.

GREEDY-K(S, Δ, k)

1: $n \leftarrow 1,\ S^n \leftarrow S$
2: **while** $\Delta > 0$ **do**
3:   **for** $s \in S_i^n$ **do**
4:     $\delta(s) \leftarrow$ LP-A$(S^n, \{s\})$
5:   **end for**
6:   $\mathbf{K} \leftarrow \{\}$
7:   **for** $j = 1$ to k **do**
8:     $t_j \leftarrow \arg\min_{s \in S_i^n \setminus \mathbf{K}} \delta(s)$
9:     $\mathbf{K} \leftarrow \{\mathbf{K}, t_j\}$
10:   **end for**
11:   $\delta^{\mathbf{K}} \leftarrow$ LP-A$(S^n, \mathbf{K})$
12:   **if** $\Delta \geq \delta^{\mathbf{K}}$ **then**
13:     $\Delta \leftarrow \Delta - \delta^{\mathbf{K}}$
14:     $S_i^{n+1} \leftarrow S_i^n \setminus \mathbf{K},\ S^{n+1} \leftarrow (S_i^{n+1}, S_{-i})$
15:   **else**
16:     **if** $\Delta \geq t_1$ **then**
17:       $\Delta \leftarrow \Delta - \delta(t_1)$
18:       $S_i^{n+1} \leftarrow S_i^n \setminus \{t_1\},\ S^{n+1} \leftarrow (S_i^{n+1}, S_{-i})$
19:     **else**
20:       $\Delta \leftarrow 0$
21:     **end if**
22:   **end if**
23:   $n \leftarrow n + 1$
24: **end while**
25: **return** S^n

tures σ_{-i} with maximization wrt *any* pure opponent strategies, s_{-i}, yielding

$$\begin{aligned} \bar{\epsilon} &= \max_{i \in I} \max_{t \in T_i} \max_{s_{-i} \in S'_{-i}} \left[u_i(t, s_{-i}) - u_i(x_i^t, s_{-i}) \right] \quad (5) \\ &\geq \max_{\sigma \in \mathcal{N}} \max_{i \in I} \max_{t \in T_i} \left[u_i(t, \sigma_{-i}) - u_i(x_i^t, \sigma_{-i}) \right] \\ &\geq \max_{\sigma \in \mathcal{N}} \max_{i \in I} \max_{t \in T_i} \left[u_i(t, \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i}) \right] = \epsilon. \end{aligned}$$

According to (5), we can bound ϵ by $\bar{\epsilon}$, which does not refer to the set $\mathcal{N}$. We can find $\bar{\epsilon}$ by solving the following optimization problem:

$$\begin{aligned} \min \quad & \bar{\epsilon} \qquad (6) \\ \text{s.t.} \quad & \\ & \bar{\epsilon} \geq u_i(t, s_{-i}) - \sum_{s_i \in S'_i} x_i^t(s_i) u_i(s_i, s_{-i}), \\ & \qquad \forall\, i \in I, t \in T_i, s_{-i} \in S'_{-i} \\ & \sum_{s_i \in S'_i} x_i^t(s_i) = 1, \qquad \forall\, i \in I, t \in T_i \\ & 0 \leq x_i^t(s_i) \leq 1, \qquad \forall\, i \in I, t \in T_i, s_i \in S'_i. \end{aligned}$$

Note that this formulation is very similar to LP-A(S, T), defined in Section 4. The major difference is that LP-A(S, T) is defined for a particular player i, whereas (6) considers all players at once. We employ this bound in experimental evaluation of our greedy heuristics, in Section 6.

5.3 δ-Dominance for Symmetric Games

Thus far, we have emphasized the operation of pruning one or more strategies from a particular player's strategy space. The method of the previous section can improve the bound by considering all players at once. For the special case of symmetric games (Definition 3), we can directly strengthen the pruning operation. Specifically, when we prune a δ-dominated strategy for one player, we can at no additional cost, prune this strategy from the strategy sets of all players.

Proposition 4 *Let Γ be a symmetric game, and suppose strategy s is δ_s-dominated in Γ. Let Γ' be the symmetric game obtained by removing s from* all *players in Γ. If σ is an ϵ-NE in Γ', then it is a $(\delta_s + \epsilon)$-NE in Γ.*

Based on Proposition 4, we can specialize our greedy elimination algorithms for the case of symmetric games. For Algorithm 1, we modify line 10, so that $\{t\}$ is pruned from all players' strategy spaces within the same iteration. For Algorithm 2, we modify lines 14 and 18 analogously.

