Scalable Equilibrium Computation in n-Player General-Sum Games

Candidate Number 1072036

Submitted in partial fulfilment of the requirements for the degree of

Master of Mathematics and Computer Science

Word count: 9965

Trinity Term 2026

Abstract

We will focus on the study of scalable equilibrium computation for n-player general-sum extensive-form games. Although correlated and coarse correlated equilibria (CE and CCE) admit polynomial-time linear programs in the number of joint actions, the joint action space scales exponentially in the number of players, and meta-game frameworks such as JPSRO inherit the same blow-up at the level of joint policies once the policy pool grows across iterations.

We introduce JPSRO-CG, a meta-solver that embeds column generation into the JPSRO outer loop, maintaining a small, dynamically chosen support of joint policies warm-started from the previous iteration, with a two-stage welfare phase that targets welfare-optimal equilibria without ever instantiating the full-support LP. On the theoretical side, we prove polynomial upper bounds on the minimum support of an optimal (C)CE under any linear objective ($2 + \sum_i |\mathcal{A}_i|(|\mathcal{A}_i| - 1)$ for CE and $2 + \sum_i |\mathcal{A}_i|$ for CCE), and generalise JPSRO's convergence theorem to show that an exact best-response oracle combined with an ϵ-(C)CE meta-solver converges to an ϵ-(C)CE of the underlying extensive-form game, recovering the original guarantee at $\epsilon = 0$.

Empirically, on Kuhn Poker (3 and 4 players), Sheriff, Trade Comm, and Tiny Bridge, JPSRO-CG matches JPSRO's exploitability trajectory on a much smaller support, converges to a CCE on 4-player Kuhn Poker within a time budget on which the full-LP variant does not, and recovers the welfare-optimal CE on Sheriff and Trade Comm. Under the CCE welfare objective it plateaus at a strictly worse value than full-LP JPSRO, since concentrating mass on extremal vertices removes the meta-distribution diversity that the marginalised CCE best response relies on.

Contents

Abstract **1**

1 Introduction **4**

1.1 Motivation 4

1.2 Problem Statement 5

1.3 Contributions 6

1.4 Outline 7

2 Background and Preliminaries **7**

2.1 Correlated Equilibrium and Coarse Correlated Equilibrium 7

2.2 Approximate (C)CE 8

2.3 Column Generation 9

2.4 Previous Work 11

2.4.1 Joint policy space response oracles 11

3 Equilibrium Generation **13**

3.1 Formulating the column generation 13

3.2 Equilibrium Optimisation 14

4 JPSRO-CG **15**

4.1 Theoretical Bounds and Convergence 16

4.2 Algorithm 20

4.3 Empirical Setup 25

4.3.1 Games choice 25

4.3.2 Method of the experiment 25

5 Results **27**

5.1 Recreation of experiments 27

5.2 Convergence to equilibrium 28
5.3 Convergence to optimal equilibrium 29

6 Discussion **32**
6.1 Support growth and pricing hardness 32
6.2 Limitations of column generation on the full game 32
6.3 Exploration versus exploitation in the training meta-solver 33
6.4 The equilibrium selection problem 35
6.5 Improvements 35
6.5.1 Stabilising the meta-distribution 35
6.5.2 Approximate best-response oracles 36
6.5.3 Further directions 37

7 Conclusion **38**

A Notation **44**

B Experimental Setup Details **46**
B.1 Game parameters 46
B.2 Solver and algorithm parameters 47
B.3 Compute resources 48
B.4 Experiment matrix 49
B.5 Logged metrics and output format 49

C Results **51**

D Pricing in Graphical Games **59**

E Support is not confined to old support plus new joint actions **60**

1 Introduction

1.1 Motivation

The study of multi-agent interactions and strategic decision-making has historically tended to focus on two-player, zero-sum games. In these strictly competitive environments, the mathematical structures are highly constrained, ensuring that Nash Equilibria (NE) are tractable to compute and exhibit a property known as interchangeability. In a two-player zero-sum game, if both players independently adopt any valid Nash Equilibrium strategy, the resulting joint policy profile remains an equilibrium, and performance guarantees are maintained. This foundational property has enabled algorithms based on linear programming and counterfactual regret minimisation to achieve strong performance in classic benchmark domains.

However, translating these successes to general-sum, n-player extensive-form games represents one of the most challenging frontiers in artificial intelligence and operations research. The real world is rarely strictly zero-sum, and interactions typically involve more than two independent actors with mixed motives of cooperation and competition. In general-sum and n-player settings, finding even an approximate Nash Equilibrium is known to be PPAD-complete (Daskalakis et al. [8]), rendering it computationally intractable for large-scale environments. Moreover, Nash Equilibria in these environments lose their interchangeability property. By contrast, if players independently compute and execute different Nash Equilibria, the resulting joint policy is not guaranteed to be an equilibrium.

To circumvent the intractability and non-interchangeability of NE, recent theoretical efforts have shifted toward alternative solution concepts, most notably Correlated Equilibria (CE) and Coarse Correlated Equilibria (CCE) (Aumann [2]). These concepts introduce a hypothetical correlating device or mediator that recommends actions to participating agents. Because the constraints defining a CE or CCE form a convex polytope defined entirely by linear inequalities, these equilibria can be computed in polynomial time via linear programming, which is much

more efficient than computing a NE (Daskalakis et al. [8]). A complementary computational route is provided by no-regret learning dynamics such as regret matching (Hart and Mas-Colell [18]). If every player follows a no-regret algorithm, the empirical distribution of joint play converges to the set of coarse correlated equilibria, with polynomial-time per-round updates. Despite this theoretical tractability, the raw dimensional size of the linear programs in extensive-form representations remains a severe bottleneck, necessitating the development of advanced policy generation algorithms.

The Joint Policy Space Response Oracles (JPSRO) framework (Marris et al. [26]) was designed to sidestep this exponential blow-up by computing (C)CE not over the raw extensive-form joint action space but over a much smaller *meta-game* whose actions are policies discovered iteratively through best response. JPSRO is guaranteed to converge to a (C)CE of the underlying game in finitely many iterations. However, as its policy pool grows across iterations, the meta-game itself accumulates an exponential number of joint policies in the number of players, and the LP its meta-solver must instantiate again becomes the computational bottleneck. JPSRO therefore inherits the same scaling wall it was designed to circumvent, but at the level of meta-policies rather than primitive actions, motivating a finer-grained decomposition of the meta-solver itself.

1.2 Problem Statement

We will investigate whether *column generation*, a classical decomposition technique for linear programs with intractably many variables, can be integrated into the JPSRO meta-solver to enable equilibrium computation in n-player general-sum games at scales beyond the reach of full-LP JPSRO. This includes exploring:

1. Can column generation maintain a small, dynamically chosen support over joint meta-policies while preserving JPSRO's convergence guarantees to a (C)CE of the underlying extensive-form game?

2. How does the resulting algorithm compare to standard JPSRO in convergence speed,

scalability with the number of players, and recovery of welfare-optimal equilibria?

3. What are the structural limitations of this approach, particularly under the CCE objective where the best-response oracle returns only a single policy per player per iteration?

