Algorithms for Stochastic Games with Perfect Monitoring*

Dilip Abreu Benjamin Brooks Yuliy Sannikov

January 30, 2020

Abstract

We study the pure-strategy subgame-perfect Nash equilibria of stochastic games with perfect monitoring, geometric discounting, and public randomization. We develop novel algorithms for computing equilibrium payoffs, in which we combine policy iteration when incentive constraints are slack with value iteration when incentive constraints bind. We also provide software implementations of our algorithms. Preliminary simulations indicate that they are significantly more efficient than existing methods. The theoretical results that underlie the algorithms also imply bounds on the computational complexity of equilibrium payoffs when there are two players. When there are more than two players, we show by example that the number of extreme equilibrium payoffs may be countably infinite.

Keywords: Stochastic game, perfect monitoring, algorithm, computation.
JEL classification: C63, C72, C73, D90.

*Abreu: Department of Economics, New York University, dilip.abreu@nyu.edu; Brooks: Department of Economics, University of Chicago, babrooks@uchicago.edu; Sannikov: Graduate School of Business, Stanford University, sannikov@gmail.com. This work has benefited from the comments of numerous seminar audiences. We have also benefited from the superb research assistance of Mathieu Cloutier, Moshe Katzwer, and Kai Hao Yang. We are very grateful to Joel Sobel for his guidance and to several anonymous referees for their valuable input. Finally, we would like to acknowledge financial support from the National Science Foundation Grant #1530823.

1 Introduction

This paper develops new algorithms for computing equilibrium payoffs in stochastic games. Specifically, we study the payoffs that can be attained in pure-strategy subgame-perfect Nash equilibria of repeated games with perfect monitoring, public randomization, and a stochastically evolving state variable. The current state determines which actions are feasible for the players as well as the payoffs of those actions. The chosen actions in turn influence the future evolution of the state. This classical structure is used to model a wide variety of phenomena in economics and in other disciplines. The range of applications include: dynamic oligopoly with investment (in, e.g., capacity, research and development, advertising), risk sharing, and the dynamics of political bargaining and compromise (cf. Ericson and Pakes, 1995; Kocherlakota, 1996; Dixit, Grossman, and Gul, 2000).

The standard methodology for computing subgame-perfect equilibrium payoffs in repeated games comes from Abreu, Pearce, and Stacchetti (1986, 1990), hereafter APS. They showed that the set of equilibrium payoffs satisfies a recursive relationship that is analogous to the Bellman equation from dynamic programming. In particular, any equilibrium payoff can be decomposed into a flow payoff from the first period of play plus the expected discounted payoff from the next period onward, which, by subgame perfection, is also an equilibrium payoff. Just as the value function is the fixed point of the Bellman operator, so too the equilibrium payoff set is the largest fixed point of an operator that produces the set of payoffs which can be generated using continuation values chosen from a given set. Moreover, APS show that iterating this operator on any set that contains all equilibrium payoffs yields a sequence of sets that asymptotically converges to the set of equilibrium payoffs. Although APS wrote explicitly about games with imperfect monitoring and without a state variable, their results mechanically extend to the class of games studied here, where payoffs are generated in each state using continuation payoffs drawn from a received payoff correspondence.[1]

Our main contribution is a refinement of the APS algorithm. In the analogy with dynamic programming, the APS algorithm is identified with value function iteration. We combine this approach with a form of policy iteration, which is used to partially solve out equilibrium payoffs when incentive constraints are slack. The resulting hybrid algorithm converges faster than existing methods, and the hybrid operator that is used in this algorithm is of bounded computational complexity when there are two players. The approach also leads to new structural insights about equilibria that generate extreme equilibrium payoffs, namely that

[1]For early extensions involving a state variable see Atkeson (1991) and Phelan and Stacchetti (2001). A more recent application is Hörner et al. (2011). For a more complete description of the self-generation methodology for stochastic games, see Mailath and Samuelson (2006).

play must be stationary until the first period in which an incentive constraint binds.

The approach has several novel elements, some of which apply even to repeated games, while others are tailored to the stochastic setting. To motivate a defining element of our refinement, consider an infinitely repeated Prisoners' Dilemma. We normalize the Nash payoffs to zero and the payoff from mutual cooperation to 1. Suppose that the discount factor δ is such that static Nash is the only action profile that can be supported, even when all feasible payoffs can be promised as continuation values. A fortiori, we can conclude that the equilibrium payoff set is $\{(0,0)\}$. Nonetheless, this fact will only be discovered by the APS operator asymptotically: at iteration k, the payoff (δ^k, δ^k) will still be in the APS approximation. For even though mutual cooperation is not used to generate new payoffs, it is still implicitly being played k periods in the future through the received set of continuation payoffs. There is, in a sense, an internal inconsistency in the way that the APS operator is generating new payoffs in this example: It is claimed that static Nash generates an equilibrium payoff that maximizes the sum of the players' payoffs, and that this sum is strictly positive. But the sum of payoffs in the first period is zero, meaning that the sum of the continuation payoffs must be even higher than the best payoff we can generate! This is obviously not sustainable in equilibrium.

This example illustrates a more general principle in repeated games: Fix a set of welfare weights, i.e., a direction in payoff space, and consider the equilibrium with the highest payoffs in this direction. The action profile played in the first period of this equilibrium must have flow payoffs that are weakly above the highest equilibrium payoffs. Indeed, this principle extends to stochastic games: Fix a direction in welfare space. For each state, there is a highest equilibrium payoff in this direction, which is generated by playing some action profile in the first period, followed by continuation equilibrium payoffs in every continuation state. The continuation payoffs are bounded by the highest equilibrium payoff in their respective states, and sometimes further if required by incentive compatibility. As a result, the highest equilibrium payoff in a given state is below a *recursive level*, obtained by playing the optimal action profile for one period with the highest equilibrium payoffs as continuation payoffs.

This principle motivates our refinement. For a given direction, consider a policy of action profiles meant to attain the highest payoffs in each state. The highest payoffs in this direction are attained by recursively continuing this policy, with the highest equilibrium continuation values, until incentive compatibility requires some value burning. This happens when the best incentive compatible payoff, i.e., the level generated by the APS operator, is lower than that attained by following the policy. This leads to the following *max-min-max* problem to

bound the level of payoffs:

$$\text{max over actions of } \Big(\text{ min of } \big(\text{the "recursive" level and max level generated by APS}\big)\Big).$$

The max-min-max operator maps a payoff correspondence into the correspondence of payoffs that are below the max-min-max level in all directions. Our first main result shows that iterative application of this operator can be used to compute equilibrium payoffs.

Because our operator uses levels that are weakly below those of APS, it generates smaller correspondences than APS. As a result, the sequence it generates converges weakly faster than the APS sequence. Indeed, there are even examples, such as the aforementioned Prisoners' Dilemma, where the APS sequence only converges asymptotically, but the max-min-max sequence converges after finitely many rounds (although this is not generally the case). That our operator is theoretically novel and cuts more sharply is clear, but perhaps its main advantage is that *it is significantly easier to compute*. This derives from additional theoretical structure that the operator embodies, as we now explain.

First, as long as the max-min-max sequence decreases at the first iteration in the sense of set containment,[2] our operator has the following crucial property: it relies on the maximum level generated by APS only when an incentive constraint binds for some player. Thus, while our operator nominally requires us to know the APS levels, in fact it is sufficient to know the maximal level attained with payoffs in which an incentive constraint binds—a considerably easier task.

In addition, when computing the max-min-max levels, we maximize over a tuple of action profiles, and we minimize over a tuple of what we call *regimes*, which indicate for each state whether the minimum level is APS or recursive. We refer to the action profiles and regimes collectively as a *policy*. We show that for each direction, there exists a policy that optimizes payoffs simultaneously in all states. For any direction, the optimal policy can be found through a form of policy iteration. The policy is optimal if and only if for every state there is no alternative action profile which, if substituted in, would increase the level, and there is no regime substitution that would lower the level. Once we have found the optimal policy for one direction, it is easy to compute a range of directions for which it remains optimal, and also the improving substitution when the optimum changes.

When there are two players, these properties yield an especially powerful implementation. We find the optimal policy for a starting direction. After that, we move clockwise. By considering one-state substitutions, we endogenously identify directions where the optimal

[2]There are many initial correspondences that guarantee this will happen. Two examples are the feasible payoff correspondence and a correspondence that in every state is equal to a large hypercube that contains all of the flow payoffs for all states. The latter is what we use in our numerical simulations.

policy changes and update the optimal policy. After a full revolution, we have bounded payoffs in all directions. Moreover, for two players, each action profile can generate at most four extreme binding payoffs, generalizing the result of Abreu and Sannikov (2014) for repeated games. This leads to a bound on both the complexity of our operator and of the equilibrium payoff correspondence.

We have implemented this algorithm as a software package that is freely available through the online supplement and an author's website.[3] We report a number of numerical examples, including a risk-sharing game à la Kocherlakota (1996).

When there are more than two players, we show by example that the number of extreme equilibrium payoffs may be countably infinite. As a result, exact computation of equilibrium payoffs may be impossible. We therefore propose a more flexible operator that bounds payoffs with the max-min-max level for a subset of directions, which is dynamically updated between applications. We show that binding payoffs will remain sufficient to determine the APS level as long as any legacy directions we drop are not needed to determine the binding payoffs or the local frontier around them. There are no restrictions on how directions can be added. For any such sequence of direction sets, the algorithm is guaranteed to converge to a correspondence that contains all equilibrium payoffs. There are many ways to use this characterization, and we focus on one simple implementation: At every round, we drop some directions that are redundant for computing binding payoffs. If the number of directions is below a fixed bound, we randomly add new directions that correspond to "faces" of the exact max-min-max correspondence. These directions are computed via a generalization of the two-player direction rotation procedure.

In Online Appendix B, we present simulations where the equilibrium payoff correspondence has a finite number of faces that are successfully discovered by the algorithm, so that the sequence of approximations converges exactly to the equilibrium payoff correspondence. We also solve a three-player risk-sharing game to show a new result of independent economic interest, which is that formal insurance contracts between a subset of the players can lead to lower payoffs for all players.

All of the aforementioned algorithms converge to equilibrium payoffs from the outside, thus providing upper bounds. As a last topic, we show how our methodology can be adapted produce a lower bound. In particular, we construct an algorithm that necessarily converges, after finitely many steps, to a correspondence that *strictly self-generates in every direction*, thus robustly certifying that payoffs in this correspondence can be attained in equilibrium.

A notable antecedent of our work is Abreu and Sannikov (2014) who studied repeated games with two players, perfect monitoring, and public randomization. They proposed a

[3] www.benjaminbrooks.net/software.shtml

distinct refinement of the APS algorithm, and our procedure does not reduce to theirs when there is a single state. As mentioned above, they give a bound on the number of extreme equilibrium payoffs, which is tighter than our bound (due to the specialization to repeated games) but is based on the same geometry.

This paper supersedes our earlier work (Abreu, Brooks, and Sannikov, 2016), in which we studied the same class of games but restricted to two players. Based on similar ideas, we proposed a related but distinct algorithm, which also proceeds by iteratively modifying a payoff tuple, one state at a time, to obtain a sequence of payoffs and corresponding bounds. In contrast to the present work, that operator did not have bounded computational complexity in the two-player case and did not apply to many-player games.

Another key reference is Judd, Yeltekin, and Conklin (2003), hereafter JYC, who proposed the approximation of the APS operator by bounding payoffs in a fixed and finite set of directions. While they wrote about repeated games, their methodology readily generalizes to the class of stochastic games we consider.[4] Key differences between the approaches are that we use the max-min-max operator rather than the APS operator, we endogenize the directions in order to identify faces, and we compute our operator exactly when there are two players. We report simulations computed with our algorithm and that of JYC. Our algorithm converges significantly faster and to a sharper approximation. Note that our approach makes heavy use of perfect monitoring, whereas JYC's methodology is easily adapted to games with imperfect public monitoring.

There are other lines of work on computing equilibria without public randomization (Berg and Kitti, 2019) or with mixed strategies (Berg, 2019). There is also a large body of work computing Markov perfect equilibria (Pakes and McGuire, 1994). Our methodology can be used to bound payoffs in Markov equilibria, but it cannot be used to compute just the set of Markov equilibrium payoffs. Renner and Scheidegger (2018) propose a machine learning algorithm for approximating feasible payoffs in dynamic principal-agent problems, whose efficacy is suggested by simulations. In contrast, our algorithms pertain to stochastic games, and we show analytically that our procedure is guaranteed to converge to the equilibrium payoff correspondence.

Finally, our methods exploit the linear structure of equilibria in ways that are evocative of linear programming. This is not altogether unexpected, given the well-known connection between linear programming and dynamic programming. The formal connection is explored in Supplemental Appendix D. Our conclusion is that while there are deep connections, there are also fundamental differences because the minimization over regimes makes the

[4]Such an extension is done by Yeltekin, Cai, and Judd (2017). A related approach is taken by Sleet and Yeltekin (2016), using what they call block correspondences instead of bounding in fixed directions.

max-min-max program non-convex, so it does not simply reduce to a linear program.

The rest of this paper is organized as follows. Section 2 describes the basic model and background material. Section 3 presents our algorithm and its key properties. Section 4 studies implementation and examples when there are two players. Section 5 extends the analysis to many players. Section 6 discusses lower bounds on payoffs, and Section 7 is a conclusion. Additional material, referenced throughout, is contained in an Online Appendix and a Supplemental Appendix.[5]

2 Setting and background

Players $i = 1, \ldots, N$ interact over infinitely many periods. There is a finite set of states S. If the current state is $s \in S$, player i takes an action a_i in a finite set $\mathbf{A}_i(s)$.[6] The set of action profiles in state s is $\mathbf{A}(s) = \times_{i=1}^{N} \mathbf{A}_i(s)$. Player i's flow utility from action profile $a \in \mathbf{A}(s)$ is $g_i(a|s)$. The resulting probability that the next state is s' is $\pi(s'|a, s)$. We will henceforth assume that each a is available in a single state and write $g_i(a)$ and $\pi(s'|a)$.[7] Players discount future payoffs at the common rate $\delta \in (0, 1)$. Actions and the state are perfectly observable.

We will study the equilibrium payoff correspondence $\mathbf{V} : S \to 2^{\mathbb{R}^N}$, where $\mathbf{V}(s)$ is the set of expected discounted payoffs that can be achieved in some pure-strategy subgame-perfect Nash equilibrium with public randomization, when the initial state of the world is s. For a formal definition of an equilibrium in this setting, see Mailath and Samuelson (2006, Sections 5.5 and 5.7), in particular Corollary 5.7.1.[8]

Subgame-perfection implies that any equilibrium payoff can be decomposed into the discount-weighted sum of a flow payoff which is obtained in the first period and expected continuation equilibrium payoffs from the second period onwards. The technique of APS is to generalize this recursive relationship in a manner that is analogous to how the Bellman operator generalizes the recursive characterization of the value function in dynamic programming. Explicitly, fix a compact-valued payoff correspondence $\mathbf{W} : S \to 2^{\mathbb{R}^N}$. Note that the assumption of compactness of $\mathbf{W}$ is maintained throughout. The associated *threat tuple* $\underline{\mathbf{w}}(\mathbf{W})$ is

$$\underline{\mathbf{w}}_i(\mathbf{W})(s) = \min \left\{ w_i | (w_i, w_{-i}) \in \mathbf{W}(s) \text{ for some } w_{-i} \in \mathbb{R}^{N-1} \right\}.$$

[5]The Supplemental Appendix is available at http://benjaminbrooks.net/.

[6]In general, we will use boldface to denote functions whose domain is the state space.

[7]This is without loss, since we could simply redefine an action to be the ordered pair (a_i, s).

[8]Strictly speaking, the definition of an equilibrium in Mailath and Samuelson (2006) differs slightly from the one which we are implicitly using. They assume that there is a probability distribution over the initial state, while we define equilibrium payoffs *conditional* on the initial state.

For an action profile $a \in \mathbf{A}(s)$, let

$$\underline{u}_i(a, \mathbf{W}) = \max_{a_i'} \left[(1-\delta) g_i\left(a_i', a_{-i}\right) + \delta \sum_{s' \in S} \pi\left(s' | a_i', a_{-i}\right) \underline{\mathbf{w}}_i(\mathbf{W})(s') \right].$$

We say that v is *generated in state s by the action profile $a \in \mathbf{A}(s)$ and the correspondence* $\mathbf{W}$ if there exist $\mathbf{w} \in \mathbf{W}$—meaning that $\mathbf{w}(\cdot)$ is a selection from $\mathbf{W}(\cdot)$—such that

$$v = (1-\delta) g(a) + \delta \sum_{s' \in S} \pi(s'|a) \mathbf{w}(s'); \tag{1}$$

$$(1-\delta) g_i(a) + \delta \sum_{s' \in S} \pi(s'|a) \mathbf{w}_i(s') \geq \underline{u}_i(a, \mathbf{W}) \quad \forall\, i = 1, \ldots, N. \tag{2}$$

This equation implicitly assumes that a deviation from a by player i will be punished by a transition to the worst continuation equilibrium for the deviator, which results in a payoff of $\underline{\mathbf{w}}_i(\mathbf{W})(s')$ if the next state is s'. This is without loss due to perfect monitoring.

Let $B\left(a, \mathbf{W}\right)$ denote the set of payoffs that are generated by $a \in \mathbf{A}(s)$ and $\mathbf{W}$, and let $B\left(\mathbf{W}\right)(s) = \text{co}\left(\cup_{a \in \mathbf{A}(s)} B\left(a, \mathbf{W}\right)\right)$, where co denotes the convex hull. It is a fact that B is increasing in $\mathbf{W}$ and maps compact-valued correspondences to compact-valued correspondences. We say that $\mathbf{W}$ is *self-generating* if $\mathbf{W} \subseteq B(\mathbf{W})$, i.e., $\mathbf{W}(s) \subseteq B(\mathbf{W})(s)$ for all $s \in S$. APS's arguments, extended to stochastic games, show that if $\mathbf{W}$ is bounded and self-generating, then $B(\mathbf{W}) \subseteq \mathbf{V}$. These properties imply that $\mathbf{V}$ is the largest bounded self-generating payoff correspondence.[9] Moreover, the following algorithm can be used to compute $\mathbf{V}$: Let $\mathbf{W}^0$ be any correspondence that contains $\mathbf{V}$, and generate the sequence $\mathbf{W}^k = B\left(\mathbf{W}^{k-1}\right)$ for $k \geq 1$. Then $\cap_{k \geq 0} \mathbf{W}^k = \mathbf{V}$. Moreover, if $B(\mathbf{W}^0) \subseteq \mathbf{W}^0$, then the sequence is decreasing: $\mathbf{W}^k \subseteq \mathbf{W}^{k-1}$ for all $k > 0$.