When a game is symmetric, symmetric equilibria are guaranteed to exist [Nash, 1951]. As Kreps [1990] argues, such equilibria are especially plausible. In our analysis of symmetric games, therefore, we focus on the symmetric NE.

6 Iterative δ-Dominance Elimination: A Case Study

To illustrate the use of δ-dominance pruning, we apply the method to a particular game of interest. On this example, we evaluate the greedy heuristics in terms of the tradeoff between reduction and accuracy. We also compare the theoretical bounds to actual approximation errors observed in the reduced games.

6.1 The TAC$\downarrow_2$ Game

The subject of our experiment is a 2-player symmetric game, based on the Trading Agent Competition (TAC) travel-shopping game [Wellman et al., 2003]. TAC Travel is actually an 8-player symmetric game, where agents interact through markets to buy and sell travel goods serving their clients. TAC$\downarrow_2$ is derivative from TAC Travel in several respects:

- TAC travel is a dynamic game with severely incomplete and imperfect information, and highly-dimensional infinite strategy sets. TAC$\downarrow_2$ restricts agents to a discrete set of strategies, all parametrized versions of the University of Michigan agent, Walverine [Wellman et al., 2005b]. The restricted game is thus representable in normal form.
- Payoffs for TAC$\downarrow_2$ are determined empirically through Monte Carlo simulation.
- The game is reduced to two players by constraining groups of four agents each to play the same strategy. This can be viewed as assigning a leader for each group to select a strategy for all to play. The game among leaders is in this case a 2-player game. The transformation from TAC$\downarrow_8$ to TAC$\downarrow_2$ is an example of the hierarchical reduction technique proposed by Wellman

et al. [2005a] for approximating games with many players. Note that this form of reduction is orthogonal to the reduction achieved by eliminating strategies through dominance analysis.

Although TAC$\downarrow_2$ is a highly simplified version of the actual TAC game, Wellman et al. [2005b] argue that analyzing such approximations can be useful, in particular for focusing on a limited set of candidate strategies to play in the actual game. Toward that end, dominance pruning can play a complementary role to other methods of analysis.

The actual instance of TAC$\downarrow_2$ we investigate comprises 27 strategies (378 distinct strategy profiles) for which sufficient samples (at least 20 per profile) were collected to estimate payoffs.

6.2 Comparison of Greedy Heuristics

Since the game is symmetric, we eliminate strategies from all players at once rather than one at a time (see Section 5.3). Starting with our 27-strategy TAC$\downarrow_2$ game, we first apply iterative elimination of strictly dominated strategies. This prunes nine strategies, leaving us with an 18-strategy game that cannot be further reduced without incurring potential approximation error. We then applied both GREEDY-1 and GREEDY-2, each with a budget of $\Delta = 200$. Figure 1 plots, for each algorithm, the cumulative error cost incurred to reach the associated number of remaining strategies. Each elimination operation takes a step down (by the number of strategies pruned) and across (by the error δ added to the total). Note that the first large step down at cost zero corresponds to the initial pruning by strict dominance.

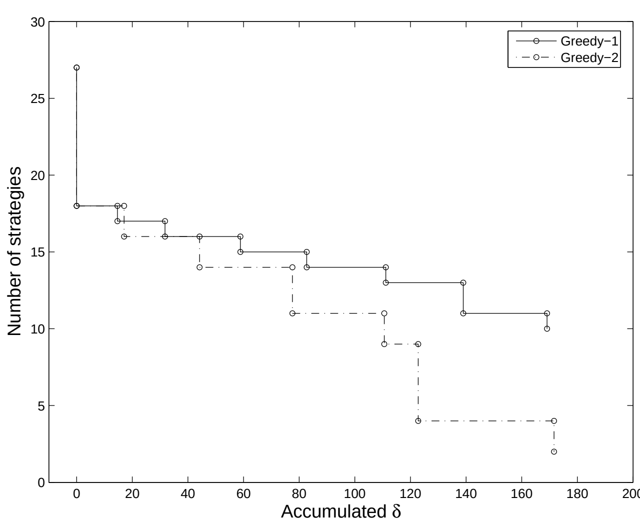

Figure 1: Number of strategies versus accumulated δ, for GREEDY-1 and GREEDY-2.