1.3 Contributions

We will make the following contributions:

- **JPSRO-CG algorithm.** We introduce JPSRO-CG (Algorithm 2), which embeds column generation into the JPSRO meta-solver, maintaining a small support S of joint policies across outer iterations by warm-starting from the previous support. A two-stage welfare phase targets welfare-optimal equilibria without ever solving the full-support LP.

- **Support-size bounds.** We prove polynomial upper bounds on the minimum support of an optimal (C)CE under any linear objective: $2 + \sum_i |\mathcal{A}_i|(|\mathcal{A}_i| - 1)$ for CE (Theorem 1) and $2 + \sum_i |\mathcal{A}_i|$ for CCE (Theorem 2), ensuring JPSRO-CG's support can in principle stay polynomial in the action sizes.

- **Convergence under approximate meta-solvers.** We generalise the JPSRO convergence proof (Theorems 3 and 4): with an exact best-response oracle and an ϵ-(C)CE meta-solver, JPSRO converges to an ϵ-(C)CE of the underlying game, recovering the original result at $\epsilon = 0$.

- **Empirical evaluation.** On Kuhn Poker (3 and 4 players), Sheriff, Trade Comm, and Tiny Bridge, JPSRO-CG converges to a CCE in 4-player Kuhn Poker within the time budget where standard JPSRO does not (Figure 5.1), recovers the welfare-optimal CE on Sheriff and Trade Comm, and matches JPSRO's convergence rate in smaller games on a much smaller support.

1.4 Outline

The remainder of this report is organised as follows. Chapter 2 reviews CE and CCE, their approximate counterparts, column generation, and prior n-player meta-game methods (α-Rank, projected replicator dynamics, CFR, and JPSRO). Chapter 3 formulates the primal and dual LPs for an approximate and welfare-optimal (C)CE over a restricted support, and derives the pricing subproblems. Chapter 4 establishes the polynomial support-size bounds, generalises JPSRO's convergence theorem to ϵ-approximate meta-solvers, presents JPSRO-CG with its two-stage welfare phase, and describes the empirical setup. Chapter 5 reports the empirical comparison on Kuhn Poker, Sheriff, Trade Comm, and Tiny Bridge. Chapter 6 analyses the limitations of column generation without JPSRO's outer loop, the exploration–exploitation tension, the equilibrium-selection problem, and concrete improvements (dual stabilisation, welfare-aware pricing, approximate BR oracles, graphical-game pricing, team extensions, and warm-starting). Chapter 7 concludes. A glossary of the symbols used throughout is collected in Table A.1 of Chapter A.

2 Background and Preliminaries

2.1 Correlated Equilibrium and Coarse Correlated Equilibrium

Let us define a game with n individual agents. Each agent i has a set of actions $a_i \in \mathcal{A}_i$. A joint action $\mathbf{a} = (a_1, \ldots, a_n)$ is an element of the joint action space $\mathcal{A} = \bigotimes_{i \in [n]} \mathcal{A}_i$. The utility function of agent i for a joint action $\mathbf{a}$ will be denoted as $u_i(\mathbf{a})$. For ease of notation, we will define the indices of all players except for player i to be $-i = \{1, \ldots, i-1, i+1, \ldots n\}$. A distribution $x \in \Delta(\mathcal{A})$ is a *correlated equilibrium* (CE) if, for all $i \in [n]$ and all $a_i, a_i' \in \mathcal{A}_i$ with $a_i \neq a_i'$:

$$\sum_{a_{-i} \in \mathcal{A}_{-i}} x(a_i, a_{-i})[u_i(a_i, a_{-i}) - u_i(a_i', a_{-i})] \geq 0 \tag{2.1}$$

where $\mathcal{A}_{-i}$ denotes the set of joint actions of all players except player i. In other words, for each player i and for each action a_i, the player gains nothing from deviating from a_i to another action a_i', conditioned on the player receiving a_i as the recommended signal (Aumann [2]).

For simplicity, we will write

$$u_i(x) = \mathop{\mathbb{E}}_{\mathbf{a}\sim x} u_i(\mathbf{a}).$$

Similarly, a distribution $x \in \Delta(\mathcal{A})$ is a *coarse correlated equilibrium* (CCE) if, for all $i \in [n]$ and for all $a_i' \in \mathcal{A}_i$:

$$\sum_{\mathbf{a}\in\mathcal{A}} x(\mathbf{a})[u_i(\mathbf{a}) - u_i(a_i', a_{-i})] \geq 0 \tag{2.2}$$

In other words, $u_i(x) \geq \mathbb{E}_{\mathbf{a}\sim x}[u_i(a_i', a_{-i})]$ (Moulin and Vial [28], Sychrovský et al. [37]). This means that before seeing any recommendation, a player prefers to commit to the distribution x rather than committing to play a fixed action a_i' against the marginal distribution of the other players. This means that the region defined by the CE constraints is a subset of that defined by the CCE constraints ($CE \subseteq CCE$).

Note that the feasible region defined by both the CE and the CCE is a convex polytope since the constraints are all linear (Gilboa and Zemel [14]). Furthermore, a Nash equilibrium (NE) is also a CE because an NE distribution x is a CE distribution that factorises into its marginals ($x(a) = \prod_p x_p(a_p)$) (Nau et al. [31]). Since an NE always exists in any finite game (Nash [30]), a CE and a CCE will also exist. Therefore, the constraints always define a feasible LP.

2.2 Approximate (C)CE

Given a joint distribution $x \in \Delta(\mathbf{A})$, the ϵ-CE gap for player i associated with a deviation from a recommended action a_i to an alternative a_i' is the expected utility gained by following this deviation rule. This expectation is weighted by the joint probability of receiving a_i and the others playing a_{-i} (Goldberg and Roth [16]). The global ϵ-gap of a distribution is the

maximum such gain across all players and all action pairs.

An ϵ-CE is a joint distribution x that satisfies the following relaxed linear constraint for all agents $i \in [n]$ and for all possible deviations $a_i \neq a_i' \in \mathcal{A}_i$:

$$\sum_{a_{-i} \in \mathcal{A}_{-i}} x(a_i, a_{-i})[u_i(a_i', a_{-i}) - u_i(a_i, a_{-i})] \leq \epsilon \tag{2.3}$$

Analogously, an ϵ-CCE is a joint distribution x that, for all $i \in [n]$ and all $a_i' \in \mathcal{A}_i$, satisfies the relaxed coarse incentive constraint:

$$\sum_{\mathbf{a} \in \mathcal{A}} x(\mathbf{a})[u_i(a_i', a_{-i}) - u_i(\mathbf{a})] \leq \epsilon \tag{2.4}$$

Equivalently, $u_i(x) \geq \mathbb{E}_{\mathbf{a} \sim x}[u_i(a_i', a_{-i})] - \epsilon$; no player can gain more than ϵ in expectation by unilaterally committing to a fixed action a_i' in advance. As $\epsilon \to 0$ both definitions recover the exact CE and CCE polytopes from (2.2), and the inclusion $\epsilon\text{-}CE \subseteq \epsilon\text{-}CCE$ is preserved.