3 A refinement of the algorithm of APS

We now describe our refinement of the APS algorithm. This algorithm will similarly generate a sequence of payoff correspondences via iterative application of a new operator $\widetilde{B}$. This algorithm converges faster, as the operator $\widetilde{B}$ generates smaller correspondences and is significantly easier to compute than the APS operator B. Note that Supplemental Appendix C contains pseudocode for the algorithms developed over the next three sections.

[9] Note that a pure-strategy subgame-perfect equilibrium need not exist. Subgame-perfection requires that the continuation equilibrium is an equilibrium after every history. Thus, an equilibrium exists in some state if and only if an equilibrium exists in every state. As a result, $\mathbf{V}$ is either empty in all states or non-empty in all states.

3.1 Our operator

Preliminary to defining $\widetilde{B}$, it is useful to reformulate the APS operator. Let $\Lambda = \left\{\lambda \in \mathbb{R}^N | \, \|\lambda\| = 1\right\}$ denote the set of N-dimensional *directions*, endowed with the subspace topology. For each $\lambda \in \Lambda$, $s \in S$, and $a \in \mathbf{A}(s)$, we define

$$x^{APS}(a, \lambda, \mathbf{W}) = \max\{\lambda \cdot v | v \in B(a, \mathbf{W})\},$$

where our convention is that the max of an empty set is $-\infty$. In addition, we say that an action profile a is *supportable (at* $\mathbf{W}$*)* if $B(a, \mathbf{W}) \neq \emptyset$, so that $x^{APS}(a, \lambda, \mathbf{W})$ is finite for all λ. Let $\mathbf{A}(\mathbf{W})(s)$ be the set of supportable action profiles in state s. We further define $x^{APS}(s, \lambda, \mathbf{W}) = \max_{a \in \mathbf{A}(\mathbf{W})(s)} x^{APS}(a, \lambda, \mathbf{W})$ to be the maximum level in the direction λ that is attained by the APS operator, which again is $-\infty$ if there are no supportable actions in state s. Then

$$B(\mathbf{W})(s) = \{v | \lambda \cdot v \leq x^{APS}(s, \lambda, \mathbf{W}) \ \forall \lambda \in \Lambda\}.$$

Note that $B(\mathbf{W})(s)$ is empty if there are no supportable action profiles in state s. In addition, as per Footnote 9, if $\mathbf{W}$ is empty in some state, then there are no continuation value profiles, no action can be supported, and $B(\mathbf{W})$ is empty in every state.

Our operator will be defined similarly in terms of bounding hyperplanes, but with tighter bounds than x^{APS}. To motivate the bounds, let us briefly consider which payoffs would be attainable in the absence of incentive constraints. In other words, what are the *feasible payoffs* that can be generated using some strategy profile? In a repeated game, the answer is simply the convex hull of the flow payoffs. In a stochastic game, things are more complicated because of how action profiles influence the evolution of the state. For any fixed welfare weights $\lambda \in \Lambda$, however, the problem of maximizing the λ-weighted sum of expected discounted payoffs is simply a Markov decision problem. Blackwell (1965) showed that there is a stationary solution, i.e., a selection of action profiles $\mathbf{a} \in \mathbf{A}$ such that an optimal strategy is to play $\mathbf{a}(s)$ whenever the state is s. Equivalently, the optimal strategy involves playing an optimal action for one period, followed by recursively starting the optimal strategy over in the next period. This strategy simultaneously attains the highest levels in all states. The associated optimal levels $x(s)$ are the unique solution to

$$x(s) = (1 - \delta)\lambda \cdot g(\mathbf{a}(s)) + \delta \sum_{s' \in S} \pi(s' | \mathbf{a}(s)) x(s') \tag{3}$$

for all $s \in S$.

The situation is more complicated with incentive constraints, because it may not be possible to attain the levels in (3) without giving some player an incentive to deviate. In particular, it may be necessary to burn some surplus in some states in the direction λ in order to give sufficiently large continuation values to deter deviations. Exactly how much surplus needs to be burnt depends on the threat point and the shape of the frontier, which are things we do not know until we actually compute $\mathbf{V}$. We can, however, use an approximation $\mathbf{W}$ that contains $\mathbf{V}$ to bound the amount of value burning required to enforce any action profile $\mathbf{a}(s)$. At the same time, as we discussed in the introduction, there are cases where the APS bound is also too generous, because of spuriously large continuation values in $\mathbf{W}$.

These considerations motivate the hybrid approach that we now adopt, which is to use the recursive methodology of Markov decision problems in some states and APS-style bounds in other states. Holding fixed the actions played in the first period, we will select the configuration of recursive or APS for each state in order to minimize the bound on payoffs, thus ensuring that our approximation is not too generous.

To that end, let us define a *policy* to be a pair $(\mathbf{a}, \mathbf{r})$, where $\mathbf{a} \in \mathbf{A}(\mathbf{W})$,[10] and $\mathbf{r} : S \to \{R, APS\}$ is a *regime* (where the R stands for *recursive*). Let $\mathbf{R}$ denote the set of regimes. For given λ and $\mathbf{W}$, consider the system

$$\mathbf{y}(s) = \begin{cases} (1-\delta)\lambda \cdot g(\mathbf{a}(s)) + \delta \sum_{s' \in S} \pi(s'|\mathbf{a}(s))\mathbf{y}(s') & \text{if } \mathbf{r}(s) = R; \\ x^{APS}(\mathbf{a}(s), \lambda, \mathbf{W}) & \text{if } \mathbf{r}(s) = APS \end{cases} \tag{4}$$

for all $s \in S$. Standard arguments can be used to show (4) has a unique solution: given $\mathbf{y} : S \to \mathbb{R}$, let $T(\mathbf{y}, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W})$ be the tuple defined by

$$T(\mathbf{y}, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W})(s) = \begin{cases} (1-\delta)\lambda \cdot g(\mathbf{a}(s)) + \delta \sum_{s' \in S} \pi(s'|\mathbf{a}(s))\mathbf{y}(s') & \text{if } \mathbf{r}(s) = R; \\ x^{APS}(\mathbf{a}(s), \lambda, \mathbf{W}) & \text{if } \mathbf{r}(s) = APS. \end{cases}$$

Clearly, any $\mathbf{y}$ that satisfies (4) must be a fixed point of the operator $T(\cdot, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W})$. Moreover, this operator is a contraction of modulus δ in $\mathbf{y}$, so that a fixed point exists and is unique. We denote it by $x(s, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W})$. We record some properties of T for future reference.

Lemma 1. *Fix λ, $\mathbf{W}$, $\mathbf{a} \in \mathbf{A}(\mathbf{W})$, and $\mathbf{r}$. As a function of $\mathbf{y} : S \to \mathbb{R}$, T is*

(L1.i) increasing;

(L1.ii) a contraction with modulus δ and hence has a unique fixed point $\mathbf{y}^*$;

[10] Our focus is primarily on computing which payoffs can be generated, taking as a given which action profiles are supportable. The computation of $\mathbf{A}(\mathbf{W})$ is a straightforward by-product of other necessary calculations. See a discussion in Section 4.2.3.

(L1.iii) *if* $T(\mathbf{y}) \leq (\geq)\mathbf{y}$ *then* $\mathbf{y}^* \leq (\geq)T(\mathbf{y})$.

Proof of Lemma 1.

(L1.i) This is immediate from positive linearity of T in $\mathbf{y}$ when $\mathbf{r}(s) = R$ and the fact that it is independent of $\mathbf{y}$ when $\mathbf{r}(s) = APS$.

(L1.ii) Let $\|\cdot\|$ denote the sup norm. Then

$$\begin{aligned}\|T(\mathbf{y}, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}) - T(\mathbf{y}', \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W})\| &= \delta \max_{\{s|\mathbf{r}(s)=R\}} \sum_{s' \in S} \pi(s'|\mathbf{a}(s))|\mathbf{y}(s') - \mathbf{y}'(s')| \\ &\leq \delta \|\mathbf{y} - \mathbf{y}'\|\end{aligned}$$

as desired. The rest of the result follows from the Banach fixed point theorem.

(L1.iii) If $T(\mathbf{y}, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}) \leq (\geq)\mathbf{y}$ then from (L1.i), we conclude that the sequence $\mathbf{y}^k$ generated by iterative application of T starting with $\mathbf{y}^0 = \mathbf{y}$ is monotonically decreasing (increasing) and, by (L1.ii), must converge to the unique fixed point $\mathbf{y}^*$. The result then follows.

□

The next key definition mirrors that of x^{APS}:

$$x(s, \lambda, \mathbf{W}) = \max_{\mathbf{a} \in \mathbf{A}(\mathbf{W})} \min_{\mathbf{r} \in \mathbf{R}} x(s, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}). \tag{5}$$

Finally, the operator $\widetilde{B}$ is defined according to

$$\widetilde{B}(\mathbf{W})(s) = \{v | \lambda \cdot v \leq x(s, \lambda, \mathbf{W}) \ \forall \lambda \in \Lambda\}.$$

We refer to this as the *max-min-max operator*, since for each direction, we maximize over action tuples, minimize over regimes, and maximize over APS payoffs. Because of the minimization over regimes, $\widetilde{B}$ generates smaller correspondences than B. Furthermore, the simultaneous determination of levels in states for which the R regime is specified collapses multiple rounds of the APS operator into a single step.

We now verify that the operator $\widetilde{B}$ satisfies all of the critical properties of the APS operator, so that it can in fact be used to compute $\mathbf{V}$:

Theorem 1 (The max-min-max operator)**.** $\widetilde{B}$ *has the following properties:*

(T1.i) $\widetilde{B}$ *is increasing in* $\mathbf{W}$*, and if* $\mathbf{W}$ *is compact, then* $\widetilde{B}(\mathbf{W})$ *is compact;*

(T1.ii) $\widetilde{B}(\mathbf{W}) \subseteq B(\mathbf{W})$, *and if* $\mathbf{W} \subseteq \widetilde{B}(\mathbf{W})$, *then* $\widetilde{B}(\mathbf{W}) \subseteq \mathbf{V}$;

(T1.iii) $\mathbf{V} = \widetilde{B}(\mathbf{V})$;

(T1.iv) *Fix a correspondence* $\widetilde{\mathbf{W}}^0$ *that contains* $\mathbf{V}$. *Define the sequence* $\left\{\widetilde{\mathbf{W}}^k\right\}_{k=0}^{\infty}$ *by* $\widetilde{\mathbf{W}}^k = \widetilde{B}\left(\widetilde{\mathbf{W}}^{k-1}\right)$. *Then* $\mathbf{V} = \cap_{k=0}^{\infty}\widetilde{\mathbf{W}}^k$.

Remark 1. Recall that it is possible that no pure-strategy equilibria exist, and $\mathbf{V}(s) = \emptyset$ for all s. If this happens, then since the $\widetilde{\mathbf{W}}^k$ correspondences are closed and decreasing, there must be some k at which $\widetilde{\mathbf{W}}^k(s) = \emptyset$ for some s. In the next iteration, there will be no supportable action profiles in any state, and $\widetilde{\mathbf{W}}^{k+1}$ will be empty in every state, at which point the algorithm converges. Theorem 1 and our subsequent results encompass the case where payoff correspondences are empty and no payoffs can be generated, and our proofs remain correct with the convention that the maximum of an empty set is $-\infty$. The explicit focus is, however, on the non-trivial case where $\mathbf{W}$ is non-empty valued.

Proof of Theorem 1.

(T1.i) It is clear that $x^{APS}(a, \lambda, \mathbf{W})$ is increasing in $\mathbf{W}$ for every $a \in \mathbf{A}(\mathbf{W})(s)$ and λ. Hence, if $\mathbf{W}' \subseteq \mathbf{W}$, then for all $(\mathbf{y}, \lambda, \mathbf{a}, \mathbf{r})$, where $\mathbf{a} \in \mathbf{A}(\mathbf{W})$, $T(\mathbf{y}, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}) \geq T(\mathbf{y}, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}')$, which immediately implies that the fixed point of $T(\cdot, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W})$ is greater than the fixed point of $T(\cdot, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}')$. Thus, $x(s, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W})$ is increasing in $\mathbf{W}$. As a result, $\min_{\mathbf{r}\in\mathbf{R}} x(s, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W})$, and $x(s, \lambda, \mathbf{W})$ are also increasing in $\mathbf{W}$. If $\mathbf{W}$ is compact, then $x^{APS}(a, \lambda, \mathbf{W})$ is bounded above for every λ. Thus, $\widetilde{B}(\mathbf{W})(s)$ is bounded and closed, being the intersection of closed half-spaces.

(T1.ii) Clearly, $x(s, \lambda, \mathbf{W}) \leq x^{APS}(s, \lambda, \mathbf{W})$, which implies that $\widetilde{B}$ is always contained in B. Thus, if $\mathbf{W} \subseteq \widetilde{B}(\mathbf{W})$, then $\mathbf{W} \subseteq B(\mathbf{W})$ and hence, by APS, $B(\mathbf{W}) \subseteq \mathbf{V}$. Consequently, $\widetilde{B}(\mathbf{W}) \subseteq \mathbf{V}$.

(T1.iii) From (T1.ii), it suffices to show that $\mathbf{V} \subseteq \widetilde{B}(\mathbf{V})$, i.e., for all λ, $x(s, \lambda, \mathbf{V}) \geq x^{APS}(s, \lambda, \mathbf{V})$. To that end, fix λ, and for all s, let $\mathbf{a}(s)$ be an action that maximizes $x^{APS}(a, \lambda, \mathbf{V})$ and let $\mathbf{w}(s')$ be the associated continuation values as a function of the next-period state s'. We will show that $\min_{\mathbf{r}\in\mathbf{R}} x(s, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{V}) \geq x^{APS}(s, \lambda, \mathbf{V})$, so that $x(s, \lambda, \mathbf{V}) \geq x^{APS}(s, \lambda, \mathbf{V})$, which implies the result. Since $\mathbf{V} = B(\mathbf{V})$, $x^{APS}(s, \lambda, \mathbf{V}) \geq \lambda \cdot u$ for all $u \in \mathbf{V}(s')$ for all s'. Since $\mathbf{w}(s') \in \mathbf{V}(s')$ for all s',

$$\begin{aligned} x^{APS}(s, \lambda, \mathbf{V}) &= (1-\delta)\lambda \cdot g(\mathbf{a}(s)) + \delta \sum_{s'\in S} \pi(s'|\mathbf{a}(s)\lambda \cdot \mathbf{w}(s') \\ &\leq (1-\delta)\lambda \cdot g(\mathbf{a}(s)) + \delta \sum_{s'\in S} \pi(s'|\mathbf{a}(s)x^{APS}(s', \lambda, \mathbf{V}). \end{aligned}$$

Thus, if we let $\mathbf{y}(s) = x^{APS}(s, \lambda, \mathbf{V})$ for all s, then for *any* regimes $\mathbf{r}$, $T(\mathbf{y}, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{V}) \geq \mathbf{y}$ (with equality if $\mathbf{r}(s) = APS$ and weak inequality if $\mathbf{r}(s) = R$). By (L1.iii), we conclude that $\mathbf{y}(s) = x^{APS}(s, \lambda, \mathbf{V}) \leq x(s, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{V}) = \mathbf{y}^*(s)$, as required.

(T1.iv) (T1.ii) implies that $\widetilde{\mathbf{W}}^k \subseteq \mathbf{W}^k$, where the latter is the kth element of the APS sequence starting from $\widetilde{\mathbf{W}}^0$. Also, the fact that $\widetilde{\mathbf{W}}^0$ contains $\mathbf{V}$, (T1.i), and (T1.iii) imply that $\mathbf{V} \subseteq \widetilde{\mathbf{W}}^k$. Thus, $\mathbf{V} \subseteq \cap_k \widetilde{\mathbf{W}}^k \subseteq \cap_k \mathbf{W}^k = \mathbf{V}$.

□

Remark 2. Roughly, our definition of a policy $(\mathbf{a}, \mathbf{r})$ allows us to treat separately states in which incentive constraints are slack from those in which they bind. When incentive constraints are slack, optimal behavior is stationary until a constraint binds, payoffs are defined recursively, and R is the optimal regime. When incentive constraints bind, value burning is required to provide incentives and the minimal regime is necessarily APS. At the fixed point (where $\mathbf{W} = \mathbf{V}$) and at the corresponding optimal policy in the direction λ, this description is exactly correct: the recursive regimes are the most permissive and yield upper bounds on attainable levels. This fundamental observation underlies the proof of (T1.iii). However, the algorithm approaches the fixed point from the "outside," i.e., $\mathbf{V} \subseteq \widetilde{\mathbf{W}}^k$, and it is possible that recursion along the path of the algorithm yields lower levels than the corresponding APS levels, since the latter may rely on spuriously generous continuation payoffs. This motivates our requirement that the regimes are chosen to minimize levels. In the next section, we show that this minimization reduces to simply setting the regime to R in states for which this yields a level below that of APS. As we are dealing with a simultaneous equation system, this test is modestly more complicated than it might appear at first.

Remark 3. $\widetilde{B}(\mathbf{W})(s)$ is the intersection of hyperplanes where the levels are given by $x(s, \cdot, \mathbf{W})$. While the former is necessarily a convex set, the latter is generally *not* a convex function, so that $x(s, \cdot, \mathbf{W})$ need not be the support function of $\widetilde{B}(\mathbf{W})(s)$. An example in which this happens is presented in Section 4.3.1. Thus, one must be careful in applying intuition from convex geometry to $\widetilde{B}$ and the sets $\widetilde{\mathbf{W}}^k$.

Remark 4. It is a straightforward consequence of (T1.iv) that the sequence $\widetilde{\mathbf{W}}^k$ converges to $\mathbf{V}$ in the Hausdorff metric. In practice, we terminate the algorithm when the distance between successive iterates falls below some threshold (which it must eventually, since the sequence is Cauchy). The distance between iterates has no simple relationship with the distance to $\mathbf{V}$, and there is no guarantee that the final correspondence is close to the fixed point. In Section 6, we adapt our algorithm to produce a sub-correspondence of $\mathbf{V}$, which can be used to bound the error in the approximation.

The remainder of this section develops further properties of $\widetilde{B}$ that make it especially tractable for computation, namely, the *state independence of the optimal policy* and the *sufficiency of binding payoffs* when the APS level is minimal.

3.2 State-independence of the optimal policy

The definition of $x(s, \lambda, \mathbf{W})$ in (5) leaves open the possibility that the optimal policy in the direction λ depends on s. We now show that there exists a policy that is simultaneously optimal for all states, and we lay the foundations for a simple algorithm to compute it.