As the graph apparently indicates, GREEDY-2 reaches any particular reduction level at a cost less than or equal to GREEDY-1. With the error tolerance $\Delta = 200$, GREEDY-1 prunes the game down to ten strategies, whereas GREEDY-2 takes us all the way down to two. However, we must decouple two factors behind the difference in measured results. First, the algorithms may prune strategies in a different order. Second, the algorithm GREEDY-K computes the bound for each iteration taking into account all k strategies pruned at once.

In this instance, in fact the sequence of eliminations is quite similar. The first four strategies eliminated by GREEDY-1 and GREEDY-2 are the same, and the next four are the same except for a one-pair order swap. Thus, we can attribute the difference in apparent cumulative error after eight removals (169 versus 78) entirely to the distinction in how they tally error bounds. In general, the elimination orders can differ almost arbitrarily, though we might expect them typically to be similar. In another instance of TAC$\downarrow_2$ (based on an earlier snapshot of the database with 26 strategies), we also observed that the first eight δ-eliminations differed only in a one-pair swap. We have not to date undertaken an empirical study of the comparison.

A more accurate assessment of the cost of iterated elimination can be obtained by computing the tighter bounds described in Section 5.2, or directly assessing the error. Figure 2 presents the data from Figure 1 (axes inverted), along with the more precise error measurements.

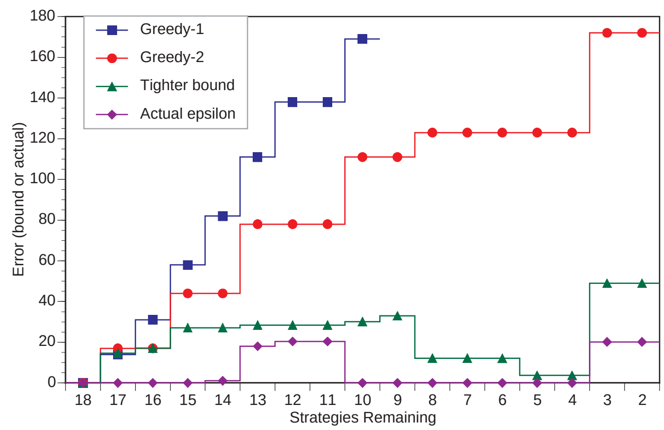

Figure 2: Error bounds derived from GREEDY-1 and GREEDY-2, compared to tighter bound estimates as well as actual errors.

Tighter bounds reported in Figure 2 are those derived from the linear program (6), applied to the respective strategy sets. We use the remaining strategy sets based on GREEDY-1 down to 10 remaining, and the sets for GREEDY-2 thereafter. We also determined the actual error for a given reduced strategy set, by computing all symmetric NE of the reduced game (using GAMBIT [McKelvey and McLennan, 1996]),[3] and for each finding the best deviation to eliminated strategies. The maximum of all these is the error for that strategy set.

As we can see from the figure, the cumulative error bounds reported by the algorithms (based on Proposition 2) are quite

[3]In some instances, GAMBIT was unable to solve our reduced games in reasonable time due to numerical difficulties. In these cases, we tried small random (symmetry-preserving) perturbations of the game until we were able to solve one for all NE. The errors reported are with respect to the solutions we found, which thus tend to overstate the error due to elimination because they include an additional source of noise.

conservative. In all cases, after a few eliminations the tighter bounds are far more accurate. The actual errors are in many cases quite small (often zero). That is, in at least this (real) example game, we can aggressively prune weaker-than-weakly dominated strategies and then still have games where all solutions are near equilibria of the original game.[4]

6.3 Loss of Equilibria

The preceding analysis considers the accuracy of solutions to the reduced game with respect to the original. We may also be concerned about losing solutions to the original that may include δ-dominated strategies. To examine this issue, we track the 21 symmetric NE found for the instance of $\text{TAC}{\downarrow}_2$ analyzed above, which has 18 strategies after eliminating those strictly dominated.[5] Figure 3 shows how many of these original NE survive after successive rounds of eliminating a δ-dominated strategy, using the GREEDY-1 algorithm. As seen in the figure, all solutions survive the first three eliminations, and two still remain after the eight iterations of GREEDY-1.