2.3 Column Generation

Column Generation (CG) is an iterative process designed for linear programs containing an intractable number of variables (columns) but a manageable number of constraints (rows). The technique traces its origins to the Dantzig–Wolfe decomposition for block-structured linear programs (Dantzig and Wolfe [7]) and to the seminal application by Gilmore and Gomory [15] to the cutting-stock problem, where the set of possible cutting patterns is too large to enumerate but each pattern can be generated on demand by solving a knapsack subproblem. The framework has since become a workhorse for large-scale linear and integer optimisation, with applications including vehicle routing, crew scheduling, and graph colouring (Lübbecke and Desrosiers [25], Lübbecke [24], Desaulniers et al. [9]). In the context of equilibrium generation, the variables represent the probabilities assigned to every possible global joint action $\mathbf{a} \in \mathbf{A}$,

while the constraints are simply the polynomial number of incentive-compatibility conditions (one for every player and every possible pair of actions for CE, or one per player per action for CCE).

The framework avoids enumerating $\mathbf{A}$ by operating on a small active support set $\mathbf{A'} \subseteq \mathbf{A}$, forming the Restricted Master Problem (RMP), a tractable LP whose feasible region is a face of the full LP polytope, with columns in $\mathbf{A} \setminus \mathbf{A'}$ implicitly assigned probability zero. Solving the RMP yields a primal x^* supported on $\mathbf{A'}$ and a dual (λ^*, μ^*) for the incentive and normalisation constraints.

For each $\mathbf{a}^\dagger \in \mathbf{A}$, LP duality defines the *reduced cost*

$$\bar{c}(\mathbf{a}^\dagger) = c(\mathbf{a}^\dagger) - (\lambda^*, \mu^*)^\top A(\mathbf{a}^\dagger),$$

the marginal change in the RMP objective per unit increase of $x(\mathbf{a}^\dagger)$ from zero, after pricing in the resources the column consumes. In a minimisation RMP, $\bar{c}(\mathbf{a}^\dagger) \geq 0$ is exactly the dual-feasibility constraint for column $\mathbf{a}^\dagger$; if it holds for every $\mathbf{a}^\dagger \in \mathbf{A}$ the algorithm terminates at the full LP optimum, while *any* column with $\bar{c}(\mathbf{a}^\dagger) < 0$ strictly improves the objective when added to $\mathbf{A'}$. The RMP is then re-solved, typically warm-started from the previous basis (Lübbecke and Desrosiers [25]).

This motivates the *pricing subproblem*,

$$\mathbf{a}^* = \arg\min_{\mathbf{a}^\dagger \in \mathbf{A}} \bar{c}(\mathbf{a}^\dagger),$$

with termination check $\bar{c}(\mathbf{a}^*) \geq 0$. Since any column with negative reduced cost improves the objective, the algorithm remains correct if the oracle returns *any* such column rather than the minimiser, and this "partial pricing" is often faster per iteration in practice.

Recent work has extended column generation to game-theoretic settings, but each existing approach is restricted in ways that prevent direct application to general-sum n-player meta-games. Celli et al. [6] apply CG to ex-ante correlated equilibria in two-player sequential games and prove the multi-player generalisation NP-hard. Farina et al. [11] rely on a two-player

bilinear saddle-point structure that breaks for $n \geq 3$. Zhang et al. [40] solve a single fixed extensive-form general-sum game with two-sided CG but provide no warm-starting mechanism across an oracle-driven meta-loop. Farina et al. [12] target zero-sum team-versus-adversary coordination rather than general-sum CE. These results motivate the integration of CG into the JPSRO meta-solver developed below.

2.4 Previous Work

Several alternative meta-solvers have been proposed for n-player general-sum games but each falls short of the equilibrium-quality target pursued here. Omidshafiei et al. [32] introduce α-Rank, a stationary-distribution ranking over pure profiles that is unique and avoids equilibrium selection, but its cost is cubic in $|\mathbf{A}|$, exponential in the number of players. Muller et al. [29] embed α-Rank in a PSRO loop and propose Projected Replicator Dynamics (PRD) as a cheap approximate-Nash meta-solver, but PRD has no closed-form exploitability bound, requires a hand-tuned probability floor, and can cycle in general-sum settings. Abou Risk and Szafron [1] and Gibson [13] apply Counterfactual Regret Minimisation (CFR) to multi-player poker, but in general-sum settings CFR converges only to a low-average-regret profile rather than an exact (C)CE and offers no equilibrium-selection mechanism. JPSRO replaces these evolutionary and regret-based meta-solvers with an exact (C)CE LP, motivating its choice as our baseline.

2.4.1 Joint policy space response oracles

To fully automate the discovery of robust strategies in intractable game spaces, the Joint Policy Space Response Oracles (JPSRO) framework was introduced in Marris et al. [26]. JPSRO is a meta-algorithmic method for computing correlated equilibria that serves as a direct extension to the traditional Policy Space Response Oracles (PSRO) algorithm described in Lanctot et al. [22]. Standard PSRO is powerful in two-player zero-sum scenarios, but it is limited by the assumption that policy mixtures are independent. This structural assumption heavily biases PSRO toward finding Nash Equilibria, which fundamentally limits its utility in n-player general-sum settings (Bighashdel et al. [5]). JPSRO circumvents this limitation by explicitly

modeling joint distributions over policies, providing a mechanism that seamlessly converges to broader solution concepts like CEs and CCEs.

JPSRO operates on the *normal-form reduction* of the underlying extensive-form game; each deterministic policy π_i is treated as a single pure action of player i in a meta-game whose payoffs are the expected returns of the corresponding joint policy. Equilibria are therefore computed over distributions on $\bigotimes_i \Pi_i^{0:t}$ rather than over information-set behaviour, so the recovered solution concept is the *normal-form* (coarse) correlated equilibrium (NFCCE / NFCE) which is strictly weaker than its extensive-form counterpart, since the correlation device randomises only over whole policies and cannot condition recommendations on realised histories (Avis et al. [3]).

JPSRO iteratively builds the solution from a restricted sub-game by alternating a Best Response (BR) oracle and a joint-distribution meta-solver (MS). At iteration t, let $\Pi_p^{0:t}$ denote the policies found so far for player p and $\Pi^{0:t} = \bigotimes_p \Pi_p^{0:t}$ the corresponding joint-policy space. The Expected Return (ER) maps these to a normal-form sub-game $G^{0:t}$; the MS returns a joint distribution σ^t over $\Pi^{0:t}$; and the BR oracle returns, for each player, an optimal deviation policy against the opponents' marginals (CCE) or against the conditional distribution given each recommendation (CE). The algorithm terminates when no player's BR yields a positive deviation value, and Marris et al. [26] prove that in finite action spaces this occurs after finitely many iterations at an exact (C)CE of the underlying extensive-form game, with monotonic exploitability reduction along the way.

The Meta-Solver is the primary computational bottleneck. At iteration t it formulates an LP with $\prod_{i\in[n]} |\Pi_i^{0:t}|$ variables, which scales exponentially in n, and its constraint matrix must be instantiated in memory; the LP is also re-solved from scratch at each iteration, discarding the optimal σ^{t-1}. Theorems 1 and 2 (Chapter 4) motivate a small-support meta-solver. We address these inefficiencies by integrating Column Generation into the MS; a Restricted Master Problem maintains only an active support of joint policies, and a pricing subproblem admits new columns with negative reduced cost, warm-starting from the previous iteration's basis.

3 Equilibrium Generation

3.1 Formulating the column generation

We split the meta-solve into two phases: a *feasibility* phase, treated below, that grows a support $\mathbf{A}'$ until it admits an exact CE, and an *optimisation* phase (Section 3.2) that maximises a linear objective over that support. In each phase we state the RMP, take its dual, and read off a pricing subproblem over $\mathbf{A}$; these are the building blocks plugged into JPSRO-CG in Chapter 4.