3.2.1 Minimal regimes

For notational economy, we shall temporarily suppress the dependence of x and other objects on $\mathbf{W}$, and simply write $x(s, \lambda)$, etc. For $a \in \mathbf{A}(s)$ define

$$x^R(a, \lambda, \mathbf{a}, \mathbf{r}) = (1-\delta)\lambda \cdot g(a) + \delta \sum\nolimits_{s'} \pi(s'|a) x(s', \lambda, \mathbf{a}, \mathbf{r}).$$

Given λ and $\mathbf{a}$, we say that the regime $\mathbf{r}$ is *minimal* if for all $s \in S$, $x(s, \lambda, \mathbf{a}, \mathbf{r}) = \min_{\mathbf{r}' \in \mathbf{R}} x(s, \lambda, \mathbf{a}, \mathbf{r}')$. In other words, they minimize the level in all states simultaneously. In addition, given $\mathbf{r}$, let $\mathbf{r} \setminus s$ be the regime that is the same as $\mathbf{r}$ in every state except s, i.e., we flip the regime in state s.

We now show that there exist minimal regimes. Moreover, there is a simple set of inequalities that characterize when regimes are minimal. These inequalities will be central to our implementations in Sections 4 and 5.

Lemma 2 (Minimal regimes)**.** *For all* $\mathbf{a} \in \mathbf{A}(\mathbf{W})$ *and* λ*,*

(L2.i) *there exist minimal regimes;*

(L2.ii) $\mathbf{r}$ *is minimal if and only if for all* $s \in S$*,*

$$x(s, \lambda, \mathbf{a}, \mathbf{r}) = \min\left\{x^{APS}(\mathbf{a}(s), \lambda), x^R(\mathbf{a}(s), \lambda, \mathbf{a}, \mathbf{r})\right\}; \tag{6}$$

(L2.iii) *if* (6) *is violated for some* s*, then* $\mathbf{r}$ *is not minimal. Moreover, for all* $s' \in S$*,* $x(s', \lambda, \mathbf{a}, \mathbf{r} \setminus s) \leq x(s', \lambda, \mathbf{a}, \mathbf{r})$*, with strict inequality in state* s*.*

Proof of Lemma 2. These results follow directly from Lemma 1:

(L2.iii) If (6) is violated at s, then letting $\mathbf{y} = x(\lambda, \mathbf{a}, \mathbf{r})$, we conclude that $T(\mathbf{y}, \lambda, \mathbf{a}, \mathbf{r} \setminus s) \leq \mathbf{y}$. By (L1.iii), $x(\lambda, \mathbf{a}, \mathbf{r} \setminus s) = \mathbf{y}^* \leq T(\mathbf{y}, \lambda, \mathbf{a}, \mathbf{r} \setminus s) \leq \mathbf{y} = x(\lambda, \mathbf{a}, \mathbf{r})$, where the penultimate inequality is strict by assumption in state s.

(L2.ii) Only if follows from (L2.iii). For the if direction, suppose that $\mathbf{r}$ satisfies (6) for all s. Then for any $\mathbf{r}' \in \mathbf{R}$, it follows that $\mathbf{y} \leq T(\mathbf{y}, \lambda, \mathbf{a}, \mathbf{r}')$, where $\mathbf{y} = x(\lambda, \mathbf{a}, \mathbf{r})$. Then by (L1.iii), $\mathbf{y}^* = x(\lambda, \mathbf{a}, \mathbf{r}') \geq x(\lambda, \mathbf{a}, \mathbf{r})$, so that $\mathbf{r}$ is indeed minimal.

(L2.i) Let $\mathbf{r}$ solve $\min_{\mathbf{r}' \in \mathbf{R}} \sum_{s \in S} x(s, \lambda, \mathbf{a}, \mathbf{r})$. We argue that $\mathbf{r}$ is minimal. Suppose not. Then by (L2.ii), (6) is violated at some $s \in S$, and by (L2.iii), $\sum_{s' \in S} x(s', \lambda, \mathbf{a}, \mathbf{r} \setminus s) < \sum_{s' \in S} x(s', \lambda, \mathbf{a}, \mathbf{r})$, a contradiction.

□

3.2.2 Maximal actions

We now extend these results to actions: as long as there are supportable action profiles in every state, there exists a selection of action profiles that maximizes the level for all states simultaneously, and maximal actions are characterized by a simple set of inequalities.

We will prove this result using an operator that is analogous to T but directly imposes minimality. For $\mathbf{y} : S \to \mathbb{R}$ let

$$T^{min}(\mathbf{y}, \lambda, \mathbf{a})(s) = \min \left\{ x^{APS}(\mathbf{a}(s), \lambda), (1-\delta)\lambda \cdot g(\mathbf{a}(s)) + \delta \sum_{s' \in S} \pi(s'|\mathbf{a}(s))\mathbf{y}(s') \right\}.$$

Lemma 3. *Fix λ and $\mathbf{a} \in \mathbf{A}(\mathbf{W})$. As a function of $\mathbf{y} : S \to \mathbb{R}$, T^{min} is*

(L3.i) increasing;

(L3.ii) a contraction with modulus δ, and hence has a unique fixed point $\mathbf{y}^*$;

(L3.iii) if $T^{min}(\mathbf{y}) \leq (\geq) \mathbf{y}$ then $\mathbf{y}^* \leq (\geq) T^{min}(\mathbf{y})$;

Proof of Lemma 3. The proof is identical to that of Lemma 1, replacing T with $T^{\min}$. □

Now, let us define $x(s, \lambda, \mathbf{a}) = \min_{\mathbf{r} \in \mathbf{R}} x(s, \lambda, \mathbf{a}, \mathbf{r})$ to be the minimal levels associated with the action tuple $\mathbf{a} \in \mathbf{A}(\mathbf{W})$. The definition of $x(s, \lambda)$ in (5) implies that $x(s, \lambda) = \max_{\mathbf{a} \in \mathbf{A}(\mathbf{W})} x(s, \lambda, \mathbf{a})$. For a given λ, we say that $\mathbf{a}$ is *maximal* if for all $s \in S$, $x(s, \lambda) = x(s, \lambda, \mathbf{a})$, i.e., $\mathbf{a}$ attains the max-min-max level in all states simultaneously. Also define

$$x^R(a, \lambda, \mathbf{a}) = (1-\delta)\lambda \cdot g(a) + \delta \sum_{s' \in S} \pi(s'|a) x(s', \lambda, \mathbf{a}).$$

Finally, for $\mathbf{a} \in \mathbf{A}$, $s \in S$, and $a \in \mathbf{A}(s)$, let us define the substituted action tuple $\mathbf{a} \setminus (s, a) \in \mathbf{A}$ to be the action a in state s and $\mathbf{a}(s')$ in each state $s' \neq s$.

Lemma 4 (Maximal actions)**.** *Suppose that $\mathbf{A}(\mathbf{W})$ is non-empty valued. Then for all λ,*

(L4.i) *there exist maximal actions;*

(L4.ii) $\mathbf{a} \in \mathbf{A}(\mathbf{W})$ *is maximal if and only if for all* $s \in S$ *and* $a \in \mathbf{A}(\mathbf{W})(s)$,

$$x(s, \lambda, \mathbf{a}) \geq \min \left\{ x^{APS}(a, \lambda), x^{R}(a, \lambda, \mathbf{a}) \right\}, \tag{7}$$

(L4.iii) *if* (7) *is violated for some* $s \in S$ *and* $a \in \mathbf{A}(\mathbf{W})(s)$*, then* $\mathbf{a}$ *is not maximal. Moreover, for all* $s' \in S$*,* $x(s', \lambda, \mathbf{a} \setminus (s, a)) \geq x(s', \lambda, \mathbf{a})$*, with strict inequality in state* s*.*

Proof of Lemma 4. The proof mirrors that of Lemma 2, and follows directly from Lemma 3:

(L4.iii) If (7) is violated at (s, a), then letting $\mathbf{y} = x(\lambda, \mathbf{a})$, we conclude that $T^{min}(\mathbf{y}, \lambda, \mathbf{a} \setminus (s, a)) \geq \mathbf{y}$. By (L3.iii), $x(\lambda, \mathbf{a} \setminus (s, a)) = \mathbf{y}^* \geq T^{min}(\mathbf{y}, \lambda, \mathbf{a} \setminus (s, a)) \geq \mathbf{y} = x(\lambda, \mathbf{a})$, where the penultimate inequality is strict by assumption, in state s.

(L4.ii) Only if follows from (L4.iii). For the if direction, suppose that $\mathbf{a}$ satisfies (7) for all s and $a \in \mathbf{A}(\mathbf{W})(s)$. Then for any $\mathbf{a}' \in \mathbf{A}(\mathbf{W})$ it follows that $\mathbf{y} \geq T^{min}(\mathbf{y}, \lambda, \mathbf{a}')$ where $\mathbf{y} = x(\lambda, \mathbf{a})$. Then by (L3.iii), $\mathbf{y}^* = x(\lambda, \mathbf{a}') \leq x(\lambda, \mathbf{a})$, so that $\mathbf{a}$ is indeed maximal.

(L4.i) Let $\mathbf{a}$ solve $\max_{\mathbf{a}' \in \mathbf{A}(\mathbf{W})} \sum_{s \in S} x(s, \lambda, \mathbf{a}')$. We argue that $\mathbf{a}$ is maximal. Suppose not. Then by (L4.ii), (7) is violated at some $s \in S$ and $a \in \mathbf{A}(\mathbf{W})(s)$ and by (L4.iii) $\sum_{s' \in S} x(s', \lambda, \mathbf{a} \setminus (s, a)) > \sum_{s' \in S} x(s', \lambda, \mathbf{a})$, a contradiction.

□

To summarize, there exits a state-independent optimal policy. This is extremely useful for computation, since it means we can solve for optimal levels in all states simultaneously.

3.3 The sufficiency of binding payoffs

In the discussion around Theorem 1, we identified the APS regime with a situation in which incentive constraints bind, so that value burning is required to deter deviations. There is nothing in the definition of x^{APS}, however, that requires that incentive constraints bind, and we have left open the possibility that the APS regime is minimal even though incentive constraints are slack. While this may happen, it does not occur in equilibrium or along the sequence $\widetilde{\mathbf{W}}^k$, as long as $B(\widetilde{\mathbf{W}}^0) \subseteq \widetilde{\mathbf{W}}^0$. This fact is immensely useful for computation, since it means that we do not need to compute optimal APS levels in all directions (which would amount to computing the APS operator itself). Instead, we can just compute the optimal APS level when incentive constraints bind.

We now develop this result formally. Let us reintroduce the payoff correspondence $\mathbf{W}$ as an argument in optimal levels. For any $s \in S$ and $a \in \mathbf{A}(\mathbf{W})(s)$, define

$$\widehat{x}^{APS}(a, \lambda, \mathbf{W}) = \max\{\lambda \cdot w \mid w \in B(a, \mathbf{W}) \text{ and } \exists\, i \text{ s.t. (2) holds as equality}\}. \quad (8)$$

We refer to the difference

$$\gamma(a, \lambda, \mathbf{W}) = x^{APS}(a, \lambda, \mathbf{W}) - \widehat{x}^{APS}(a, \lambda, \mathbf{W})$$

as the *APS gap*.[11] We shall see that for any λ and $\mathbf{a} \in \mathbf{A}(\mathbf{W})$, if $\gamma(\mathbf{a}(s), \lambda, \mathbf{W})$ is strictly positive in state s, then R is a minimal regime in state s. On the other hand, if $\gamma(\mathbf{a}(s), \lambda, \mathbf{W})$ is zero, then by definition we can restrict attention to APS payoffs for which at least one player's incentive constraint binds.

To prove this result, we need two intermediate lemmas. We say that the APS operator B *sub-generates at* $\mathbf{W}$ *in the direction* λ if for all $s \in S$, $x^{APS}(s, \lambda, \mathbf{W}) \leq \max\{\lambda \cdot w | w \in \mathbf{W}(s)\}$, and B *sub-generates at* $\mathbf{W}$ if $B(\mathbf{W}) \subseteq \mathbf{W}$. These notions extend to $\widetilde{B}$ in the obvious way.

Lemma 5. *Suppose that B sub-generates at $\mathbf{W}$ in the direction λ. Then for any $\mathbf{a} \in \mathbf{A}(\mathbf{W})$ and s, if $\gamma(\mathbf{a}(s), \lambda, \mathbf{W}) > 0$, then*

$$x^{APS}(\mathbf{a}(s), \lambda, \mathbf{W}) \geq x^{R}(\mathbf{a}(s), \lambda, \mathbf{a}, \mathbf{W}) = x(s, \lambda, \mathbf{a}, \mathbf{W}). \quad (9)$$

Moreover, there exist minimal regimes such that $\mathbf{r}(s) = R$ for all s with $\gamma(\mathbf{a}(s), \lambda, \mathbf{W}) > 0$.

Proof of Lemma 5. Suppose that $\gamma(\mathbf{a}(s), \lambda, \mathbf{W}) > 0$. Then any maximal continuation values in $\mathbf{W}$ in the direction λ, denoted $\mathbf{w}$, must be incentive compatible for $\mathbf{a}(s)$, and

$$x^{APS}(\mathbf{a}(s), \lambda, \mathbf{W}) = (1-\delta)\lambda \cdot g(\mathbf{a}(s)) + \delta \sum_{s' \in S} \pi(s'|\mathbf{a}(s))\, \lambda \cdot \mathbf{w}(s').$$

Sub-generation and the definition of x imply that for all s', $\lambda \cdot \mathbf{w}(s') \geq x^{APS}(\mathbf{a}(s'), \lambda, \mathbf{W}) \geq x(s', \lambda, \mathbf{W})$. Hence,

$$\begin{aligned} x^{APS}(\mathbf{a}(s), \lambda, \mathbf{W}) &\geq (1-\delta)\lambda \cdot g(\mathbf{a}(s)) + \delta \sum_{s' \in S} \pi(s'|\mathbf{a}(s))\, x(s', \lambda, \mathbf{W}) \\ &\geq (1-\delta)\lambda \cdot g(\mathbf{a}(s)) + \delta \sum_{s' \in S} \pi(s'|\mathbf{a}(s))\, x(s', \lambda, \mathbf{a}, \mathbf{W}) = x^{R}(\mathbf{a}(s), \lambda, \mathbf{a}, \mathbf{W}) \end{aligned}$$

[11] Note that if a is supportable, then there must exist a payoff where (2) holds as an equality, so that $\widehat{x}^{APS}$ is finite. This follows from the definition of $\underline{u}_i(a, \mathbf{W})$, which is at least the payoff obtained by playing a with the worst continuation values in $\mathbf{W}$. Thus, it cannot be that $\underline{u}_i(a, \mathbf{W})$ is strictly below every payoff that satisfies (1) for some $\mathbf{w} \in \mathbf{W}$.

as desired.

Finally, suppose $\mathbf{r}$ is minimal and $\gamma(\mathbf{a}(s), \lambda, \mathbf{W}) > 0$. If $x^{APS}(\mathbf{a}(s), \lambda, \mathbf{W}) > x^R(\mathbf{a}(s), \lambda, \mathbf{a}, \mathbf{W})$, then $\mathbf{r}(s) = R$. Otherwise, (9) implies that $x^{APS}(\mathbf{a}(s), \lambda, \mathbf{W}) = x^R(\mathbf{a}(s), \lambda, \mathbf{a}, \mathbf{W})$. Thus, if we set $\mathbf{r}'(s) = R$ for all states with $\gamma(\mathbf{a}(s), \lambda, \mathbf{W}) > 0$ and $\mathbf{r}'(s') = \mathbf{r}(s')$ otherwise, then $x(\cdot, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W})$ is a fixed point of $T(\cdot, \lambda, \mathbf{a}, \mathbf{r}', \mathbf{W})$, so that $\mathbf{r}'$ satisfies (6) and is minimal. □

Remark 5. For the remainder of our analysis, we work with payoff correspondences at which B sub-generates. We shall therefore without loss restrict attention to minimal regimes that satisfy the selection of Lemma 5, i.e., ones for which $\mathbf{r}(s) = R$ whenever the APS gap is positive. This allows us to avoid the computation of non-binding APS payoffs, which would basically entail computing all of $B(\mathbf{W})$, whereas the sign of the APS gap is automatically computed in the process of finding the optimal binding APS payoffs. This is discussed further in Section 4.2.3.

Note that monotonicity of B implies that if B sub-generates at $\mathbf{W}$, then it will sub-generate at $B(\mathbf{W})$ as well. It does not follow that B will sub-generate at other sub-correspondences of $\mathbf{W}$. However:

Lemma 6. *If $\widetilde{B}$ sub-generates at $\mathbf{W}$, then B sub-generates at $\widetilde{B}(\mathbf{W})$.*

Proof of Lemma 6. Towards a contradiction, suppose that some action profile $a \in \mathbf{A}(\mathbf{W})(s)$, with continuation values $\mathbf{w} \in \widetilde{B}(\mathbf{W})$, generates a payoff outside of $\widetilde{B}(\mathbf{W})$. Then for some λ,

$$\begin{aligned} x(s, \lambda, \mathbf{W}) < x^{APS}(a, \lambda, \widetilde{B}(\mathbf{W})) &= \lambda \cdot \left((1-\delta) g(a) + \delta \sum_{s' \in S} \pi(s'|a) \mathbf{w}(s') \right) \\ &\leq (1-\delta) \lambda \cdot g(a) + \delta \sum_{s' \in S} \pi(s'|a) x(s', \lambda, \mathbf{W}), \end{aligned}$$

where the last inequality holds because $\lambda \cdot \mathbf{w}(s') \leq x(s', \lambda, \mathbf{W})$, since $\mathbf{w}(s') \in \widetilde{B}(\mathbf{W})(s')$. The right-hand side of this inequality equals $x^R(a, \lambda, \mathbf{a}, \mathbf{W})$ for any $\mathbf{a} \in \mathbf{A}(\mathbf{W})$ that is maximal in the direction λ (given $\mathbf{W}$). Since $\widetilde{B}(\mathbf{W}) \subseteq \mathbf{W}$, we know that $x^{APS}(s, \lambda, \mathbf{W}) > x(s, \lambda, \mathbf{W})$ as well. That is, $x(s, \lambda, \mathbf{a}, \mathbf{W}) < \min\{x^{APS}(a, \lambda, \mathbf{W}), x^R(a, \lambda, \mathbf{a}, \mathbf{W})\}$, contradicting (L4.ii). □

This leads to the following result about the sequence generated by $\widetilde{B}$:

Proposition 1 (Sufficiency of binding payoffs)**.** *Let $\widetilde{\mathbf{W}}^k$ be the sequence from (T1.iv). Suppose B sub-generates at $\widetilde{\mathbf{W}}^0$. Then for any $k \geq 0$, B sub-generates at $\widetilde{\mathbf{W}}^k$. Hence, for any λ and $\mathbf{a} \in \mathbf{A}(\widetilde{\mathbf{W}}^k)$, there exist minimal regimes $\mathbf{r}$ such that if $\gamma(\mathbf{a}(s), \lambda, \widetilde{\mathbf{W}}^k) > 0$, then $\mathbf{r}(s) = R$.*

Proof of Proposition 1. By assumption, $B(\widetilde{\mathbf{W}}^0) \subseteq \widetilde{\mathbf{W}}^0$. Hence, $\widetilde{B}(\widetilde{\mathbf{W}}^0) \subseteq \widetilde{\mathbf{W}}^0$. Since $\widetilde{B}$ is increasing (T1.i), it follows that $\widetilde{B}(\widetilde{\mathbf{W}}^k) \subseteq \widetilde{\mathbf{W}}^k$ for all $k \geq 0$. By Lemma 6, $B(\widetilde{\mathbf{W}}^k) \subseteq \widetilde{\mathbf{W}}^k$ for all $k \geq 0$. The second part of the proposition then follows from Lemma 5. □

3.4 Computing $x(s, \lambda, \mathbf{W})$

In the next sections, we use the characterization of optimal policies and the sufficiency of binding payoffs to construct simple algorithms for computing $\widetilde{B}$. These algorithms depend on a subroutine that computes $x(\lambda, \mathbf{W})$. We have already referenced pieces of this routine, but we now synthesize these results into a unified algorithm, together with various simplifications that are possible when B sub-generates at $\mathbf{W}$.