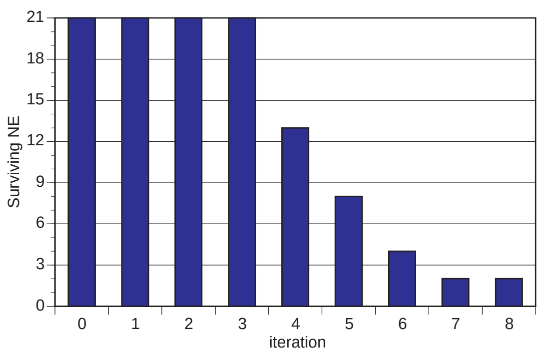

Figure 3: Original NE surviving after successive iterations of δ-dominated strategy elimination.

For situations where the purpose of analysis is to characterize all or most (approximate) equilibria, eliminating δ-dominated strategies sacrifices potentially desired coverage. If the objective, in contrast, is to identify samples (i.e., particular examples) of relatively stable profiles, this loss of equilibria is not a paramount concern.

7 Conclusion

Eliminating strategies that are only nearly dominated enables significantly more aggressive pruning than standard dominance, while introducing a controllable amount of solution error. Our δ-dominance concept represents such a relaxation, and we exhibit bounds on the degree of approximation for solutions of the reduced game with respect to the original, for individual or iterated eliminations of single strategies or strategy sets. Results are generally order dependent, however greedy selection techniques may work well in practice. The bounds for iterated elimination are quite conservative, and can be tightened by retrospective analysis of the actual set of strategies eliminated.

A case study applying iterated elimination of δ-dominated strategies to an empirical game illustrates the approach. The exercise demonstrates the possibility of identifying a much smaller subgame with solutions that are excellent approximations wrt the original. Further work should evaluate the methods more broadly over a range of games.

[4] Again, we have not to date performed a comprehensive empirical study. The other instance of $\text{TAC}{\downarrow}_2$ mentioned above exhibited similar results.

[5] Of course if we knew these equilibria, we would not eliminate strategies for purposes of simplifying equilibrium computation. Our point here and in the preceding section is to analyze the effect of elimination using the known solutions to measure error.

Acknowledgments

We thank Kevin Lochner and Daniel Reeves for assistance with the $\text{TAC}{\downarrow}_2$ analysis. This work was supported in part by grant IIS-0414710 from the National Science Foundation.

References

Michael Benisch, George Davis, and Tuomas Sandholm. Algorithms for rationalizability and CURB sets. In *Twenty-First National Conference on Artificial Intelligence*, pages 598–604, Boston, 2006.

Vincent Conitzer and Tuomas Sandholm. Complexity of (iterated) dominance. In *Sixth ACM Conference on Electronic Commerce*, pages 88–97, Vancouver, 2005.

D. Gale, H. W. Kuhn, and A. W. Tucker. Reductions of game matrices. In *Contributions to the Theory of Games*, volume 1, pages 89–96. Princeton University Press, 1950.

I. Gilboa, E. Kalai, and E. Zemel. On the order of eliminating dominated strategies. *Operations Research Letters*, 9:85–89, 1990.

David M. Kreps. *Game Theory and Economic Modelling*. Oxford University Press, 1990.

R. D. Luce and H. Raiffa. *Games and Decisions*. Wiley, New York, 1957.

Richard D. McKelvey and Andrew McLennan. Computation of equilibria in finite games. In *Handbook of Computational Economics*, volume 1. Elsevier, 1996.

Roger B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, 1991.

John Nash. Non-cooperative games. *Annals of Mathematics*, 54:286–295, 1951.

Michael P. Wellman, Amy Greenwald, Peter Stone, and Peter R. Wurman. The 2001 trading agent competition. *Electronic Markets*, 13:4–12, 2003.

Michael P. Wellman, Daniel M. Reeves, Kevin M. Lochner, Shih-Fen Cheng, and Rahul Suri. Approximate strategic reasoning through hierarchical reduction of large symmetric games. In *Twentieth National Conference on Artificial Intelligence*, pages 502–508, Pittsburgh, 2005a.

Michael P. Wellman, Daniel M. Reeves, Kevin M. Lochner, and Rahul Suri. Searching for Walverine 2005. In *IJCAI-05 Workshop on Trading Agent Design and Analysis*, Edinburgh, 2005b.