Suppose that we select a support set $\mathbf{A}' \subseteq \mathbf{A}$ of joint actions whose probabilities will be non-zero for our distribution. We would like to find the distribution that minimises the ϵ-gap to a CE. We will assume that a CE supported by $\mathbf{A}'$ doesn't exist, so ϵ will be positive:

$$
\begin{aligned}
\min_{x,\epsilon} \quad & \epsilon \\
\text{s.t.} \quad & \sum_{a_{-i} \in A_{-i}} x(a_i, a_{-i})[u_i(a'_i, a_{-i}) - u_i(a_i, a_{-i})] \leq \epsilon \qquad \forall i \in [n], a_i \neq a'_i \in A_i \\
& \sum_{\mathbf{a} \in \mathbf{A}'} x(\mathbf{a}) = 1 \\
& x(\mathbf{a}) \geq 0 \qquad \forall \mathbf{a} \in \mathbf{A}'
\end{aligned} \tag{3.1}
$$

The dual of the LP can be formulated as follows:

$$
\begin{aligned}
\max_{\lambda,\mu} \quad & \mu \\
\text{s.t.} \quad & \sum_{i=1}^{n} \sum_{a_i \in A_i} \sum_{a'_i \neq a_i} \lambda_{i,a_i,a'_i} = 1 \\
& \mu \leq \sum_{i=1}^{n} \sum_{a'_i \neq a_i} \lambda_{i,a_i,a'_i}[u_i(a'_i, a_{-i}) - u_i(a_i, a_{-i})] \qquad \forall \mathbf{a} = (a_i, a_{-i}) \in \mathbf{A}' \\
& \lambda_{i,a_i,a'_i} \geq 0 \qquad \forall i, a_i, a'_i : a_i \neq a'_i
\end{aligned} \tag{3.2}
$$

When solving the dual to get optimal variables μ^*, $\lambda^*_{i,a_i,a'}$, we can formulate the subproblem for finding a new column:

$$\bar{c}(\mathbf{a}^\dagger) = 0 - \left(\mu^* + \sum_{i \in [n]} \sum_{a' \in A_i} \lambda^*_{i,\mathbf{a}^\dagger_i,a'} \cdot \left(u_i(\mathbf{a}^\dagger) - u_i(a', \mathbf{a}^\dagger_{-i}) \right) \right)$$

Since μ^* is independent of $\mathbf{a}^\dagger$, finding the column with most negative reduced cost reduces to the maximisation problem:

$$\mathbf{a}^* = \arg\max_{\mathbf{a}^\dagger \in \mathbf{A}} \sum_{i=1}^{n} \sum_{a'_i \neq \mathbf{a}^\dagger_i} \lambda_{i,\mathbf{a}^\dagger_i,a'_i} [u_i(\mathbf{a}^\dagger) - u_i(a'_i, \mathbf{a}^\dagger_{-i})] \tag{3.3}$$

3.2 Equilibrium Optimisation

Now suppose that we have a support set $\mathbf{A'} \subseteq \mathbf{A}$ on which a CE exists. Now say we want to maximise some linear objective on the distribution that is supported by $\mathbf{A'}$, say $f(x)$. We now have this LP:

$$\begin{aligned}
\max_{x} \quad & f(x) \\
\text{s.t.} \quad & \sum_{a_{-i} \in A_{-i}} x(a_i, a_{-i})[u_i(a_i, a_{-i}) - u_i(a'_i, a_{-i})] \geq 0 \qquad \forall i \in [n], a_i \neq a'_i \in A_i \\
& \sum_{\mathbf{a} \in \mathbf{A'}} x(\mathbf{a}) = 1 \\
& x(\mathbf{a}) \geq 0 \qquad \forall \mathbf{a} \in \mathbf{A'}
\end{aligned} \tag{3.4}$$

The dual of the LP can be formulated as follows:

$$
\begin{aligned}
\min_{\lambda,\mu} \quad & \mu \\
\text{s.t.} \quad & \mu + \sum_i \sum_{a'_i \neq a_i} \lambda_{i,a_i,a'_i}[u_i(a_i, a_{-i}) - u_i(a'_i, a_{-i})] \geq f(\mathbf{a}) && \forall \mathbf{a} \in \mathbf{A}' \\
& \lambda_{i,a_i,a'_i} \geq 0 && \forall i, a_i \neq a'_i \qquad (3.5)
\end{aligned}
$$

When solving the dual to get optimal variables μ^*, $\lambda^*_{i,a_i,a'}$, we can formulate the subproblem for finding a new column:

$$
\bar{c}(\mathbf{a}^\dagger) = f(\mathbf{a}^\dagger) - \left(\mu^* + \sum_i \sum_{a'_i \neq a_i} \lambda^*_{i,a_i,a'_i}[u_i(a_i, a_{-i}) - u_i(a'_i, a_{-i})] \right)
$$

Since μ^* is independent of $\mathbf{a}^\dagger$, finding the column with the most positive reduced cost reduces to the maximisation problem:

$$
\mathbf{a}^* = \arg\max_{\mathbf{a} \in \mathbf{A}} \left(f(\mathbf{a}) + \sum_{i=1}^{n} \sum_{a'_i \neq a_i} \lambda^*_{i,a_i,a'_i}[u_i(a'_i, a_{-i}) - u_i(a_i, a_{-i})] \right) \qquad (3.6)
$$

4 JPSRO-CG

This chapter introduces JPSRO-CG, the column-generation extension of JPSRO. The first section proves two results that frame the design: support bounds (Theorems 1 and 2) show that an optimal (C)CE lives on a polynomial-size support, justifying a small-support meta-solver; and ϵ-convergence (Theorems 3 and 4) shows that an ϵ-(C)CE of the restricted game lifts to an ϵ-(C)CE of the full game, justifying early termination of the inner CG loop. The second section then presents the algorithm.

4.1 Theoretical Bounds and Convergence

The support bounds below follow from an argument that uses Carathéodory's Theorem. The CE polytope is cut out by polynomially many linear inequalities, and fixing $f(x) = f(y)$ adds one more equality, so any feasible point can be re-expressed on a support whose size matches the constraint count.

Theorem 1. *For any CE y, there exists a CE x such that $f(x) = f(y)$ and $|\operatorname{supp}(x)| \leq 2 + \sum_{i\in[n]} |\mathcal{A}_i|(|\mathcal{A}_i| - 1)$, where f is a linear function on the joint action space and $|\mathcal{A}_i|$ is the number of actions for player i.*

Proof. Let $k \coloneqq \sum_{i\in[n]} |\mathcal{A}_i|(|\mathcal{A}_i| - 1)$ denote the number of CE inequality constraints, and let $d \coloneqq k + 2$. We construct x as a feasible point of the following linear feasibility problem:

$$
\begin{aligned}
\text{Find} \quad & x \\
\text{s.t.} \quad & \sum_{a_{-i}\in A_{-i}} x(a_i, a_{-i})[u_i(a_i, a_{-i}) - u_i(a'_i, a_{-i})] \geq 0 && \forall i \in [n], a_i \neq a'_i \in \mathcal{A}_i \\
& \sum_{\mathbf{a}\in\mathbf{A}} x(\mathbf{a}) = 1 \\
& f(x) = f(y) \\
& x(\mathbf{a}) \geq 0 && \forall \mathbf{a} \in \mathbf{A}
\end{aligned}
$$

To apply Carathéodory's Theorem, we convert the k inequality constraints into equality constraints by introducing a non-negative slack vector $s \in \mathbb{R}^k$, giving $Bx - s = 0$, where $B \in \mathbb{R}^{k\times|\mathbf{A}|}$ collects the coefficients of the CE inequalities, with rows indexed by triples (i, a_i, a'_i) and columns indexed by $\mathbf{a} \in \mathbf{A}$.