The computation of $x(\lambda, \mathbf{W})$ consists of an outer routine, in which we maximize over $\mathbf{a}$, and an inner routine, where we minimize over $\mathbf{r}$. For the inner routine (Algorithm 1 in Supplemental Appendix C), we use (L2.ii), which says that the regimes $\mathbf{r}$ are not minimal if and only if (6) is violated in some state s, and (L2.iii), which says that $\mathbf{r} \setminus s$ has lower levels in all states. This suggests a simple iterative procedure: Starting from any $\mathbf{r}$, check for violations of (6). If there are none, then $\mathbf{r}$ is minimal. Otherwise, replace $\mathbf{r}$ with $\mathbf{r} \setminus s$, where s is associated with a violation of (6), and continue. By (L2.iii), The levels decrease at each substitution, so the regimes must converge after at most $|\mathbf{R}|$ substitutions.[12]

This routine nominally requires us to compute $x^{APS}(a, \lambda, \mathbf{W})$ to check (6). But under the hypothesis that B sub-generates at $\mathbf{W}$, we can use Lemma 5 to simplify this process: In any state where $\gamma(\mathbf{a}(s), \lambda, \mathbf{W}) > 0$, the minimal regime can be taken to be recursive. Otherwise, by definition $x^{APS} = \widehat{x}^{APS}$, so we can use binding payoffs in (6).

The outer routine (Algorithm 2) is analogous, with actions instead of regimes: Starting from some $\mathbf{a}$, check for violations of (7). If there are none, then by (L4.ii), $\mathbf{a}$ is maximal. Otherwise, replace $\mathbf{a}$ with $\mathbf{a} \setminus (s, a)$ where (s, a) is associated with a violation, compute new minimal regimes, and continue. By (L4.iii), the levels increase at every substitution, so the actions converge in at most $|\mathbf{A}|$ steps.

As with regimes, we can simplify the computation using Lemma 5: if $\gamma(a, \lambda, \mathbf{W}) > 0$, then Lemma 5 implies that the minimal regime can taken to be recursive. Lemma 3 then implies that $x(s, \lambda, \mathbf{a} \setminus (s, a), \mathbf{W}) > x(s, \lambda, \mathbf{a}, \mathbf{W})$ if and only if $x^R(a, \lambda, \mathbf{a}, \mathbf{W}) > x(s, \lambda, \mathbf{a}, \mathbf{W})$. Otherwise, binding APS payoffs are maximal, and we can replace x^{APS} with $\widehat{x}^{APS}$ in (7).

Thus, we are able to compute an optimal policy and the optimal levels in each direction using only binding payoffs and the sign of the APS gap. We summarize this discussion in

[12] In fact, since the levels are decreasing, it is not hard to see that if $x^R(\mathbf{a}(s), \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}) \leq x^{APS}(\mathbf{a}(s), \lambda, \mathbf{W})$, the recursive regime will be minimal at all subsequent rounds. Thus, a switch from APS to R is irreversible, and the maximum number of regime substitutions is $2|S|$.

the following result:

Proposition 2. *Suppose that* $\mathbf{A}(\mathbf{W})$ *is non-empty valued and* B *sub-generates at* $\mathbf{W}$*. Then the preceding algorithm converges to an optimal policy in at most* $|\mathbf{A}||\mathbf{R}|$ *steps.*

Remark 6. Given a policy $(\mathbf{a}, \mathbf{r})$, there may be multiple action substitutions that lead to higher levels or regime substitutions which lead to lower levels. The arguments above shows that the procedure is order independent, and it will converge to an optimal policy as long at least one improving substitution is implemented at each round. The pseudocode in Supplemental Appendix C makes a substitution in each state where there is an improvement, although it does not specify how to select from multiple improving substitutions in a given state.

By the discussion preceding Proposition 2, we obtain a refinement of (L2.ii) and (L4.ii), which we record for future reference:

Lemma 7. *Suppose that* B *sub-generates at* $\mathbf{W}$ *in the direction* λ*, and fix* $\mathbf{a} \in \mathbf{A}(\mathbf{W})$*. The regimes* $\mathbf{r}$ *are minimal for* $(\lambda, \mathbf{a})$ *if and only if*

$$x(s, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}) = \begin{cases} \min\left\{\widehat{x}^{APS}(\mathbf{a}(s), \lambda, \mathbf{W}), x^R(\mathbf{a}(s), \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W})\right\} & \text{if } \gamma(a, \lambda, \mathbf{W}) = 0; \\ x^R(\mathbf{a}(s), \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}) & \text{if } \gamma(\mathbf{a}(s), \lambda, \mathbf{W}) > 0. \end{cases} \tag{10}$$

Also, $\mathbf{a}$ *is optimal if and only if for all* $s \in S$ *and* $a \in \mathbf{A}(\mathbf{W})(s)$*,*

$$x(s, \lambda, \mathbf{a}, \mathbf{W}) \geq \begin{cases} \min\left\{\widehat{x}^{APS}(a, \lambda, \mathbf{W}), x^R(a, \lambda, \mathbf{a}, \mathbf{W})\right\} & \text{if } \gamma(a, \lambda, \mathbf{W}) = 0; \\ x^R(a, \lambda, \mathbf{a}, \mathbf{W}) & \text{if } \gamma(a, \lambda, \mathbf{W}) > 0. \end{cases} \tag{11}$$

3.5 Optimal payoffs

Our analysis thus far has simplified matters by focusing on optimal levels and suppressing the payoffs, either APS or recursive, that attain these levels. This perspective is useful when computing the bound in a single direction, but it is insufficient for understanding how $x(s, \lambda, \mathbf{W})$ changes as we vary λ. As noted in Remark 3, $x(s, \lambda, \mathbf{W})$ may be non-convex in λ, so that the payoffs that attain the optimal level need not be elements of $\widetilde{B}(\mathbf{W})$. Nonetheless, they turn out to be an essential part of our analysis. We re-introduce payoffs and establish basic results that will be used in the following sections. To do so, we will enrich the regimes to include information on which payoff is used when the regime is APS (in which case we presume that incentive constraints bind, per the hypothesis that B sub-generates at $\mathbf{W}$ and Lemma 5).

Given $\mathbf{W}$ and $a \in \mathbf{A}(\mathbf{W})(s)$, we define

$$C(a, \mathbf{W}) = \text{ext}\{v | v \in B(a, \mathbf{W}) \text{ and (2) binds for some } i\},$$

where ext X denotes the set of extreme points of the set X. Thus, for every λ, $\widehat{x}^{APS}(a, \lambda, \mathbf{W}) = \max\{\lambda \cdot v | v \in C(a, \mathbf{W})\}$. For any $\mathbf{a} \in \mathbf{A}(\mathbf{W})$, denote by $\mathbf{P}(\mathbf{a}, \mathbf{W})$ the set of selections $\mathbf{p}(s) \in \{R\} \cup C(\mathbf{a}(s), \mathbf{W})$. Thus, $\mathbf{p}$ encodes whether the regime is recursive or APS, and a choice of binding payoff if the regime is APS. Given a tuple of payoffs $\mathbf{u}$ and $a \in \mathbf{A}(s)$, let

$$u^R(a, \mathbf{u}) = (1 - \delta)g(a) + \delta \sum_{s' \in S} \pi(s'|a)\mathbf{u}(s').$$

Next, we refer to a triple (s, a, p) with $a \in \mathbf{A}(\mathbf{W})(s)$ and $p \in \{R\} \cup C(a, \mathbf{W})$ as a *substitution*. Given a substitution (s, a, p), let $u(s, a, p, \mathbf{u}) = u^R(a, \mathbf{u})$ if $p = R$, and $u(s, a, p, \mathbf{u}) = p$ if $p \in C(a, \mathbf{W})$. A pair $(\mathbf{a}, \mathbf{p})$ with $\mathbf{p} \in \mathbf{P}(\mathbf{a}, \mathbf{W})$ induces a tuple of payoffs $\mathbf{u}$ that solve the recursive system

$$\mathbf{u}(s) = u(s, \mathbf{a}(s), \mathbf{p}(s), \mathbf{u}) \tag{12}$$

for all $s \in S$. Note that (12) has a unique solution, by analogous arguments as for (L1.ii).

We now explain how to translate an optimal policy into a corresponding pair and payoffs. Fix λ and $\mathbf{a} \in \mathbf{A}(\mathbf{W})$. We maintain that B sub-generates at $\mathbf{W}$, so that there exist minimal $\mathbf{r}$ satisfying the selection of Lemma 5, i.e., $\mathbf{r}(s) = APS$ only if $x^{APS}(\mathbf{a}(s), \lambda, \mathbf{W}) = \widehat{x}^{APS}(\mathbf{a}(s), \lambda, \mathbf{W})$. We say that $\mathbf{p} \in \mathbf{P}(\mathbf{a}, \mathbf{W})$ is *min-max for* $(\lambda, \mathbf{a})$ if for such minimal regimes $\mathbf{r}$, $\mathbf{p}(s) = R$ if $\mathbf{r}(s) = R$, and $\lambda \cdot \mathbf{p}(s) = x^{APS}(\mathbf{a}(s), \lambda, \mathbf{W}) = \widehat{x}^{APS}(\mathbf{a}(s), \lambda, \mathbf{W})$ otherwise. In addition, the pair $(\mathbf{a}, \mathbf{p})$ is *optimal for* λ if $\mathbf{a}$ is maximal for λ and $\mathbf{p}$ is min-max for $(\lambda, \mathbf{a})$. Finally, we say that the payoffs $\mathbf{u}$ are *optimal for* λ if they are induced by a pair that is optimal for λ.

We next establish that optimal pairs exist and attain the optimal level:

Lemma 8. *Fix λ and $\mathbf{W}$, and suppose that B sub-generates at $\mathbf{W}$. Then for all $\mathbf{a} \in \mathbf{A}(\mathbf{W})$, there exists a $\mathbf{p} \in \mathbf{P}(\mathbf{a}, \mathbf{W})$ that is min-max for $(\lambda, \mathbf{a})$, and $x(\lambda, \mathbf{a}, \mathbf{W}) = \lambda \cdot \mathbf{u}$ where $\mathbf{u}$ is induced by $(\mathbf{a}, \mathbf{p})$. As a result, if $\mathbf{A}(\mathbf{W})$ is non-empty valued, then there exists a pair $(\mathbf{a}, \mathbf{p})$ that is optimal for λ, and for any payoffs $\mathbf{u}$ that are optimal for λ, $x(s, \lambda, \mathbf{W}) = \lambda \cdot \mathbf{u}$.*

Proof of Lemma 8. The existence of a min-max $\mathbf{p}$ is immediate from the preceding discussion. Let $\mathbf{u}$ denote the payoffs induced by $(\mathbf{a}, \mathbf{p})$. From the definition of a min-max $\mathbf{p}$, it follows directly that if $\mathbf{p}(s) \neq R$, then $\lambda \cdot \mathbf{u}(s) = \widehat{x}^{APS}(\mathbf{a}(s), \lambda, \mathbf{W}) = x(s, \lambda, \mathbf{a}, \mathbf{W})$. And if $\mathbf{r}(s) = \mathbf{p}(s) = R$, then $\mathbf{u}(s) = u^R(\mathbf{a}(s), \mathbf{u})$. A routine calculation shows that $\lambda \cdot \mathbf{u}$ is the fixed

point of $T(\cdot, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W})$, so that the minimal levels are attained. Finally, when $\mathbf{a}$ are optimal actions, these levels must coincide with $x(s, \lambda, \mathbf{W})$. □

We next state optimality conditions for a pair $(\mathbf{a}, \mathbf{p})$ that are analogous to (10) and (11), which will allow us to work directly with pairs rather than policies in computing $\widetilde{B}(\mathbf{W})$.

Lemma 9. *Suppose that B sub-generates at $\mathbf{W}$, and fix λ, $\mathbf{a} \in \mathbf{A}(\mathbf{W})$, and $\mathbf{p} \in \mathbf{P}(\mathbf{a}, \mathbf{W})$, and let $\mathbf{u}$ be induced by $(\mathbf{a}, \mathbf{p})$. Then $\mathbf{p}$ is min-max for $(\lambda, \mathbf{a})$ if and only if for all s:*

$$\lambda \cdot \mathbf{u}(s) = \min \left\{ \max \left\{ \lambda \cdot v | v \in C(\mathbf{a}(s), \mathbf{W}) \right\}, \lambda \cdot u^R(\mathbf{a}(s), \mathbf{u}) \right\}$$

if $\gamma(\mathbf{a}(s), \lambda, \mathbf{W}) = 0$, and $\mathbf{p}(s) = R$ if $\gamma(\mathbf{a}(s), \lambda, \mathbf{W}) > 0$ (so that $\mathbf{u}(s) = u^R(\mathbf{a}(s), \mathbf{u})$). Moreover, $(\mathbf{a}, \mathbf{p})$ is optimal for λ if and only if: $\mathbf{p}$ is min-max and for all s and $a \in \mathbf{A}(\mathbf{W})(s)$,

$$\lambda \cdot \mathbf{u}(s) \geq \begin{cases} \min \left\{ \max \left\{ \lambda \cdot v | v \in C(a, \mathbf{W}) \right\}, \lambda \cdot u^R(a, \mathbf{u}) \right\} & \text{if } \gamma(a, \lambda, \mathbf{W}) = 0; \\ \lambda \cdot u^R(a, \mathbf{u}) & \text{if } \gamma(a, \lambda, \mathbf{W}) > 0. \end{cases}$$

Proof of Lemma 9. This follows from Lemmas 7 and 8, with the observation that $x(s, \lambda, \mathbf{a}) = \lambda \cdot \mathbf{u}(s)$ for the payoffs $\mathbf{u}$ induced by $(\mathbf{a}, \mathbf{p})$, and that $x^R(a, \lambda, \mathbf{a}) = \lambda \cdot u^R(a, \mathbf{u})$. □

Remark 7. The algorithm of Proposition 2 is easily adapted to directly compute an optimal pair. Each policy $(\mathbf{a}, \mathbf{r})$ satisfying the selection of Lemma 5 can be identified with an equivalent pair $(\mathbf{a}, \mathbf{p})$, where $\mathbf{p}(s) = R$ if $\mathbf{r}(s) = R$ and otherwise $\mathbf{p}(s)$ is a highest binding payoff for $\mathbf{a}(s)$. It is immediate that the induced payoffs $\mathbf{u}$ satisfy $\lambda \cdot \mathbf{u}(s) = x(s, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W})$ for all s. In iterating over pairs, just as with policies, there is an inner min-maximization over $\mathbf{p}$ and an outer maximization over $\mathbf{a}$. The computation of the APS gap remains the same (and we will comment below on how to do this efficiently). But when comparing the recursive and binding APS levels to test for an improvement, instead of using $x^R(a, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W})$ and $\widehat{x}^{APS}(a, \lambda, \mathbf{W})$, we use $\lambda \cdot u^R(a, \mathbf{u})$ and $\lambda \cdot v$, where v is a highest binding payoff. Starting from $(\mathbf{a}, \mathbf{p})$ where $\mathbf{p}(s)$ is recursive or maximal in $C(a, \mathbf{W})$, the computed sequence of pairs corresponds to a sequence of policies generated by the algorithm of Proposition 2, as long as ties are broken the same way (cf. Remark 6).

4 Two players: Further implications and implementation

We now specialize to two-player games which, as we shall see, have an especially simple structure: the number of extreme equilibrium payoffs is bounded, $\widetilde{B}$ has bounded computational

complexity, and it can be computed via a simple procedure.

4.1 The complexity of V

We first establish a bound on the number of optimal pairs.

Lemma 10. *If $N = 2$ and $\mathbf{W}$ is convex valued, then $|C(a, \mathbf{W})| \leq 4$, and the number of pairs is at most*

$$\overline{L} = 5^{|S|} \times_{s \in S} |\mathbf{A}(s)|. \tag{13}$$

Proof of Lemma 10. Recall the definition of $C(a, \mathbf{W})$ and condition (2). If w is a feasible and incentive compatible expected continuation value for which player i's incentive constraint binds, then $w_i = \underline{u}_i(a, \mathbf{W})$. The set of such points is either empty, a singleton, or it has at most two extreme points. Thus, $|C(a, \mathbf{W})| \leq 4$. Finally, each optimal payoff tuple is associated with some $\mathbf{a} \in \mathbf{A}(\mathbf{W})$ and $\mathbf{p} \in \mathbf{P}(\mathbf{a}, \mathbf{W})$. But by the preceding argument, $|C(\mathbf{a}(s), \mathbf{W})| \leq 4$, so $|\mathbf{P}(\mathbf{a}, \mathbf{W})| \leq 5^{|S|}$. The bound immediately follows. □

We immediately obtain a bound on the complexity of $\mathbf{V}$, which generalizes an analogous result of Abreu and Sannikov (2014) for repeated games:

Corollary 1. *If $N = 2$, then $\mathbf{V}$ has at most $\overline{L}$ extreme points.*

Proof of Corollary 1. The proof of (T1.iii) shows that $x(s, \lambda, \mathbf{V})$ is the support function of $\mathbf{V}$. Since there are at most $\overline{L}$ pairs, $x(s, \lambda, \mathbf{V})$ can have at most $\overline{L}$ linear segments. As a result, $\mathbf{V}$ has at most $\overline{L}$ extreme points. □

Note that Lemma 10 does not immediately yield a bound on the complexity of $\widetilde{\mathbf{W}}^k$. The fact that these correspondences have bounded complexity will, however, be proven using the computational procedure we describe next.

4.2 Implementation with two players

Under the hypothesis that B sub-generates at $\mathbf{W}$, there is a simple procedure to compute $\widetilde{B}(\mathbf{W})$, which consists of iterative application of two subroutines. Starting from an optimal pair for some initial direction, we compute a set of clockwise rotations of the direction for which that pair remains optimal. We then rotate the direction of optimization clockwise as far as we can, subject to the incumbent pair remaining optimal, and compute the pair that would become optimal if the direction were to rotate further by a small amount. We then continue iteratively: compute a new range of directions, rotate, and re-optimize. The main

result for this section, Theorem 2, shows that this procedure maps out the entire frontier of $\widetilde{B}(\mathbf{W})$ in a bounded number of steps.

The key step in the algorithm is to compute a range of directions for which an optimal pair remains optimal. We will show that a suitably chosen optimal pair can only cease to be optimal at what we call a *test direction*. These directions are identified with the substitutions of either actions or regimes that are used in the optimization routines of Proposition 2 and Remark 7. Specifically, the test directions corresponding to a substitution are the critical directions at which that substitution would leave the level unchanged. Proposition 3 below establishes two fundamental results: First, except when $\widetilde{B}(\mathbf{W})$ is degenerate, a test direction always exists. Second, the incumbent optimal pair remains optimal for all directions between the initial direction and the "shallowest" test direction, i.e., the test direction with the smallest clockwise angle of rotation from the initial direction.

This high-level summary neglects two complications that are dealt with in the following pages. First, we will impose a refinement on the test directions considered by our algorithm that we call *legitimacy*. This condition is used to weed out some spurious test directions at which the optimal pair would not change. The legitimacy condition can, however, be dropped without affecting our results on convergence and computational complexity.

A more substantive issue is what is meant by a "suitably chosen optimal pair." The test direction methodology requires us to start from an optimal pair that is *robust*, in the sense that it remains optimal for small clockwise perturbations of the direction. Proposition 4 characterizes a procedure called *lexicographic optimization*, which is guaranteed to produce a robustly optimal pair for any direction.

Thus, Propositions 3 and 4 establish three results that are used to prove Theorem 2: Given a robustly optimal pair, there exists a legitimate test direction, and the incumbent pair remains optimal to the next shallowest legitimate test direction (Proposition 3). Moreover, for any direction, there exists a robustly optimal pair and a method to compute it (Proposition 4). As a result, iterative application of direction rotation and optimization will necessarily identify an optimal pair for every direction.

4.2.1 Rotating the direction clockwise

We now proceed formally. For rotating the direction, we will work with pairs that satisfy a mild refinement: A pair $(\mathbf{a}, \mathbf{p})$ with induced payoffs $\mathbf{u}$ is *canonical* if $\mathbf{p}(s) = R$ whenever $u^R(\mathbf{a}(s), \mathbf{u}) \in C(\mathbf{a}, \mathbf{W})$. In other words, if a binding payoff is exactly equal to the recursive payoff, then we break ties in favor of the recursive regime. In this situation, recursive is "canonically" minimal in the sense that it generates the lowest level in all directions, as the recursive payoff is also an APS payoff. (Note, however, that $\mathbf{p}$ in a canonical pair

need not be min-max, nor does the definition depend on the direction of optimization). It is immediate that for any optimal pair, there is another optimal pair with the same payoffs that is canonical, which is obtained by switching the regime to $\mathbf{r}$ whenever $\mathbf{p}(s) = u^R(\mathbf{a}(s), \mathbf{u})$:

Lemma 11. *If $(\mathbf{a}, \mathbf{p})$ is optimal for λ and induces $\mathbf{u}$, then there exists a $\mathbf{p}' \in \mathbf{P}(\mathbf{a}, \mathbf{W})$ such that $(\mathbf{a}, \mathbf{p}')$ is optimal for λ, induces $\mathbf{u}$, and is canonical.*

Now, fix a payoff tuple $\mathbf{u}$ and substitution (s, a, p), where $a \in \mathbf{A}(\mathbf{W})(s)$ and $p \in \{R\} \cup C(a, \mathbf{W})$. Recall that $u(s, a, p, \mathbf{u})$ is equal to p if $p \in C(a, \mathbf{W})$ and is equal to $u^R(a, \mathbf{u})$ if $p = R$. We say that λ' is *test direction* for (s, a, p) at $\mathbf{u}$ if $u(s, a, p, \mathbf{u}) \neq \mathbf{u}(s)$ and

$$\lambda' \cdot (u(s, a, p, \mathbf{u}) - \mathbf{u}(s)) = 0. \tag{14}$$

In other words, λ' is normal to the direction in which the substitution moves payoffs. Furthermore, we say that (s, a, p) and a corresponding test direction λ' are *legitimate* if

$$\lambda' \cdot u(s, a, p, \mathbf{u}) \leq \min \left\{ \lambda' \cdot u^R(s, a, p, \mathbf{u}), x^{APS}(a, \lambda, \mathbf{W}) \right\}.$$

We note for future reference that the number of test directions is bounded. For there are at most $\overline{L}$ pairs, and the number of substitutions is at most

$$\overline{M} = 5 \sum_{s \in S} |\mathbf{A}(s)|.$$

Moreover, each pair and substitution has at most two associated test directions that solve (14). As a result, there are at most $2\overline{L}\,\overline{M}$ test directions.

Next, let us define $[\lambda, \lambda']$ to be the *closed arc* of directions obtained by moving clockwise from λ to λ'. We extend this convention to open and half-closed arcs in the obvious way. We say that a pair $(\mathbf{a}, \mathbf{p})$ is *robustly optimal* at λ if there exists $\lambda' \neq \lambda$ such that $(\mathbf{a}, \mathbf{p})$ is optimal for all directions in $[\lambda, \lambda']$. A payoff $\mathbf{u}$ is robustly optimal if it is induced by a robustly optimal pair.

The following proposition characterizes the search for shallowest legitimate test directions and their relationship with robustly optimal pairs.

Proposition 3 (Test directions)**.** *Suppose that $N = 2$ and that B sub-generates at $\mathbf{W}$. Suppose further that $(\mathbf{a}, \mathbf{p})$ is robustly optimal at λ and induces $\mathbf{u}$. Then either $\mathbf{u}$ is optimal for all directions, or there exists a legitimate test direction at $\mathbf{u}$ that is not equal to λ. Moreover, if $\lambda' \neq \lambda$ is the shallowest test direction and $(\mathbf{a}, \mathbf{p})$ is canonical, then $(\mathbf{a}, \mathbf{p})$ is optimal for all directions in $[\lambda, \lambda']$.*

We now prove Proposition 3. Note that the remaining results of this section all maintain the implicit hypotheses that B sub-generates at $\mathbf{W}$, so that binding payoffs are sufficient, and that $N = 2$. We first record a simple technical result.

Lemma 12. *For each* $(\mathbf{a}, \mathbf{p})$*, the set of directions in which* $\mathbf{p}$ *is min-max for* $\mathbf{a}$ *is closed, and the set of directions in which* $(\mathbf{a}, \mathbf{p})$ *is optimal is closed. For each* $\mathbf{u}$*, the set of directions in which* $\mathbf{u}$ *is optimal is closed.*

Proof of Lemma 12. This is a consequence of Lemma 9 and the fact that γ, $\widehat{x}^{APS}$, and x^R are all continuous in the direction. As a result, if $(\mathbf{a}, \mathbf{p})$ satisfies either the min-max or optimality conditions of Lemma 9 along a convergent sequence of directions, then it also satisfies them in the limit. Finally, $\mathbf{u}$ is optimal on the union of the finitely many closed sets for which pairs that induce $\mathbf{u}$ are optimal. □

The next two lemmas establish the first part of Proposition 3.

Lemma 13. *If* $\mathbf{u}$ *and* $\mathbf{u}' \neq \mathbf{u}$ *are both optimal at some direction* λ'*, then* λ' *is a legitimate test direction at* $\mathbf{u}$*.*

Proof of Lemma 13. Let $(\mathbf{a}', \mathbf{p}')$ be an optimal pair in the direction λ' that induces $\mathbf{u}'$. It must be that $u(s, \mathbf{a}'(s), \mathbf{p}'(s), \mathbf{u}) \neq \mathbf{u}(s)$ for some s. Otherwise, $\mathbf{u}$ is a solution to (12) for $(\mathbf{a}', \mathbf{p}')$, and uniqueness of the solution would imply that $\mathbf{u} = \mathbf{u}'$, a contradiction. As a result, λ' is a test direction for the substitution $(s, \mathbf{a}'(s), \mathbf{p}'(s))$. Moreover, $\lambda' \cdot u(s, \mathbf{a}'(s), \mathbf{p}'(s), \mathbf{u}) = \lambda' \cdot \mathbf{u}'(s)$ for all s, so that legitimacy follows from the optimality conditions for $(\mathbf{a}', \mathbf{p}')$. □

Lemma 14. *Suppose that* $\mathbf{u}$ *is robustly optimal at* λ *and is not optimal at* $\widehat{\lambda} \neq \lambda$*. Then there is a legitimate test direction in* $(\lambda, \widehat{\lambda})$*.*

Proof of Lemma 14. By Lemma 12, the set of directions at which $\mathbf{u}$ is optimal is closed, so that there is a largest closed arc $[\lambda, \lambda']$ on which $\mathbf{u}$ is optimal. We will show that there exists $\mathbf{u}' \neq \mathbf{u}$ that is optimal at λ', so the result follows from Lemma 13.

By assumption, $\mathbf{u}$ is not optimal at $\widehat{\lambda}$, and since $\mathbf{u}$ is robustly optimal at λ, we conclude that $\lambda' \in (\lambda, \widehat{\lambda})$. The definition of λ' then implies that there is a sequence of directions in $(\lambda', \widehat{\lambda}]$ converging to λ' at which $\mathbf{u}$ is not optimal. Since there is an optimal pair for every direction (Lemma 8) and only finitely many pairs (Lemma 10), there must be a pair $(\mathbf{a}', \mathbf{p}')$ which induces payoffs $\mathbf{u}' \neq \mathbf{u}$, such that $(\mathbf{a}', \mathbf{p}')$ is optimal for directions arbitrarily close to λ'. Lemma 12 then implies that $(\mathbf{a}', \mathbf{p}')$ and $\mathbf{u}'$ are also optimal at λ'. □

The next lemma is used to prove the second part of Proposition 3.

Lemma 15. *Suppose that* $(\mathbf{a}, \mathbf{p})$ *is robustly optimal at* λ *and induces* $\mathbf{u}$*, and that* $\lambda' \neq \lambda$ *is the shallowest legitimate test direction at* $\mathbf{u}$*. If* $(\mathbf{a}, \mathbf{p})$ *is canonical, then it is optimal for all directions in* $[\lambda, \lambda']$*.*

Proof of Lemma 15. The proof consists of three steps.

Step 1: $\mathbf{u}$ is optimal for all directions in (λ, λ'). If not, then there is a $\widehat{\lambda} \in (\lambda, \lambda')$ at which it is not optimal, and Lemma 14 then implies that there is a legitimate test direction in $(\lambda, \widehat{\lambda})$, which contradicts the hypothesis that λ' is shallowest.

Step 2: If $(\mathbf{a}, \mathbf{p})$ is canonical and is optimal at $\widehat{\lambda} \in (\lambda, \lambda')$, then there is a closed neighborhood of $\widehat{\lambda}$ on which $(\mathbf{a}, \mathbf{p})$ is optimal.

By Step 1 and the fact that $(\mathbf{a}, \mathbf{p})$ induces $\mathbf{u}$, $(\mathbf{a}, \mathbf{p})$ is optimal in a neighborhood of $\widehat{\lambda}$ if and only if $\mathbf{p}$ is min-max for $\mathbf{a}$ in this neighborhood. The following two cases establish that for every s, there is a neighborhood of $\widehat{\lambda}$ on which the min-max condition in Lemma 9 is satisfied. Thus, $\mathbf{p}$ is min-max on the intersection of these finitely many neighborhoods, which, together with Lemma 12, implies the result.

Case 1: $\mathbf{p}(s) \neq R$. Then for all $v \in C(\mathbf{a}(s), \mathbf{W}) \setminus \{\mathbf{p}(s)\}$, it must be that $\widehat{\lambda} \cdot v < \widehat{\lambda} \cdot \mathbf{p}(s)$. Otherwise $\mathbf{p}(s)$ would not be maximal or, if there is a tie for maximal payoff, then Lemma 13 implies that $\widehat{\lambda}$ is a legitimate test direction, contradicting the hypothesis that λ' is shallowest. Thus, there is a neighborhood of $\widehat{\lambda}$ for which $\mathbf{p}(s)$ is the highest binding APS payoff. Next, since $(\mathbf{a}, \mathbf{p})$ is canonical, it must be that $u^R(\mathbf{a}(s), \mathbf{u}) \neq \mathbf{p}(s)$.[13] Moreover, it must be that $\widehat{\lambda} \cdot u^R(\mathbf{a}(s), \mathbf{u}) > \widehat{\lambda} \cdot \mathbf{p}(s)$, for otherwise $\widehat{\lambda}$ would again be a legitimate test direction. Finally, if there are directions arbitrarily close to $\widehat{\lambda}$ for which the APS gap is positive, then Lemma 5 implies that $\widehat{\lambda} \cdot u^R(\mathbf{a}(s), \mathbf{u}) \leq x^{APS}(\widehat{\lambda}, \mathbf{W})$ for $\widehat{\lambda}$ arbitrarily close to $\widehat{\lambda}$. Continuity of x^{APS} then implies that $\widehat{\lambda} \cdot u^R(\mathbf{a}(s), \mathbf{u}) \leq x^{APS}(\widehat{\lambda}, \mathbf{W}) = \widehat{\lambda} \cdot \mathbf{p}(s)$, again a contradiction. We conclude that there is a neighborhood of $\widehat{\lambda}$ for which $u^R(\mathbf{a}(s), \mathbf{u})$ is strictly above $\mathbf{p}(s)$.

Case 2: $\mathbf{p}(s) = R$. If $\gamma(\mathbf{a}(s), \widehat{\lambda}, \mathbf{W}) > 0$, then continuity of the APS gap implies that $\mathbf{p}(s) = R$ is minimal for a neighborhood of $\widehat{\lambda}$. Otherwise, the APS gap is zero and there is a maximal APS payoff at $\widehat{\lambda}$ that is binding. If $\widehat{x}^{APS}(\widehat{\lambda}, \mathbf{W}) > \widehat{\lambda} \cdot \mathbf{u}(s)$, then again, continuity implies that the recursive regime is minimal in a neighborhood of $\widehat{\lambda}$. If $\widehat{x}^{APS}(\widehat{\lambda}, \mathbf{W}) = \widehat{\lambda} \cdot \mathbf{u}(s)$, then $\widehat{\lambda} \cdot v = \widehat{\lambda} \cdot \mathbf{u}(s)$ for some maximal $v \in C(\mathbf{a}(s), \mathbf{W})$. If $\mathbf{u}(s) \neq v$, then $\widehat{\lambda}$ is a legitimate test direction corresponding to the substitution $(s, \mathbf{a}(s), v)$, which contradicts λ' being shallowest. Otherwise, it must be that $\mathbf{u}(s) = v$. As a result, the recursive payoff is also an APS payoff,

[13] This is the only step in the argument that uses the hypothesis $(\mathbf{a}, \mathbf{p})$ is canonical. Without this hypothesis, it could be that $\mathbf{p}(s) \neq R$ and $\mathbf{p}(s) = \mathbf{u}(s) = u^R(\mathbf{a}(s), \mathbf{u})$. Hence, the recursive regime is minimal for all directions, but the recursive and APS regimes happen to tie at $\widehat{\lambda}$. As the direction rotates, the recursive regime may become *uniquely* minimal, because the APS gap becomes positive. This need not happen at a test direction. Instead of selecting canonical pairs, we could have included additional test directions when the APS gap switches sign. These are the normals to the non-binding APS frontier at a binding payoff.

so that $\mathbf{p}(s) = R$ is min-max for $\mathbf{a}$ in all directions.

Step 3: Let $[\underline{\lambda}, \overline{\lambda}]$ be a largest closed arc in $[\lambda, \lambda']$ on which $(\mathbf{a}, \mathbf{p})$ is optimal, which by hypothesis is non-empty. If $\overline{\lambda} \neq \lambda'$, then Step 2 with $\widehat{\lambda} = \overline{\lambda}$ implies that there is a closed neighborhood of $\overline{\lambda}$, denoted U, on which $(\mathbf{a}, \mathbf{p})$ is optimal. Then $(\mathbf{a}, \mathbf{p})$ is optimal on $[\underline{\lambda}, \overline{\lambda}] \cup U$, which is a strict superset of $[\underline{\lambda}, \overline{\lambda}]$, a contradiction. We similarly conclude that $\underline{\lambda} = \lambda$, so that $(\mathbf{a}, \mathbf{p})$ is optimal for all directions in $[\lambda, \lambda']$. □

Proof of Proposition 3. The proposition follows directly from Lemmas 14 and 15. □

4.2.2 Finding a robustly optimal pair

The last task is to find a robustly optimal pair and payoffs in a direction λ. This is accomplished with a procedure that we refer to as *lexicographic optimization*: Starting from $(\mathbf{a}, \mathbf{p})$, we optimize according to the procedure of Proposition 2, using payoffs as described in Remark 7, except that in ranking any pair of payoffs v and v', we use the *lexicographic ordering*, whereby a payoff v is greater than v' if $\lambda \cdot v > \lambda \cdot v'$ or if $\lambda \cdot v = \lambda \cdot v'$ and $\widetilde{\lambda} \cdot v > \widetilde{\lambda} \cdot v'$, where $\widetilde{\lambda}$ is equal to λ rotated 90 degrees clockwise. In this case, we write $v >_\lambda v'$. Note that the use of the lexicographic order only affects how ties are broken in the algorithm (cf. Remark 6). In particular, instead of using any highest payoff in $C(a, \mathbf{W})$, we use the lexicographically highest, and when the APS and recursive payoffs are tied, we select whichever is lexicographically minimal. In addition, we break ties in favor of the recursive regime if the APS gap is zero but would become strictly positive for small clockwise rotations: In particular, wherever the condition $\gamma(a, \lambda, \mathbf{W}) > 0$ was used previously, we now use the condition that there exists a $\lambda' \neq \lambda$ such that $\gamma(a, \lambda'', \mathbf{W}) > 0$ for all $\lambda'' \in (\lambda, \lambda']$. In this case, we say that the APS gap for a is *lexicographically positive* at λ'. (Note that continuity of γ in λ implies that the APS gap is lexicographically positive whenever it is positive.) This procedure is fully described in Algorithms 4 and 5 in Supplemental Appendix C.