Now, let $c \in \mathbb{R}^{|\mathbf{A}|}$ be the coefficient vector defining the linear function f, such that $f(x) = c^T x$. Let $\mathbf{1}^T$ be the row vector of all ones. We can combine all the equality constraints (the CE constraints with slack variables, the probability constraint, and the function value constraint) into a single augmented linear system:

$$\begin{bmatrix} B & -I \\ \mathbf{1}^T & \mathbf{0} \\ c^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \\ f(y) \end{bmatrix}$$

Let M denote this combined block matrix, let $z = \begin{bmatrix} x \\ s \end{bmatrix}$, and let $b = \begin{bmatrix} \mathbf{0} \\ 1 \\ f(y) \end{bmatrix}$. The system can be written simply as $Mz = b$, and M has exactly $d = k + 2$ rows.

Since y is a valid Correlated Equilibrium, setting $s_y = By \geq 0$, the vector $z_y = \begin{bmatrix} y \\ s_y \end{bmatrix} \geq 0$ satisfies $Mz_y = b$, so b lies in the conical hull of the columns of M.

By the Conical Carathéodory's Theorem (Schrijver [35], Corollary 7.1i), if a vector b is in the conical hull of a set of vectors in $\mathbb{R}^d$, it can be expressed as a non-negative linear combination of at most d linearly independent vectors from that set.

Therefore, there exists a non-negative vector $z^* = \begin{bmatrix} x^* \\ s^* \end{bmatrix} \geq 0$ such that $Mz^* = b$, and z^* has at most d strictly positive entries.

Since x^* is a sub-vector of z^*, it has at most d non-zero entries. Together with $Mz^* = b$, this shows that $x \coloneqq x^*$ is a CE with $f(x) = f(y)$ and support size bounded by:

$$|\text{supp}(x)| \leq d = \sum_{i \in [n]} |\mathcal{A}_i|(|\mathcal{A}_i| - 1) + 2$$

□

Theorem 2. *For any CCE y, there exists a CCE x such that $f(x) = f(y)$ and $|\text{supp}(x)| \leq 2 + \sum_{i \in [n]} |\mathcal{A}_i|$, where f is a linear function on the joint action space and $|\mathcal{A}_i|$ is the number of actions for player i.*

Proof. The argument is identical to Theorem 1, with the CE incentive constraints replaced by the CCE incentive constraints $\sum_{\mathbf{a} \in \mathbf{A}} x(\mathbf{a})[u_i(\mathbf{a}) - u_i(a_i', \mathbf{a}_{-i})] \geq 0$ for all $i \in [n]$ and $a_i' \in \mathcal{A}_i$. These are indexed by pairs (i, a_i') rather than triples (i, a_i, a_i'), since a CCE only

requires that no player gains by deviating to a fixed action against the marginal. This gives $k \coloneqq \sum_{i\in[n]} |\mathcal{A}_i|$ inequalities, so the augmented system $Mz = b$ has $d \coloneqq k + 2$ rows, and applying the Conical Carathéodory's Theorem as before yields a CCE x with $f(x) = f(y)$ and $|\operatorname{supp}(x)| \le d = \sum_{i\in[n]} |\mathcal{A}_i| + 2$. □

From Theorem 1 and Theorem 2, we are able to use a reduced-support meta-solver in the JPSRO iteration to find a (C)CE while still recovering the optimum under f. Since the minimum-support optimal (C)CE is bounded polynomially in the number of actions, each iteration of JPSRO can in principle converge to a meta (C)CE far faster than the exponential-size full-support LP would suggest.

We now apply the (C)CE LPs of Chapter 3 to JPSRO's meta-game. Under the normal-form reduction, each player's action set is the policy set $\Pi_i^{0:t}$ and a joint action is a joint policy in $\Pi^{0:t} = \bigotimes_i \Pi_i^{0:t}$. In the remainder of this chapter we therefore instantiate the abstract action notation of Chapters 2 and 3 on the meta-game by writing π_i for a_i, $\pi \in \Pi^{0:t}$ for the joint action $\mathbf{a}$, and $\lambda^t_{i,\pi_i,\pi_i'}$ for the incentive-compatibility duals $\lambda_{i,a_i,a_i'}$.

Theorem 3. *For an MS that produces a ϵ-CCE distribution and an exact BR oracle, the JPSRO algorithm will converge to a ϵ-CCE distribution.*

Proof. Let's define σ^t be the meta-game distribution at iteration t. For CCE, the BR oracle is:

$$\mathrm{BR}_p^{t+1} \in \arg\max_{\pi_p^*\in\Pi_p} \sum_{\pi_{-p}\in\Pi_{-p}^{0:t}} \sigma^t(\pi_{-p}) u_p(\pi_p^*, \pi_{-p}).$$

If σ^t is an ϵ-CCE of the restricted game, then for all p and all $\pi_p' \in \Pi_p^{0:t}$:

$$\sum_{\pi\in\Pi^{0:t}} \sigma^t(\pi)\big(u_p(\pi_p', \pi_{-p}) - u_p(\pi)\big) \le \epsilon.$$

Assume JPSRO terminates at iteration T. Termination requires that, for every p, the BR oracle returns a policy that is already in $\Pi_p^{0:T}$; call it π_p^*. Substituting π_p^* as the deviation in

the restricted game's ϵ-CCE inequality:

$$\sum_{\pi \in \Pi^{0:T}} \sigma^T(\pi)\big(u_p(\pi_p^*, \pi_{-p}) - u_p(\pi)\big) \leq \epsilon. \tag{4.1}$$

By exactness of the BR oracle, π_p^* maximises expected utility against σ_{-p}^T over the full policy space Π_p, so for any $\hat{\pi}_p \in \Pi_p$:

$$\sum_{\pi_{-p}} \sigma^T(\pi_{-p}) u_p(\hat{\pi}_p, \pi_{-p}) \leq \sum_{\pi_{-p}} \sigma^T(\pi_{-p}) u_p(\pi_p^*, \pi_{-p}).$$

Combining with (4.1) yields, for all p:

$$\max_{\hat{\pi}_p \in \Pi_p} \sum_{\pi \in \Pi^{0:T}} \sigma^T(\pi)\big(u_p(\hat{\pi}_p, \pi_{-p}) - u_p(\pi)\big) \leq \epsilon.$$

Thus σ^T is an ϵ-CCE against deviations in the full policy space.