The following proposition characterizes lexicographic optimization and, as a corollary, shows that a robustly optimal pair exists in every direction:

Proposition 4. *Suppose that $N = 2$, $\mathbf{A}(\mathbf{W})$ is non-empty valued, and B sub-generates at $\mathbf{W}$. Then lexicographic optimization in a direction λ terminates in finitely many steps at a pair $(\mathbf{a}, \mathbf{p})$ that is robustly optimal at λ. Moreover, once the algorithm has reached an optimal pair for λ, every subsequent pair is also optimal for λ.*

Proof of Proposition 4. Since lexicographic optimization only affects how ties are broken, convergence of the algorithm described in Remark 7 implies that lexicographic optimization will reach a pair that is optimal for λ. Once such a pair is reached, there are no substitutions

that strictly improve in the direction λ. Thus, any substitution considered by lexicographic optimization will keep the λ level the same and move payoffs in the direction $\widetilde{\lambda}$, which is λ rotated 90 degrees clockwise. A straightforward adaptation of the argument for Proposition 2 shows that lexicographic regime minimization produces a sequence of payoffs that monotonically decrease in the direction $\widetilde{\lambda}$, and lexicographic action maximization produces payoffs that monotonically increase in the direction $\widetilde{\lambda}$, so that no pair is repeated and the sequence converges after finitely many steps to a limit $(\mathbf{a}, \mathbf{p})$.

We claim that $(\mathbf{a}, \mathbf{p})$ is robustly optimal at λ. An important fact used below is that $v >_\lambda v'$ if and only if there exists $\lambda' \neq \lambda$ such that $\lambda'' \cdot v > \lambda'' \cdot v'$ for all $\lambda'' \in (\lambda, \lambda']$. Now, we will argue that $(\mathbf{a}, \mathbf{p})$ satisfies the optimality conditions of Lemma 9 for small clockwise rotations from λ. Since lexicographic optimization has converged, we know that the analogue of these conditions is satisfied at λ, where we select the lexicocographically lowest of the recursive payoff and the lexicographically highest binding payoff, and we select the recursive payoff if the APS gap is lexicographically positive at λ. We shall argue that this implies that the conditions in Lemma 9 are satisfied for small clockwise rotations.

Let us first consider the min-max conditions. If the APS gap is lexicographically positive for $\mathbf{a}(s)$, then $\mathbf{u}(s) = u^R(\mathbf{a}(s), \mathbf{u})$, and, moreover, the APS gap is positive for some arc $(\lambda, \lambda']$ with $\lambda' \neq \lambda$, so that $\mathbf{p}(s) = R$ is min-max on $(\lambda, \lambda']$. If the APS gap is not lexicographically positive, then there is an arc $(\lambda, \lambda'']$ over which (i) the APS gap is zero, (ii) the payoff v that is lexicographically highest at λ remains highest, and (iii) the ranking between v and $u^R(\mathbf{a}(s), \mathbf{u})$ is strict if and only if there is the same strict lexicographic ranking at λ. The fact that we have selected the lexicographically minimal of v and $u^R(\mathbf{a}(s), \mathbf{u})$ then implies that the Lemma 9 min-max conditions are satisfied on $(\lambda, \lambda']$.

The analysis for optimality is entirely analogous. We conclude that for every regime or action substitution, there is a non-trivial clockwise arc over which the Lemma 9 conditions are satisfied. Since there are finitely many substitutions, there exists a non-trivial clockwise arc over which $(\mathbf{a}, \mathbf{p})$ is optimal, so that $(\mathbf{a}, \mathbf{p})$ is robustly optimal. □

4.2.3 Computing $\widetilde{B}(\mathbf{W})$

We now summarize the computation of $\widetilde{B}(\mathbf{W})$ with two players, assuming that B subgenerates at $\mathbf{W}$ and there is a supportable action profile in each state (otherwise $\widetilde{B}(\mathbf{W})(s)$ is empty for some state, so that there are no pure-strategy subgame-perfect equilibria). A preliminary step is to compute, for each action profile, whether it is supportable, the sets $C(a, \mathbf{W})$, and the APS gap. This is done as follows. To compute $C(a, \mathbf{W})$, we simply intersect each of the binding incentive rays with the half spaces that define $\mathbf{W}$, and a is supportable

if and only if $C(a, \mathbf{W})$ is non-empty (cf. Footnote 11).[14] In addition, for each $v \in C(a, \mathbf{W})$, we record the direction on the frontier of $B(a, \mathbf{W})$ that points into the incentive compatible region, denoted $d(v)$. The APS gap is positive at λ if there is a $v \in \arg\max_{v \in C(a,\mathbf{W})} \lambda \cdot v'$ such that $\lambda \cdot d(v) > 0$, and it is lexicographically positive if $\lambda \cdot d(v) > 0$ or $d(v) = \alpha\widetilde{\lambda}$ for some $\alpha > 0$, where $\widetilde{\lambda}$ is equal to λ rotated 90 degrees clockwise.

After these preliminary calculations, we pick an initial direction $\lambda^0 \in \Lambda$, at which we compute a robustly optimal pair $(\mathbf{a}^0, \mathbf{p}^0)$ and its payoffs $\mathbf{u}^0$. If there are no legitimate test directions at $\mathbf{u}^0$, then we stop. Otherwise, proceeding inductively from $\mathbf{u}^{k-1}$ for $k \geq 1$, we set λ^k equal to the shallowest legitimate test direction from $\mathbf{u}^{k-1}$ and λ^{k-1}. We then compute a robustly optimal pair $(\mathbf{a}^k, \mathbf{p}^k)$ and corresponding robustly optimal payoffs $\mathbf{u}^k$ for λ^k, via lexicographic optimization starting from $(\mathbf{a}^{k-1}, \mathbf{p}^{k-1})$. We stop when the direction of optimization passes λ^0, at step K. This procedure is described in Algorithm 6.

Theorem 2. *Suppose that $N = 2$, $\mathbf{A}(\mathbf{W})$ is non-empty valued, and B sub-generates at $\mathbf{W}$. Then the previously described procedure terminates in at most $2\overline{L}\,\overline{M}$ substitutions and runtime $O(\overline{L}\,\overline{M}^2)$. If there are no legitimate test directions at $\mathbf{u}^0$, then $\widetilde{B}(\mathbf{W})(s) = \{\mathbf{u}^0(s)\}$ for all s. Otherwise,*

$$\widetilde{B}(\mathbf{W})(s) = \{v | \lambda^k \cdot v \leq \lambda^k \cdot \mathbf{u}^k(s) \ \forall k = 1, \ldots, K\}. \tag{15}$$

Proof of Theorem 2. As there are supportable actions in every state, convergence to the initial robustly optimal pair is guaranteed by Proposition 4. If there are no legitimate test directions, Proposition 3 implies $\mathbf{u}^0$ is optimal in every direction and is equal to $\widetilde{B}(\mathbf{W})$. Otherwise, there are legitimate test directions at every step of the algorithm. Proposition 3 implies that $\mathbf{u}^{k-1}$ is optimal on $[\lambda^{k-1}, \lambda^k]$, so that $\widetilde{B}(\mathbf{W})$ is equal to right-hand side of (15).

We next argue the complexity bound. The computation of $(\mathbf{a}^0, \mathbf{p}^0)$ takes at most $\overline{L}$ substitutions. In addition, there are at most $2\overline{L}\,\overline{M}$ test directions in which we optimize, and to compute the shallowest legitimate substitution requires the consideration of $\overline{M}$ substitutions. It remains to bound the number of substitutions that are actually made. If (s, a, p) is substituted into $(\mathbf{a}^k, \mathbf{p}^k)$, then both $(\mathbf{a}^k, \mathbf{p}^k)$ and $(\mathbf{a}^k, \mathbf{p}^k) \setminus (s, a, p)$ are optimal in a direction λ^k. As a result, λ^k is one of the two test directions satisfying (14). Since the direction of optimization rotates monotonically clockwise, it follows that (s, a, p) can be substituted into $(\mathbf{a}^k, \mathbf{p}^k)$ at most twice over the course of the algorithm. Thus, the total number of substitutions (and hence K) is at most $2\overline{L}\,\overline{M}$. Hence, the total runtime is $O(\overline{L}\,\overline{M}^2)$. □

[14] It is easy to see that $\mathbf{A}(\mathbf{W})$ is decreasing in $\mathbf{W}$, so that when $\widetilde{B}$ is applied iteratively to generate the sequence $\widetilde{\mathbf{W}}^k$ from (T1.iv), $C(a, \mathbf{W})$ need only be computed for action profiles in $\mathbf{A}(\widetilde{\mathbf{W}}^{k-1})$.

Remark 8. We have not ruled out the possibility that there are more than $\overline{L}$ bounding hyperplanes. This is because $x(s, \lambda, \mathbf{W})$ need not be convex, and the algorithm may come back to a pair that was previously found to be optimal. We note, however, that the run-time of the algorithm is sensitive to the actual number of bounding hyperplanes, which in practice we have found to be much smaller than $\overline{L}$.

4.2.4 Further improvements and single-state substitutions

The algorithm characterized by Section 4.2.3 nominally considers all substitutions when it searches for the next robust optimum. Proposition 4 shows that we can without loss restrict attention to substitutions which leave the pair optimal in the shallowest test direction, speeding up computation. It is possible to go even further, by restricting attention to test directions for which the associated substitutions which would become strict improvements for small clockwise rotations, using a lexicographic version of the legitimacy test. This is done in our software implementation.

In addition, there are many cases where we can find the next robustly optimal pair directly in a single step, as we now explain. If a "shallowest substitution" (that is, one that generates a shallowest test direction) entails a new action or a switch to or from a recursive regime, then frequently it will be unique, as ties across states or actions are exceptional. It is then trivial to check if the substitution generates the next robustly optimal pair, or whether the incumbent pair continues to be optimal at λ'.

On the other hand, if all shallowest substitutions entail a change from one binding APS payoff to another, then the next robustly optimal pair is obtained by switching all of the binding payoffs to the ones that are lexicographically optimal. For example, if the shallowest direction is $\lambda' = (-1, 0)$, which corresponds to punishing player 1, then payoffs may jump from minimizing to maximizing player 2's payoff, subject to minimizing player 1's payoff. This and the analogous situation for $\lambda' = (0, -1)$ occur frequently in our simulations.

Our code does not currently implement all of these simplifications. We view the software as a living entity which will continue to be improved by ourselves and the community. Such improvements can only lead to better performance than that reported in the next section.

4.3 Examples

We have implemented this algorithm in a software package called SGSolve, which is available through an author's website and Github, under the terms of the GPLv3 license. The code consists of routines that implement the algorithm and graphical interface for specifying games and visualizing solutions. The following two examples were solved using this package. An

a_1/a_2	D	C
C	$(-1,5)$	$(4,2)$
D	$(0,0)$	$(5,-1)$

Figure 1: An asymmetric Prisoners' Dilemma.

additional two-player example is reported in Online Appendix B.

4.3.1 Asymmetric Prisoners' Dilemma

Our first example is the asymmetric repeated Prisoners' Dilemma of Figure 1, with $\delta = 1/2$. As there is a single state, we temporarily drop the argument s and use regular font weight.

The computation is depicted in Figure 2. The flow payoffs are four circles. We take $\widetilde{W}^0$ to be the feasible and individually rational payoffs, which are in gray.[15] In the center panel, the shaded polygons are the sets $B(a, \widetilde{W}^0)$ generated by the APS operator. For each a and λ, $x^{APS}(a, \lambda, \widetilde{W}^0)$ is the level of the highest payoff in $B(a, \widetilde{W}^0)$, whereas $x(a, \lambda, \widetilde{W}^0)$ is the minimum of $x^{APS}(a, \lambda, \widetilde{W}^0)$ and the recursive level, which in a repeated game is just the level of the flow payoff.

The initial direction of optimization is $\lambda^0 = (0, 1)$. For APS, the optimal level is $x^{APS}((C, D), \lambda^0, \widetilde{W}^0)$, which is attained at the top left corner of the red set. This is also the max-min-max level, since $g(C, D) = (-1, 5)$ is strictly higher in the direction λ^0. Note that player 1's incentive constraint binds at the optimum, consistent with Proposition 1.

Rotating clockwise from λ^0, the algorithm computes a sequence of test directions and robustly optimal pairs. Between λ^0 and λ^1, $((C, D), APS)$ remains optimal. At λ^1, both $((C, D), R)$ is tied with $((C, C), APS)$, but the latter optimum is robust. At λ^2, $((D, C), R)$ becomes robustly optimal, as $g(D, C)$ is still below the best APS payoff for (D, C). Between λ^3 and λ^4, the robust optimum is $((D, C), APS)$. At λ^4, the optimal switches to $((D, D), R)$, which remains optimal until λ^5, at which point $((C, D), APS)$ is again optimal.

The corresponding hyperplanes for these directions and levels are then intersected to form $\widetilde{W}^1$. It is not hard to see that at the second round, the exact same set will be generated, i.e., $\widetilde{B}(\widetilde{W}^1) = \widetilde{W}^1$. For even though the binding payoff that was used between λ^2 and λ^3 is no longer available, the half spaces in this direction were redundant anyway. Thus, the operator converges after exactly one round.

In contrast, the APS, JYC, and Abreu and Sannikov (2014) operators would all cut less in the directions between λ^2 and λ^3. For APS, this is because the best APS payoffs for (D, C) are higher than the flow payoff. The operator of Abreu and Sannikov (2014) would

[15] If we started with $\widetilde{W}^0$ equal to the feasible set, then our algorithm would only compute the equilibrium threats asymptotically, and the algorithm would only converge asymptotically.

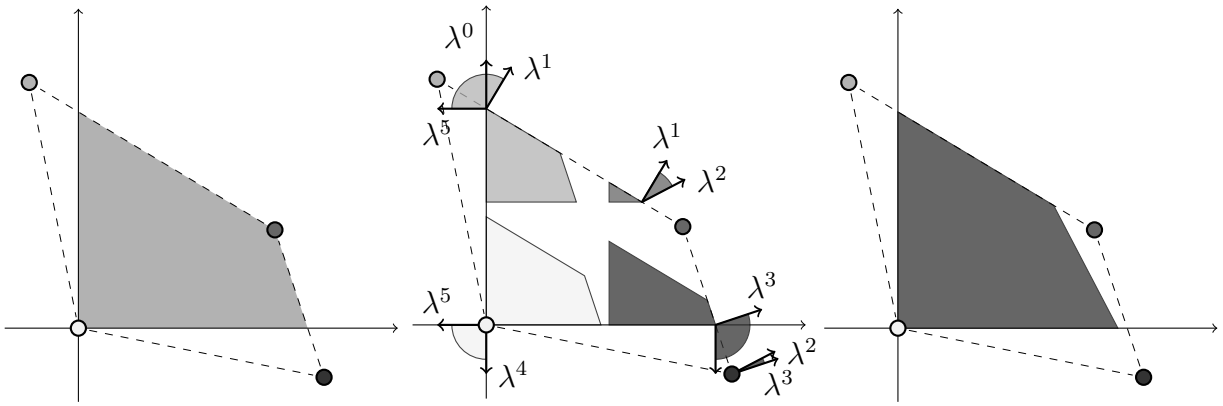

Figure 2: The asymmetric Prisoners' Dilemma. Left: Feasible and individually rational payoffs. Center: APS payoffs and optimal max-min-max levels. Right: Equilibrium payoffs.

use the best APS binding payoff for (D, C) between λ^2 and λ^3, thus leading to a higher level. Hence, all of these operators would generate a strictly larger $\widetilde{W}^1$. Moreover, in the second iteration, they would still be able to generate payoffs that are higher than $(5, -1)$ in directions between λ^2 and λ^3, and in fact, the optimal level converges only asymptotically.

4.3.2 Risk sharing

Our second example is a risk sharing game in the style of Kocherlakota (1996). Two agents receive time-varying endowments of a consumption good. The total endowment is always equal to 1, and the state s represents player 1's share of the endowment.[16] The state is i.i.d. uniform on an evenly spaced grid between 0 and 1. We let $e_i(s)$ denote player i's endowment, i.e., $e_1(s) = s$, $e_2(s) = 1 - s$. The agents can make transfers to one another in increments of $1/(M(|S| - 1))$, up to their own endowment. Player i's consumption is $c_i(a, s) = e_i(s) + a_j - a_i$. Flow utility is $g_i(a, s) = \sqrt{c_i(a, s)}$.

As is well-known, minimum equilibrium payoffs are attained in autarky, where players make no transfers and consume their endowments. The resulting payoffs are

$$\underline{v}_i(s) = (1 - \delta)\sqrt{e_i(s)} + \delta \frac{1}{|S|} \sum_{s' \in S} \sqrt{e_i(s')}.$$

There are more efficient equilibria, wherein the players use transfers to smooth consumption over time. Specifically, the set of feasible payoffs is the convex hull of the vectors $(\sqrt{c}, \sqrt{1 - c})$ for all feasible c.[17] For δ sufficiently large, the folk theorem says that it is possible to attain

[16] This version of the model is studied by Ljungqvist and Sargent (2004, Chapter 20).

[17] It is a special feature of this game that the set of possible flow payoffs is the same across all states, so that the set of feasible payoffs is just the convex hull of the flow payoffs.

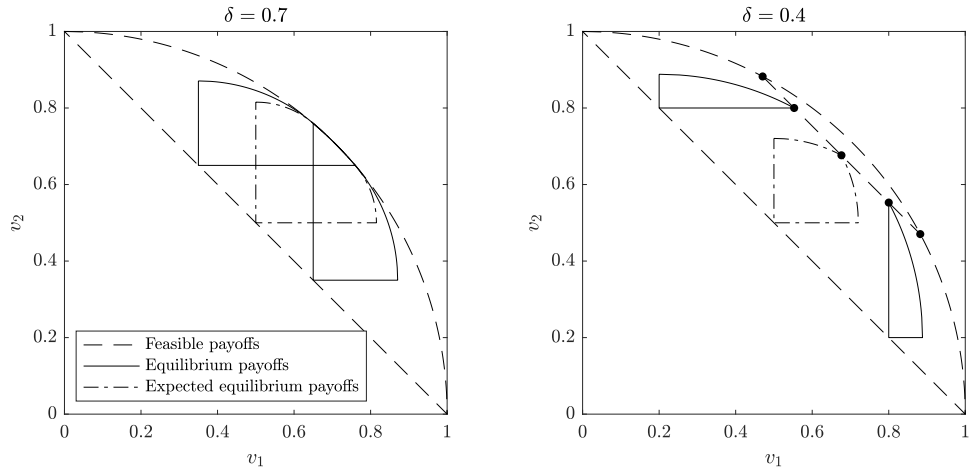

Figure 3: The two-state risk sharing game.

$\lvert S\rvert$	M	# Faces	MMM	JYC-100	JYC-200
2	20	27	1.5s	24.6s	78.3s
2	40	47	4.3s	1m 54.9s	5m 30.0s
5	20	102	1m 19.7s	9m 55.4s	30m 55.0s
9	15	145	6m 31.2s	44m 18.7s	2h 25m 46.4s

Table 1: Run times on the risk-sharing game for max-min-max and JYC with 100 and 200 evenly spaced directions.

payoffs on the frontier, provided that both players' insured payoff is at least that of autarky.

Our first example has two states, $M = 200$, and $\delta \in \{0.4, 0.7\}$. Equilibrium payoffs are depicted in Figure 3. We stopped iterating when the Hausdorff distance between $\widetilde{\mathbf{W}}^k$ and $\widetilde{\mathbf{W}}^{k-1}$ was less than 10^{-8}. At $\delta = 0.4$, the algorithm converged in 33 iterations and 16 seconds.[18] At $\delta = 0.7$, it converged in 36 iterations and 1 minute and 18 seconds.

At $\delta = 0.7$, full insurance can be supported at a range of consumption levels. This can be seen because the equilibrium payoff sets overlap with the frontier of feasible payoffs. The only way to generate these payoffs is to repeat the action profile that generates the given flow payoff forever. At the low discount factor ($\delta = 0.4$), equilibrium payoffs are bounded away from the feasible frontier, and efficient risk sharing cannot be supported.

We also computed equilibrium payoffs for a variety of $|S|$ and M, with $\delta = 0.7$ and a convergence threshold of 10^{-6}. The number of faces of $\mathbf{V}$ and run times are reported in Table 1. For comparison, we have also solved this game using the JYC algorithm, which approximates the APS operator in a fixed grid of directions using linear programming. This methodology is readily adapted to stochastic games. Our implementation uses the commercial linear programming software Gurobi, but otherwise it is integrated into the rest of our

[18] All benchmarks were measured on a mid-2014 MacBook Pro.

program and uses the same data structures as our implementation of the max-min-max algorithm.[19] Columns JYC-100 and JYC-200 in Table 1 report runtimes for our implementation of the JYC algorithm with 100 and 200 fixed directions, respectively.

We find that JYC takes between one and two orders of magnitude longer than the max-min-max operator. For example, when there are 5 states and $M = 20$ (which corresponds to 81 consumption levels), max-min-max takes 1 minute and 20 seconds, while JYC with 100 directions takes 10 minutes, and with 200 directions takes 31 minutes. Note that these algorithms produce different outputs: the max-min-max limit has faces which approximate those of $\mathbf{V}$, whereas the JYC limit bounds payoffs in exogenous directions that are unrelated to faces of the equilibrium payoff correspondence.

We should be cautious in drawing general conclusions from these simulations: Run times will vary from game to game, implementation to implementation, and computer to computer. In many applications, it may not be mission critical whether the algorithm terminates in three minutes or in three days. Nonetheless, these simulations strongly suggest that our algorithm will provide faster and more accurate solutions than other known methods.

5 Implementation with many players

5.1 The complexity of V

We now return to the general setting with many players. A critical complication is that the number of extreme equilibrium payoffs is no longer bounded. Indeed, we will show that the following three-player repeated game has countably infinitely many extreme equilibrium payoffs.

The flow payoffs are depicted in Figure 4, where x is a very low negative payoff, and $\delta = 1/2$. This game has twenty-seven action profiles, but only four can be played in equilibrium. Specifically, (A, A, A) is a static Nash equilibrium, and the permutations of (C, B, B) can be sustained when the discount factor is sufficiently large (in particular when $\delta = 1/2$). Since each player can guarantee themselves a payoff of 0 by always playing A, no other action profile can be played in equilibrium as long as x is sufficiently low.

The equilibrium payoff set V is depicted in the left-hand panel of Figure 5. Online Appendix A contains detailed analysis of the game. We now present an informal overview.

[19] Our methodology uses the optimal solution in one direction as a starting point for computing solutions in adjacent directions, thereby accelerating convergence to a solution. JYC emphasize the separability of the linear programs for each action profile and direction, and their original Fortran implementation starts each computation from scratch. Our implementation of JYC starts each optimization from the adjacent solution, which considerably speeds up the computation.

	$a_3 = A$			$a_3 = B$			$a_3 = C$		
a_1/a_2	A	B	C	A	B	C	A	B	C
A	$(4,4,4)$	$(0,x,0)$	$(0,x,0)$	$(0,0,x)$	$(3,x,x)$	$(0,x,x)$	$(0,0,x)$	$(0,0,x)$	$(0,0,x)$
B	$(x,0,0)$	$(x,x,3)$	$(x,x,0)$	$(x,3,x)$	(x,x,x)	$(8,0,8)$	$(x,0,x)$	$(8,8,0)$	(x,x,x)
C	$(x,0,0)$	$(x,x,0)$	$(x,x,0)$	$(x,0,x)$	$(0,8,8)$	(x,x,x)	$(x,0,x)$	(x,x,x)	(x,x,x)

Figure 4: A three player game.

Since only four action profiles can be played in equilibrium, we know that V must be contained in the triangular pyramid with corners at $(4,4,4)$ and the permutations of $(0,8,8)$. It includes $(4,4,4)$ and a large flat on the efficient frontier where $v_1 + v_2 + v_3 = 16$. It is easy to show that the minimum equilibrium payoff is 3. For each $i = 1,2,3$, there is a face D_i where player i's payoff is minimized. Besides the Nash payoff, all of the extreme points of V lie in one of these flats. The remaining faces are triangles with the Nash payoff as one vertex.

Let us focus on D_3, which is depicted in the right panel of Figure 5. The set of extreme points of D_3 has two accumulation points, each of which is approached by two sequences. One sequence starts on the efficient frontier, at points denoted u and u', and moves down, and the other starts at inefficient payoffs, denoted v and v', and moves up. All of these payoffs are generated by playing (B,B,C) for one period, followed by a continuation payoff in $C(B,B,C)$.

The set $C(B,B,C)$ is generated as follows. The height of the blue plane in the left-hand panel of Figure 5 is the continuation value at which player 3's incentive constraint binds, i.e., $v_3 = 6$. For each D_k with $k = 1,2$, exactly one of the accumulation points and its corresponding sequences lies above the plane. For each element $\widehat{v}$ of these sequences, there is a corresponding extreme point of $C(B,B,C)$, where the line between $\widehat{v}$ and $(4,4,4)$ crosses $v_3 = 6$. In addition, there are four more payoffs in $C(B,B,C)$, two of which are generated by randomizing between u and u', and two of which are generated by randomizing between v and v'. Thus, half of the sequences in D_3 are generated from sequences in each D_k, for $k = 1,2$.

We note that the same basic structure obtains if we perturb the players' payoffs, and it seems to be a generic possibility that the number of extreme equilibrium payoffs is infinite. As an aside, the development of this example illustrates the potential value of computational methods in repeated games. Only after being inspired by numerical results were we able to construct the example analytically.

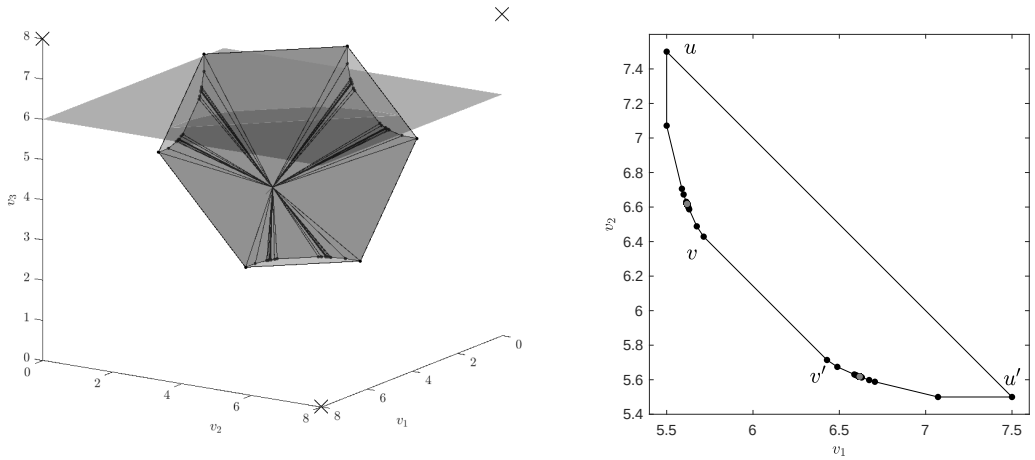

Figure 5: Left: Equilibrium payoffs set for the game in Figure 4 with $\delta = 1/2$, looking down on the Pareto frontier. Right: equilibrium payoffs in which $v_3 = 3$.

5.2 Implementation and approximation with $N \geq 3$

Given that $\mathbf{V}$ may have infinitely many extreme points, exact computation of the sequence from (T1.iv) may be impossible. We now describe a procedure that can compute $\widetilde{B}$ exactly when the number of extreme points is small and approximates it when the number of extreme points blows up.

At every iteration, there will be a set of directions $\widehat{\Lambda}^k$, with $\widehat{\Lambda}^0$ being arbitrary, and we compute the new correspondence $\widehat{\mathbf{W}}^{k+1} = \widetilde{B}(\widehat{\mathbf{W}}^k, \widehat{\Lambda}^k)$, where

$$\widetilde{B}(\mathbf{W}, \widehat{\Lambda})(s) = \{v | \lambda \cdot v \leq x(s, \lambda, \mathbf{W}) \ \forall \lambda \in \widehat{\Lambda}\}.$$

Thus, $\widetilde{B}(\cdot, \widehat{\Lambda})$ is analogous to $\widetilde{B}$, but where we only bound payoffs for directions in $\widehat{\Lambda}$. For future reference, we similarly define

$$B(\mathbf{W}, \widehat{\Lambda})(s) = \{\lambda \cdot v \leq x^{APS}(s, \lambda, \mathbf{W}) \text{ for all } \lambda \in \widehat{\Lambda}\}.$$

We will fully specify the procedure for updating $\widehat{\Lambda}^k$ shortly. At a high level, we will drop directions that are redundant and add new directions that correspond to faces of $\widetilde{B}(\widehat{\mathbf{W}}^k)$, in a manner analogous to what we did for two players. We will, however, cap the size of $\widehat{\Lambda}^k$ so that the complexity is bounded. Note that for any sequence $\{\widehat{\Lambda}^k\}$, each correspondence $\widehat{\mathbf{W}}^k$ must contain $\widetilde{\mathbf{W}}^k$, and thus $\mathbf{V} \subseteq \cap_{k \geq 0} \widehat{\mathbf{W}}^k$. By updating the directions endogenously, the sequence $\widehat{\mathbf{W}}^k$ will converge to $\mathbf{V}$ if memory permits, and otherwise we coarsen the approximation to satisfy the complexity bound.

It is critical, however, to choose $\widehat{\Lambda}^k$ so that we retain a key computational advantage of the max-min-max operator, which is the sufficiency of binding payoffs. When $\widehat{\Lambda}^k = \Lambda$, this was established by Lemmas 5 and 6. (Recall that Λ is the set of all directions.) These results still apply, but the following weaker formulation of Lemma 6 will be useful:

Lemma 16. *For any $\widehat{\Lambda} \subseteq \Lambda$, if $\widetilde{B}(\mathbf{W}, \widehat{\Lambda}) \subseteq \mathbf{W}$, then $B\left(\widehat{\mathbf{W}}\right) \subseteq \widehat{\mathbf{W}}$, where $\widehat{\mathbf{W}} = \widetilde{B}(\mathbf{W}, \widehat{\Lambda})$.*

Proof of Lemma 16. The proof of Lemma 6 directly implies that $B(\widehat{\mathbf{W}}, \widehat{\Lambda}) \subseteq \widehat{\mathbf{W}}$. Since B is decreasing in its second argument, the weaker conclusion of the lemma follows. □

By Lemma 5, if $B(\mathbf{W}, \widehat{\Lambda}) \subseteq \mathbf{W}$, then a positive APS gap in some direction implies that the minimal regime can be taken to be recursive. As a consequence of Lemma 16, as long as we choose $\widehat{\Lambda}^k$ such that $\widetilde{B}(\widehat{\mathbf{W}}^k, \widehat{\Lambda}^k) \subseteq \widetilde{B}(\widehat{\mathbf{W}}^{k-1}, \widehat{\Lambda}^{k-1}) = \widehat{\mathbf{W}}^k$, the inductive hypothesis that B sub-generates will be satisfied, so that binding payoffs are sufficient and we need only compute binding APS payoffs and the local frontier around those payoffs.

In fact, we can go a step further. Recall that the computation of $\widetilde{B}(\mathbf{W}, \widehat{\Lambda})$ only depends on the binding APS level $\widehat{x}^{APS}(s, \lambda, \mathbf{W})$ and the sign of $\gamma(a, \lambda, \mathbf{W})$. If these two functions are the same for correspondences $\mathbf{W}$ and $\mathbf{W}'$, then we say that they have the same *local binding frontier*. Now, suppose that $\widehat{\Lambda}^k$ is such that the correspondence $\widetilde{B}(\widehat{\mathbf{W}}^{k-1}, \widehat{\Lambda}^k)$ has the same local binding frontier as $\widehat{\mathbf{W}}^k$. Then $\widetilde{B}(\widehat{\mathbf{W}}^{k-1}, \widehat{\Lambda}^k)$ need not be a subset of $\widetilde{B}(\widehat{\mathbf{W}}^{k-1}, \widehat{\Lambda}^{k-1})$. But as long as we continue to use $\widetilde{B}$ in subsequent iterations, the resulting computations will be *exactly the same*:

Lemma 17. *Suppose $\mathbf{W} \subseteq \mathbf{W}'$, both correspondences have the same local binding frontier, and B sub-generates at $\mathbf{W}$. Then for all s, λ, and $\mathbf{a} \in \mathbf{A}(\mathbf{W})$, $x(s, \lambda, \mathbf{a}, \mathbf{W}) = x(s, \lambda, \mathbf{a}, \mathbf{W}')$.*

Proof of Lemma 17. Since B sub-generates at $\mathbf{W}$, the hypothesis of Lemma 5 is satisfied. Let us then take $\mathbf{r}$ to be minimal regimes for $\mathbf{a}$ at $\mathbf{W}$ in the direction λ that satisfy the refinement of Lemma 5, i.e., the regime is recursive whenever the APS gap is strictly positive. Since $\mathbf{W}$ and $\mathbf{W}'$ have the same sign APS gap and the same binding payoffs, we conclude that

$$x(s, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}) = x(s, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}'). \tag{16}$$

To verify that $\mathbf{r}$ is minimal for $\mathbf{a}$ at $\mathbf{W}'$, we need only check the minimality conditions (6). Equation (16) implies that

$$x^R(\mathbf{a}(s), \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}) = x^R(\mathbf{a}(s), \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}'). \tag{17}$$

When $\mathbf{r}(s) = APS$, the optimal APS level is binding, and by hypothesis this level is the same for $\mathbf{W}$ and $\mathbf{W}'$, so that both sides of (6) are unchanged. When $\mathbf{r}(s) = R$, (6) implies

that $x^R(\mathbf{a}(s), \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}) \leq x^{APS}(\mathbf{a}(s), \lambda, \mathbf{W})$. As x^{APS} is increasing in its third argument, this implies that $x^R(\mathbf{a}(s), \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}) \leq x^{APS}(\mathbf{a}(s), \lambda, \mathbf{W}')$. Combining this last inequality with (17), we conclude that $\mathbf{r}$ satisfies (6) at $\mathbf{W}'$ as well. □

In addition, if we *add* a new direction to $\widehat{\Lambda}^k$ that was not present in $\widehat{\Lambda}^{k-1}$, then this will only cause $\widehat{\mathbf{W}}^k$ to shrink further, so that the hypothesis for binding payoffs to be sufficient will still be satisfied:

Theorem 3. *Suppose the sequence* $\{\widehat{\Lambda}^k\}_{k\geq 0}$ *is such that for every* $k \geq 1$, $\overline{\mathbf{W}}^k = \widetilde{B}(\widehat{\mathbf{W}}^{k-1}, \widehat{\Lambda}^k \cap \widehat{\Lambda}^{k-1})$ *has the same local binding frontier as* $\widehat{\mathbf{W}}^k = \widetilde{B}(\widehat{\mathbf{W}}^{k-1}, \widehat{\Lambda}^{k-1})$*, and assume* $B(\widehat{\mathbf{W}}^0, \widehat{\Lambda}^0) \subseteq \widehat{\mathbf{W}}^0$*. Then for every* $k \geq 0$*,* $B(\widehat{\mathbf{W}}^k) \subseteq \widehat{\mathbf{W}}^k$*. As a result, for any* $\mathbf{a} \in \mathbf{A}(\widehat{\mathbf{W}}^k)$*, there exist minimal regimes such that* $\mathbf{r}(s) = R$ *whenever* $\gamma(\mathbf{a}(s), \lambda, \widehat{\mathbf{W}}^k) > 0$*.*

Proof of Theorem 3. We will argue that for every k, $B(\widehat{\mathbf{W}}^k) \subseteq \widehat{\mathbf{W}}^k$. The second part of the theorem then follows directly from Lemma 5.

Take as inductive hypotheses that (i) $\widetilde{B}(\overline{\mathbf{W}}^{k-1}, \widehat{\Lambda}^{k-1}) \subseteq \overline{\mathbf{W}}^{k-1}$ and (ii) $B(\widehat{\mathbf{W}}^{k-1}) \subseteq \widehat{\mathbf{W}}^{k-1}$. We set $\overline{\mathbf{W}}^0 = \widehat{\mathbf{W}}^0$ so that these hypotheses are true for the base case $k = 1$. Also note for current and later use that $\widetilde{B}(\cdot, \cdot)$ is increasing in its first argument and decreasing in its second, as is $B(\cdot, \cdot)$. As a result, $\widehat{\mathbf{W}}^{k-1} \subseteq \overline{\mathbf{W}}^{k-1}$. The assumption that $\overline{\mathbf{W}}^j$ and $\widehat{\mathbf{W}}^j$ have the same local binding frontier for every $j \geq 1$ is used freely below. Together with the inductive hypothesis (ii), Lemma 17 then implies that $\widehat{\mathbf{W}}^{k-1}$ and $\overline{\mathbf{W}}^{k-1}$ are interchangeable in the $\widetilde{B}$ operator. Consequently,

$$\widetilde{B}(\overline{\mathbf{W}}^{k-1}, \widehat{\Lambda}^{k-1}) = \widetilde{B}(\widehat{\mathbf{W}}^{k-1}, \widehat{\Lambda}^{k-1}) = \widehat{\mathbf{W}}^k. \tag{18}$$

By the inductive hypothesis (i) and Lemma 16 (where $\widehat{\Lambda} = \widehat{\Lambda}^{k-1}$, $\mathbf{W} = \overline{\mathbf{W}}^{k-1}$, and $\mathbf{W}' = \widehat{\mathbf{W}}^k$), it follows that $B(\widehat{\mathbf{W}}^k) \subseteq \widehat{\mathbf{W}}^k$, thus extending the inductive hypothesis (ii) to k.