□

Theorem 4. *For an MS that produces a ϵ-CE distribution and an exact BR oracle, the JPSRO algorithm will converge to a ϵ-CE distribution.*

Proof. For CE, the conditional BR oracle is:

$$\mathrm{BR}_p^{t+1}(\pi_p) \in \arg\max_{\pi_p^* \in \Pi_p} \sum_{\pi_{-p} \in \Pi_{-p}^{0:t}} \sigma^t(\pi_{-p}|\pi_p) u_p(\pi_p^*, \pi_{-p}).$$

If σ^t is an ϵ-CE of the restricted game, then for all p and all $\pi_p \neq \pi_p' \in \Pi_p^{0:t}$:

$$\sum_{\pi_{-p} \in \Pi_{-p}^{0:t}} \sigma^t(\pi_p, \pi_{-p})[u_p(\pi_p', \pi_{-p}) - u_p(\pi_p, \pi_{-p})] \leq \epsilon.$$

Assume JPSRO terminates at iteration T. Termination requires that, for every p and every $\pi_p \in \Pi_p^{0:T}$ with $\sigma^T(\pi_p) > 0$, the conditional BR returns a policy that is already in $\Pi_p^{0:T}$; call it

π_p^*. Substituting π_p^* as the deviation:

$$\sum_{\pi_{-p}} \sigma^T(\pi_p, \pi_{-p})[u_p(\pi_p^*, \pi_{-p}) - u_p(\pi_p, \pi_{-p})] \leq \epsilon. \tag{4.2}$$

By exactness of the CE-BR oracle, for any $\hat{\pi}_p \in \Pi_p$:

$$\sum_{\pi_{-p}} \sigma^T(\pi_{-p}|\pi_p) u_p(\hat{\pi}_p, \pi_{-p}) \leq \sum_{\pi_{-p}} \sigma^T(\pi_{-p}|\pi_p) u_p(\pi_p^*, \pi_{-p}).$$

Multiplying by $\sigma^T(\pi_p) > 0$ and combining with (4.2) gives, for all p:

$$\max_{\hat{\pi}_p \in \Pi_p} \sum_{\pi_{-p}} \sigma^T(\pi_p, \pi_{-p})[u_p(\hat{\pi}_p, \pi_{-p}) - u_p(\pi_p, \pi_{-p})] \leq \epsilon.$$

Hence σ^T is an ϵ-CE against deviations in the full policy space. □

Intuitively, this result makes sense because in JPSRO, the exact best response oracle searches the entire full game policy space for the single most profitable deviation against the current joint meta-distribution. If the algorithm terminates, it means the absolute best deviation available anywhere in the full game is already contained within the restricted game's policy space. Therefore, the maximum possible regret in the full game cannot possibly exceed the maximum regret within the restricted game. Because the restricted game solver guarantees the regret is bounded by ϵ, the full game's regret is inherently bounded by that exact same ϵ. Notably, the standard convergence proof presented in the original JPSRO paper is simply the special case of this theorem where $\epsilon = 0$, demonstrating convergence to an exact Correlated or Coarse Correlated Equilibrium.

4.2 Algorithm

The proposed solution to extend JPSRO will be called JPSRO-CG, which incorporates column generation in the JPSRO loop to maintain a small support set. As illustrated in Figure 4.1 and formalised in Figure 4.2, standard JPSRO (Algorithm 1) alternates between finding best

responses to expand the individual policy spaces and computing a meta-strategy σ^t over the full empirical game $G^{0:t}$.

JPSRO-CG (Algorithm 2) introduces an inner column-generation loop to construct a small, restricted support set S dynamically. Instead of solving the full empirical game outright, the meta-solver (MS) evaluates a restricted game over the current support S (the sparse active cells shown in the right panel of Figure 4.1). In the inner `while` loop of Algorithm 2, this restricted MS yields both a tentative meta-strategy σ^t and its associated dual variables λ^t. These dual values inform the Column Generation (CG) subroutine, which searches for a new joint policy (or column) c that improves the objective by identifying a column with a negative reduced cost, and adds it to S. This inner loop iterates until the desired gap falls below a specified tolerance ϵ.

At each inner iteration, the MS solves one of the RMP (C)CE LP from Chapter 3, restricted to the joint policies in S. When the objective is to find an approximate (C)CE, the MS solves the ϵ-gap minimisation LP (3.1), which minimises the maximum incentive to deviate subject to the distribution summing to one. When a specific equilibrium objective must be optimised (for example, maximum welfare), the MS instead solves the equilibrium-optimisation LP 3.4, which maximises $f(x)$ subject to the standard CE incentive-compatibility constraints. In both cases, the primal solution yields the current meta-strategy σ^t; complementary slackness then delivers the optimal dual multipliers $\lambda^t_{i,\pi_i,\pi'_i}$ attached to the incentive-compatibility constraints (the λ variables in LP 3.2 and LP 3.5 respectively).

These dual multipliers serve as the price signal passed from the restricted MS to the Column Generation (CG) subroutine (the λ^t, σ^t arrow in Figure 4.1; line 11 of Algorithm 2). Following the pricing subproblem derived in equation 3.3 and equation 3.6, CG searches for a joint policy $\pi^\dagger \in \Pi^{0:t} \setminus S$ with the most negative reduced cost, whose inclusion in S would improve the current LP bound the most. A column $\pi^\dagger$ has reduced cost

$$\bar{c}(\pi^\dagger) = -\left(\mu^* + \sum_{i=1}^{n} \sum_{\pi'_i \neq \pi^\dagger_i} \lambda^*_{i,\,\pi^\dagger_i,\,\pi'_i} \Big[u_i(\pi^\dagger) - u_i(\pi'_i, \pi^\dagger_{-i})\Big]\right),$$

where μ^* is the dual on the normalisation constraint and $\lambda^*_{i,\pi_i^\dagger,\pi_i'}$ are the duals on the incentive-compatibility constraints from the most recent RMP solve. A column has a negative reduced cost ($\bar{c}(\pi^\dagger) < 0$) if and only if adding it to S would improve the current LP bound; CG therefore selects the column with the most negative reduced cost and adds it to S. If no such column exists—that is, $\bar{c}(\pi^\dagger) \geq 0$ for all $\pi^\dagger \in \Pi^{0:t} \setminus S$—the current σ^t is already optimal over all of $\Pi^{0:t}$ and the inner `while` loop of Algorithm 2 terminates. This termination condition exactly satisfies the hypothesis of the convergence theorem above; once the inner CG loop exits, the meta-strategy σ^t is an ϵ-(C)CE over the full empirical game $G^{0:t}$, so the outer JPSRO convergence guarantee is preserved.

Between outer iterations, S is retained rather than reset. As seen in algorithm 2, it initialises S once to the initial joint policy (line 4) and keeps track of it between outer iterations. When the outer loop adds new best-response policies and the empirical game $G^{0:t}$ expands, all joint policies already in S are remembered; only the new joint policy combinations involving the freshly added best responses are initially absent from S (the orange cells appearing at the boundary of the matrix in Figure 4.1), leaving the next inner CG loop to selectively admit them.

In zero-sum games we do not run the welfare stage; the gap-minimisation phase alone suffices, since social welfare is constant across the (C)CE polytope. In a general-sum game, however, the set of correlated equilibria forms a polytope and equilibria can differ substantially in social welfare $\sum_p \mathbb{E}_\sigma[u_p]$. The gap-minimisation phase finds a CE supported on a small column-generation subspace S, but this equilibrium need not be welfare-optimal. The second stage addresses this; it records the gap ϵ^* achieved by the first phase, uses it as a relaxed feasibility tolerance, and runs a second column-generation loop whose objective is to maximise social welfare subject to (C)CE gap $\leq \epsilon^*$. Columns are added to S one at a time via the same pricing mechanism, driven by the welfare objective's dual variables, until no welfare-improving column remains. A final LP solve over the expanded support then yields the welfare-optimal CE within the subspace, without ever solving the full LP over all joint policies. Note that if this second stage is enabled, a full-support Maximum Welfare CE (MWCE) meta-solver is not

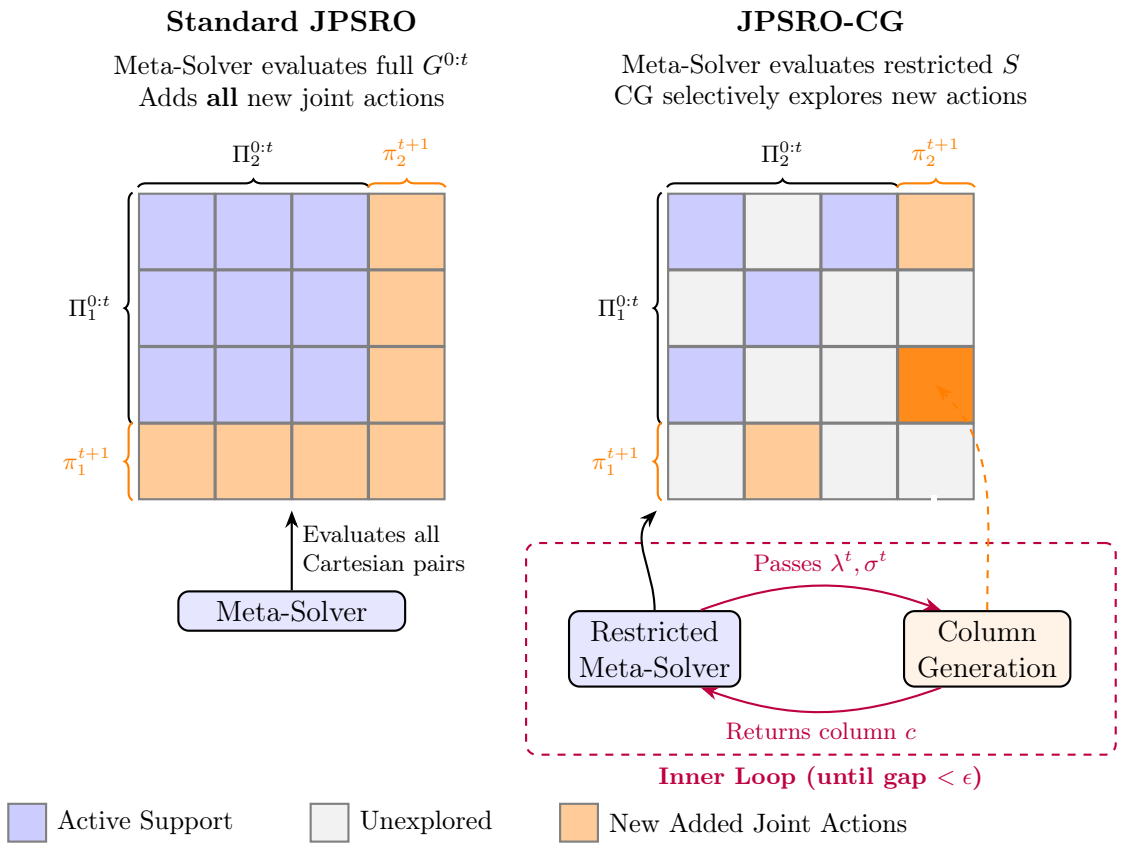


Figure 4.1: Comparison of the evaluated joint policy space in standard JPSRO versus JPSRO-CG. While standard JPSRO requires the meta-solver to evaluate the complete Cartesian product of the discovered policy sets (the full empirical game $G^{0:t}$), JPSRO-CG actively maintains a sparse, restricted support S. The inner loop emphasizes how the Restricted Meta-Solver uses the dual variables λ^t to guide the Column Generation (CG) subroutine, dynamically isolating and adding only high-value joint policies c to the support.

required in the standard JPSRO baseline either, since the welfare CG stage would still be able to find the welfare-optimal CE within the subspace even if the initial gap-minimisation stage returned a non-welfare-optimal CE.

More generally, the second inner loop can optimise any linear objective defined over the joint action space. This follows directly from the pricing subproblem derived in (3.6); the objective term $f(\mathbf{a})$ can be any linear function of the joint action, and the column-generation procedure is otherwise unchanged. One could equally target, for example, maximising the utility of a single player (e.g. $f(\mathbf{a}) = u_1(\mathbf{a})$ for player 1) or any other linear criterion. We focus on welfare maximisation here because the existing JPSRO literature reports welfare as its

Algorithm 1 JPSRO

$\Pi_1^0, \dots, \Pi_n^0 \leftarrow \{\pi_1^0\}, \dots, \{\pi_n^0\}$
$G^0 \leftarrow \mathrm{ER}(\Pi^0)$
$\sigma^0 \leftarrow \mathrm{MS}(G^0)$
for $t \leftarrow \{1, \dots\}$ **do**
  for $p \leftarrow \{1, \dots, n\}$ **do**
    $\{\pi_p^t, \dots\}, \{\Delta_p^t, \dots\} \leftarrow \mathrm{BR}_p(\Pi^{0:t-1}, \sigma^{t-1})$
    $\Pi_p^{0:t} \leftarrow \Pi_p^{0:t-1} \cup \{\pi_p^t, \dots\}$
  $G^{0:t} \leftarrow \mathrm{ER}(\Pi^{0:t})$
  $\sigma^t \leftarrow \mathrm{MS}(G^{0:t})$
  if $\sum_p \Delta_p^t = 0$ **then**
    break
return $\Pi^{0:t}, \sigma^t$

Algorithm 2 JPSRO-CG

$\Pi_1^0, \dots, \Pi_n^0 \leftarrow \{\pi_1^0\}, \dots, \{\pi_n^0\}$
$G^0 \leftarrow \mathrm{ER}(\Pi^0)$
$\sigma^0 \leftarrow \mathrm{MS}(G^0)$
$S \leftarrow (\pi_1^0, \dots, \pi_n^0)$
for $t \leftarrow \{1, \dots\}$ **do**
  for $p \leftarrow \{1, \dots, n\}$ **do**
    $\{\pi_p^t, \dots\}, \{\Delta_p^t, \dots\} \leftarrow \mathrm{BR}_p(\Pi^{0:t-1}, \sigma^{t-1})$
    $\Pi_p^{0:t} \leftarrow \Pi_p^{0:t-1} \cup \{\pi_p^t, \dots\}$
  $G^{0:t} \leftarrow \mathrm{ER}(\Pi^{0:t})$
  while (C)CE-gap$(\sigma^t, G^{0:t}) > \epsilon$ **do**
    $\sigma^t, \lambda^t \leftarrow \mathrm{MS}(G^{0:t}, S)$
    $c \leftarrow \mathrm{CG}(\sigma^t, \lambda^t, G^{0:t})$
    $S \leftarrow S \cup c$
  $\epsilon^* \leftarrow$ (C)CE-gap$(\sigma^t, G^{0:t})$ ▷ achieved gap
  while $\exists$ welfare-improving $c \notin S$ **do** ▷ this loop optimizes the (C)CE
    $\sigma^t, \lambda^t \leftarrow \mathrm{MS}_W(G^{0:t}, S, \epsilon^*)$
    $c \leftarrow \mathrm{CG}(\sigma^t, \lambda^t, G^{0:t})$
    $S \leftarrow S \cup c$
  $\sigma^t \leftarrow \mathrm{MS}_W(G^{0:t}, S, \epsilon^*)$
  if $\sum_p \Delta_p^t = 0$ **then**
    break
return $\Pi^{0:t}, \sigma^t$