Lemma 17 then implies that $\widehat{\mathbf{W}}^k$ and $\overline{\mathbf{W}}^k$ are also interchangeable in $\widetilde{B}$. Also, (i) and interchangeability of $\widehat{\mathbf{W}}^{k-1}$ and $\overline{\mathbf{W}}^{k-1}$ imply that $\widehat{\mathbf{W}}^k = \widetilde{B}(\widehat{\mathbf{W}}^{k-1}, \widehat{\Lambda}^{K-1}) = \widetilde{B}(\overline{\mathbf{W}}^{k-1}, \widehat{\Lambda}^{K-1}) \subseteq \overline{\mathbf{W}}^{k-1}$. Hence,

$$\begin{aligned}
\widetilde{B}(\overline{\mathbf{W}}^k, \widehat{\Lambda}^k) &\subseteq \widetilde{B}(\overline{\mathbf{W}}^k, \widehat{\Lambda}^k \cap \widehat{\Lambda}^{k-1}) \\
&= \widetilde{B}(\widehat{\mathbf{W}}^k, \widehat{\Lambda}^k \cap \widehat{\Lambda}^{k-1}) \\
&\subseteq \widetilde{B}(\overline{\mathbf{W}}^{k-1}, \widehat{\Lambda}^k \cap \widehat{\Lambda}^{k-1}) \\
&= \widetilde{B}(\widehat{\mathbf{W}}^{k-1}, \widehat{\Lambda}^k \cap \widehat{\Lambda}^{k-1}) = \overline{\mathbf{W}}^k,
\end{aligned}$$

where we have used, in order, monotonicity of $\widetilde{B}$ in its second argument, interchangeability of $\overline{\mathbf{W}}^k$ and $\widehat{\mathbf{W}}^k$, monotonicity of $\widetilde{B}$ in its first argument, interchangeability of $\overline{\mathbf{W}}^{k-1}$ and

$\widehat{\mathbf{W}}^{k-1}$, and the definition of $\overline{\mathbf{W}}^k$. This extends the inductive hypothesis (i) to k. $\square$

Having established conditions on $\{\widehat{\Lambda}^k\}$ for binding payoffs to be sufficient, we now fill in the remaining details of the algorithm. Fix a positive integer $L > 0$, which is the maximum number of directions. At every iteration, we construct $\widehat{\Lambda}^k$ by first dropping directions in a set Λ' such that $\widetilde{B}(\widehat{\mathbf{W}}^{k-1}, \widehat{\Lambda}^{k-1})$ and $\widetilde{B}(\widehat{\mathbf{W}}^{k-1}, \widehat{\Lambda}^{k-1} \setminus \Lambda')$ have the same local binding frontier. This is in fact easy to do: in the process of computing $C(a, \widetilde{\mathbf{W}}^{k-1})$, we intersect the binding payoff sets with the half-spaces that define the expected continuation value set $\sum_{s \in S} \pi(s|a)\widehat{\mathbf{W}}^k(s)$. There is some order in which the directions are intersected. If we find that the half space $(\lambda, x(s, \lambda, \widehat{\mathbf{W}}^{k-1})$ does not result in a change to $C(a, \mathbf{W})$ or the local frontier around those payoffs for any a, then the direction is redundant and can be dropped.[20]

After dropping directions, we can add new directions as long as the total number is less than L. The directions we add are faces of $\widetilde{B}(\widehat{\mathbf{W}}^{k-1})$, i.e., directions for which there are N different optimal payoffs which are not contained in an affine subspace of dimension less than $N-1$. This is the natural generalization of test directions from the two-player case.

To compute a face direction, we first pick an initial direction λ^0 uniformly in Λ and compute optimal payoffs $\mathbf{u}$, which with probability one are uniquely optimal (and hence remain optimal for perturbations in all directions). We then draw a direction of rotation $\widetilde{\lambda}^1$ uniformly on Λ and compute the legitimate test direction with the smallest rotation from λ^0 in the direction $\widetilde{\lambda}^1$, which is denoted λ^1. This step is analogous to that described in Section 4. The corresponding substitution is (s^1, a^1, p^1), and let $d^1 = u(s^1, a^1, p^1, \mathbf{u}) - \mathbf{u}(s^1)$ be the direction in which the substitution moves payoffs. We then pick a new direction $\widetilde{\lambda}^2$ uniformly on the set of directions which are orthogonal to d^1, and compute the shallowest legitimate test direction from λ^0 in the direction $\widetilde{\lambda}^2$, denoted λ^2, with substitution (s^2, a^2, r^2). Continuing inductively, after n steps, we will have a direction λ^n and n substitutions which move payoffs in linearly-independent directions $d^1, \ldots, d^n$ that are orthogonal to λ^n. After $N-1$ steps, the process converges to a face direction λ^{N-1}. Pseudocode for the face-finding procedure is in Algorithms 8 and 9.

We run this procedure $L - |\widehat{\Lambda}^{k-1} \setminus \Lambda'|$ times and add the new face directions (skipping duplicates) to $\widehat{\Lambda}^{k-1}$ to obtain $\widehat{\Lambda}^k$. (Note that L may be greater than the number of face directions, in which case this procedure necessarily encounters duplicates.) Algorithm 10 gives pseudocode for this step. In practice, we have found it better to generate new directions once every five iterations or so, while redundant directions are dropped every iteration. This completes the many-player algorithm.

[20] Our code randomizes the order in which the half spaces are intersected, in a crude effort to identify a non-redundant set of directions. This process could in principle be systematized using standard convex hull algorithms.

We may compare this procedure with that of JYC. Both algorithms bound payoffs in a finite number of directions. While JYC use the APS level for each direction, we use the max-min-max level. The difference is significant, since there are many fewer payoffs that can be optimal for max-min-max, and we solve out the optimal payoffs when the regime is recursive. In addition, JYC hold the directions fixed, whereas our procedure adjusts the directions dynamically to achieve a sharper approximation.

We have implemented this procedure as part of our software package. Online Appendix B describes two numerical examples. The first is a simple three-player binary-action contribution game, in which the equilibrium payoff set has a small number of faces, to which the algorithm converges exactly. The second example is a three-player version of the risk-sharing game considered in Section 4. We use our algorithm to illustrate how partial contracts, in which two of the three players commit to perfectly insure one another, can lead to lower welfare for all parties, as the partial contract limits the players' ability to punish deviations.

The procedure we implemented is just one of many possible ways to generate a sequence $(\widetilde{\mathbf{W}}^k, \widehat{\Lambda}^k)$ that satisfies the hypotheses of Theorem 3. For example, we could take $\widetilde{\mathbf{W}}^0$ to be the feasible payoff correspondence, or to be large hypercubes that contain all flow payoffs (as we have done in our simulations). In either case, the initial correspondence has finitely many extreme points, so that $\widetilde{B}(\mathbf{W}^0)$ is guaranteed to have finitely many faces. As long as L is sufficiently large, we can set $\widehat{\Lambda}^0 = \Lambda$, i.e., all directions, and compute $\widetilde{B}(\mathbf{W}^0)$ exactly. This can be done via a generalization of the face-finding procedure, whereby we recursively map faces that are adjacent to one another via successive one-dimensional searches. This precision could be maintained as long as the number of face directions is less than L. One could even use the above stochastic algorithm temporarily, while the number of face directions is large, but reverts to exact computation when the payoff sets simplify. We have not attempted to explore all of these possibilities, and they are promising directions for future research.

6 Lower bounds on V

Like that of APS, our methodology generates a sequence that converges to $\mathbf{V}$ from the "outside," meaning that every element is a superset of $\mathbf{V}$. This sequence may only converge asymptotically, so that if we stop iterating after finitely many rounds, we only obtain an upper bound on $\mathbf{V}$. As a final topic, we adapt our methodology to compute a corresponding lower bound.

For a fixed $\epsilon > 0$, consider the following perturbed APS operator: $B^\epsilon(\mathbf{W})(s) = \{v | \lambda \cdot v \leq x^{APS}(s, \lambda, \mathbf{W}) - \epsilon \ \forall \lambda \in \Lambda\}$. Just like the APS operator, B^ϵ takes compact correspondences to compact correspondences and is increasing in $\mathbf{W}$. As a result, there is a largest bounded

fixed point, denoted $\mathbf{V}^\epsilon$, which can be computed by iterative application of B^ϵ on any correspondence that contains $\mathbf{V}^\epsilon$. Moreover, B^ϵ is decreasing in ϵ, which implies that $\mathbf{V}^\epsilon$ is decreasing in ϵ, so that $\mathbf{V}^\epsilon \subseteq \mathbf{V}^0 = \mathbf{V}$.

We propose to compute a lower bound on $\mathbf{V}$ by generating a sequence that converges to $\mathbf{V}^\epsilon$. At first glance, this plan seems to have the same problem: How do we know when we have converged to $\mathbf{V}^\epsilon$? The difference is that we do not want to compute $\mathbf{V}^\epsilon$; we just want to find a set that self generates. In particular, since the iterates of B^ϵ converge to $\mathbf{V}^\epsilon$, eventually we will obtain a correspondence $\mathbf{W}$ such that the Hausdorff distance between $\mathbf{W}$ and $B^\epsilon(\mathbf{W})$ is less than $\epsilon/2$. As a result, $B(\mathbf{W})$ will be strictly larger than $\mathbf{W}$ by at least $\epsilon/2$ in every direction, so that we can robustly certify that $B(\mathbf{W}) \subseteq \mathbf{V}$. Note that we have no guarantee that $\mathbf{V}^\epsilon$ is close to $\mathbf{V}$, or even non-empty valued. In simulations discussed below, however, the lower bound we obtain seems to converge to $\mathbf{V}$ as ϵ goes to zero.[21]

At the same time, a premise of this paper is that B is hard to compute, which is why we developed the operator $\widetilde{B}$, and the same computational difficulties arise when computing B^ϵ. Fortunately, our methodology can be adapted to compute $\mathbf{V}^\epsilon$. Given a policy $(\mathbf{a}, \mathbf{r})$, we define $x^\epsilon(s, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W})$ to be the unique solution to

$$x^\epsilon(s, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}) = -\epsilon + \begin{cases} (1-\delta)\lambda \cdot g(\mathbf{a}(s)) + \delta \sum_{s' \in S} \pi(s'|\mathbf{a}(s)) x^\epsilon(s', \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W}) & \text{if } \mathbf{r}(s) = R; \\ x^{APS}(\mathbf{a}(s), \lambda, \mathbf{W}) & \text{if } \mathbf{r}(s) = APS. \end{cases}$$

We define $x^\epsilon(s, \lambda, \mathbf{W}) = \max_{\mathbf{a} \in \mathbf{A}(\mathbf{W})} \min_{\mathbf{r} \in \mathbf{R}} x^\epsilon(s, \lambda, \mathbf{a}, \mathbf{r}, \mathbf{W})$ and $\widetilde{B}^\epsilon(\mathbf{W}) = \{v | \lambda \cdot v \leq x^\epsilon(s, \lambda, \mathbf{W}) \, \forall \lambda \in \Lambda\}$. The operator $\widetilde{B}^\epsilon$ satisfies all the desirable properties of $\widetilde{B}$. In particular, $\widetilde{B}^\epsilon$ is increasing, maps compact correspondences to compact correspondences, and has $\mathbf{V}^\epsilon$ as a fixed point. As a result, if we fix a correspondence $\widetilde{\mathbf{W}}^0$ that contains $\mathbf{V}^\epsilon$ and generate the sequence $\widetilde{\mathbf{W}}^k = \widetilde{B}^\epsilon(\mathbf{W}^{k-1})$, then $\mathbf{V}^\epsilon = \cap_{k=0}^\infty \widetilde{\mathbf{W}}^k$. In fact, we can even take $\widetilde{\mathbf{W}}^0$ to be the upper bound on $\mathbf{V}$ obtained by iterative application of $\widetilde{B}$.

This result is reported as Theorem 1 in Supplemental Appendix E, where we rederive all of our key results, adding in ϵ's where appropriate. All of our other results extend as well: For every direction, there exists a state-independent optimal policy. Binding payoffs are sufficient as long as B^ϵ sub-generates, and B^ϵ will sub-generate along the sequence we compute as long as it sub-generates at the first round.[22] There is a simple set of conditions that characterize

[21] We have good reason to think that $\mathbf{V}^\epsilon$ will not collapse in general. For example, if we were to drop incentive constraints, then the analogue of $\mathbf{V}^\epsilon$ is simply the contraction of the feasible payoff correspondence in every direction by $\epsilon/(1-\delta)$.

[22] For this result, it is essential that the ϵ penalty is recursively compounded in the definition of x^ϵ. If we simply added an ϵ penalty in both regimes, the resulting operator *would* result in a sequence that converges to $\mathbf{V}^\epsilon$, but the penalty attached to recursive payoffs would be too small, so that eventually, the minimal regime would always be APS.

optimal policies and optimal pairs. When there are two players, the correspondence $\mathbf{V}^\epsilon$ has at most $\overline{L}$ extreme payoffs, and $\widetilde{B}^\epsilon$ can be computed in runtime $O(\overline{L}\,\overline{M}^2)$.[23]

We have implemented the operator $\widetilde{B}^\epsilon$ for two players as part of our software package. In Online Appendix B, we report an application to the risk-sharing example of Section 4, for which the lower bound and upper bound are virtually indistinguishable. Beyond two players, further approximation of $\widetilde{B}^\epsilon$ may be needed, as discussed in Section 5.

7 Conclusion

It has been our purpose to study the subgame perfect equilibria of stochastic games. We have developed a new fundamental structural property of extremal equilibria, namely that equilibrium play is stationary until incentive constraints bind. We developed a new "max-min-max" algorithm that exploits this structure, using policy iteration when incentive constraints are slack to obtain tighter bounds than the APS operator on which payoffs can be generated. The bounds can also be computed using only knowledge of binding payoffs and the slope of the frontier around those payoffs. Moreover, the optimal equilibrium structure changes in only one state at a time as the direction of optimization moves, which greatly simplifies the computation of the set of payoffs that can be generated. When there are two players, the resulting algorithm, and by extension the equilibrium payoff correspondence, are of bounded complexity. We have shown by example that the number of extreme equilibrium payoffs may be infinite with more than two players, but we have provided a flexible routine that can approximate equilibrium payoffs when computing power is limited and compute them exactly when the equilibrium correspondence is not too complicated.

The insights that we have developed are obviously particular to the special class of games we have considered. We have made heavy use of perfect monitoring, public randomization, and the restriction to pure-strategy equilibria. These assumptions are widely used both in theory and application and, in our view, are eminently worthy of study. While the basic results on the max-min-max operator can be extended, these particular insights will presumably be more or less useful for computation depending on the class of game being studied. At a broader level, our approach is to develop methods that are tailored to the special structure that arises in extremal equilibria. It is our hope that similar efforts will bear fruit for other classes of games and solution concepts, for example, those involving imperfect monitoring or mixed strategies.

[23] A subtlety here is that the function x^ϵ is no longer linear in λ. Nonetheless, as we argue in Supplemental Appendix E, $\widetilde{B}^\epsilon$ can be computed by intersecting bounds at test directions.

References

Abreu, D., B. Brooks, and Y. Sannikov (2016): "A "pencil-sharpening" algorithm for two-player stochastic games with perfect monitoring," Tech. rep., New York University and University of Chicago and Stanford University.

Abreu, D., D. Pearce, and E. Stacchetti (1986): "Optimal cartel equilibria with imperfect monitoring," *Journal of Economic Theory*, 39, 251–269.

——— (1990): "Toward a theory of discounted repeated games with imperfect monitoring," *Econometrica*, 58, 1041–1063.

Abreu, D. and Y. Sannikov (2014): "An algorithm for two-player repeated games with perfect monitoring," *Theoretical Economics*, 9, 313–338.

Atkeson, A. (1991): "International lending with moral hazard and risk of repudiation," *Econometrica: Journal of the Econometric Society*, 1069–1089.

Berg, K. (2019): "Set-valued games and mixed-strategy equilibria in discounted supergames," *Discrete Applied Mathematics*, 255, 1–14.

Berg, K. and M. Kitti (2019): "Equilibrium paths in discounted supergames," *Discrete Applied Mathematics*.

Blackwell, D. (1965): "Discounted dynamic programming," *The Annals of Mathematical Statistics*, 226–235.

Dixit, A., G. M. Grossman, and F. Gul (2000): "The dynamics of political compromise," *Journal of political economy*, 108, 531–568.

Ericson, R. and A. Pakes (1995): "Markov-perfect industry dynamics: A framework for empirical work," *The Review of Economic Studies*, 62, 53–82.

Hörner, J., T. Sugaya, S. Takahashi, and N. Vieille (2011): "Recursive methods in discounted stochastic games: An algorithm for $\delta \to 1$ and a folk theorem," *Econometrica*, 79, 1277–1318.

Judd, K. L., S. Yeltekin, and J. Conklin (2003): "Computing supergame equilibria," *Econometrica*, 71, 1239–1254.

Kocherlakota, N. R. (1996): "Implications of efficient risk sharing without commitment," *The Review of Economic Studies*, 63, 595–609.

Ljungqvist, L. and T. J. Sargent (2004): *Recursive macroeconomic theory*, MIT press.

Mailath, G. J. and L. Samuelson (2006): "Repeated games and reputations: long-run relationships," *OUP Catalogue*.

Pakes, A. and P. McGuire (1994): "Computing Markov-Perfect Nash Equilibria: Numerical Implications of a Dynamic Differentiated Product Model," *RAND Journal of Economics*, 25, 555–589.

Phelan, C. and E. Stacchetti (2001): "Sequential equilibria in a Ramsey tax model," *Econometrica*, 69, 1491–1518.

Renner, P. and S. Scheidegger (2018): "Machine learning for dynamic incentive problems," Tech. rep., Lancaster University and University of Lausanne.

Sleet, C. and Ş. Yeltekin (2016): "On the computation of value correspondences for dynamic games," *Dynamic Games and Applications*, 6, 174–186.

Yeltekin, Ş., Y. Cai, and K. L. Judd (2017): "Computing equilibria of dynamic games," *Operations Research*, 65, 337–356.