Figure 4.2: Pseudocode for JPSRO-CG. The algorithm contains two nested column-generation stages. The first loop iteratively adds columns to the support S until the (C)CE gap falls below ϵ. The second (welfare) loop then fixes the achieved gap ϵ^* as a relaxed feasibility tolerance and adds further columns to maximise social welfare, returning the welfare-optimal equilibrium within the CG subspace.

primary equilibrium-quality metric, making it the natural baseline for comparison.

Let $N \coloneqq \prod_{i \in [n]} |\Pi_i^{0:t}|$ denote the size of the full empirical joint policy space at outer iteration t, and let K denote the number of (C)CE incentive constraints, which is polynomial in $\sum_i |\Pi_i^{0:t}|$. Standard JPSRO solves a single LP with N variables and K constraints, where N grows exponentially in n. JPSRO-CG instead performs a sequence of inner iterations, each solving an RMP with only $|S|$ variables and K constraints (time polynomial in $|S|$ and K) together with a pricing scan over $\Pi^{0:t}$ that costs $O(N)$ time but stores only the current best column. Theorems 1 and 2 guarantee that a polynomial-sized support with $|S| \leq K + 2$ exists, but column generation is not guaranteed to recover such a support, and in the worst case the inner loop may add more columns before terminating. However, empirically, the support produced is typically far smaller than N (as can be seen in the experimental results), so the per-iteration memory and LP cost of the MS step are expected to be substantially reduced

relative to standard JPSRO.

4.3 Empirical Setup

The experimental games are open-sourced on OpenSpiel (Lanctot et al. [23]), which provides the game implementations and the exact best-response oracle used to compute (C)CE gaps. The JPSRO outer loop and meta-solvers are a fork of the reference implementation released with Marris et al. [26]. All LPs and QPs are solved with ECOS via `cvxpy`. Full game parameters, solver tolerances, compute resources, and a per-configuration breakdown of the experimental sweep are given in Chapter B.

4.3.1 Games choice

We evaluate JPSRO-CG against JPSRO on four extensive-form games of differing structure and scale (full parameters in Table B.1). **Kuhn Poker** (Kuhn [20]), extended to n players by Lanctot [21], is a zero-sum simplified poker game with $2^{(n+1)2^{n-1}}$ pure strategies per player. We run it at $n \in \{3, 4, 5\}$ to stress-test scaling. We prefer it to the more commonly used Leduc Poker (Southey et al. [36]) because Leduc's larger per-player action set (from two betting rounds and a community card) conflates the per-player and joint-space scaling axes, whereas Kuhn isolates the latter. **Trade Comm** is a 2-player cooperative trading game prone to sub-optimal local equilibria. **Sheriff** is a 2-player general-sum game of multi-round bribe negotiation modelled on *Sheriff of Nottingham*. **Tiny Bridge** (Lanctot et al. [23]) is a 2-player cooperative simplification of contract bridge with coordination under partial information as the dominant difficulty. Kuhn is used purely to measure scalability; the three general-sum games are used to evaluate recovery of the welfare-optimal (C)CE.

4.3.2 Method of the experiment

Both JPSRO and JPSRO-CG were run either for 30 iterations or until 30 minutes had elapsed, whichever stopping condition was achieved first. In each configuration, we run an instance of the algorithm for both correlated equilibria (CE) and coarse correlated equilibria (CCE). All

runs used uniform policy initialisation, the largest-gap best response selection strategy, and updated all players simultaneously at each iteration. The CE gap is computed for both the training and evaluation meta-solvers at every iteration to measure convergence. Each iteration is also timed, and the expected value for each player under both meta-distributions is recorded.

The meta-solver configuration is tailored to the game type. For the general-sum games (Sheriff and Trade Comm), where finding any CE is easy but welfare is the primary target, we use the zero-tolerance CG configuration paired with the two-stage welfare CG of Figure 4.2, testing whether JPSRO-CG recovers the same welfare-optimal CE as full-support MWCE JPSRO. For zero-sum Kuhn Poker, welfare is moot, so we compare JPSRO against ε-approximate and zero-tolerance JPSRO-CG (welfare stage disabled) to isolate the scaling benefit of a restricted support across 3-, 4-, and 5-player instances.

JPSRO configurations

For each equilibrium type, three meta-solver choices were compared, all paired with a fixed welfare-maximising evaluation meta-solver (MWCE / MWCCE) for cross-configuration comparability. **MGCE** (Marris et al. [26]'s primary recommendation) maximises the Gini coefficient over the joint distribution, giving the most diverse training signal at the cost of solving a QP. **Approximate MGCE** relaxes the CE constraints to an ε-CE polytope ($\varepsilon = 1/100$, scaled by $\max_j |\bar{A}_j|$ as in Table B.3), isolating the effect of constraint relaxation on speed and quality. **Approximate MWCE** replaces the Gini QP with a linear welfare objective under the same ε-relaxation, reducing the MS to an LP and contrasting exploration-focused (MGCE) against welfare-focused (MWCE) training signals. CCE runs use the analogous MGCCE, Approx-MGCCE, and Approx-MWCCE variants.

JPSRO-CG configurations

Three JPSRO-CG variants were evaluated for each equilibrium type, formed by combining two binary axes: CG termination (ε-**approximate**, stopping at gap $\leq \varepsilon = 1/100$; or **zero-tolerance**, gap $= 0$), and the support visible to the evaluation MS (**restricted**, limited to

S; or **full**, the entire meta-game). **Restricted-support CG** uses ε-approximate termination with restricted evaluation, keeping iteration times low. **Full-support CG** uses ε-approximate termination with full evaluation, serving as a diagnostic, where agreement with the restricted-support gap means S captures the equilibrium and divergence reveals omitted joint policies. **Zero-tolerance CG** runs CG until the restricted-support gap is exactly zero and pairs naturally with the two-stage welfare CG of Figure 4.2 on the general-sum games. In all variants the CG support warm-starts from the previous iteration.

Logged metrics

The full schema of per-iteration CSV columns is listed in Chapter B. The central convergence metric is the *(C)CE gap*, the most a player can gain by deviating from σ^t. For a CCE target,

$$\mathrm{gap}_p^{\mathrm{CCE}}(\sigma^t) = \max\left\{ \max_{\pi_p^*} \mathbb{E}_{\pi_{-p}\sim\sigma^t_{-p}}\big[u_p(\pi_p^*, \pi_{-p})\big] - \mathbb{E}_{\pi\sim\sigma^t}\big[u_p(\pi)\big],\ 0 \right\};$$

for a CE target the deviation is conditional on each recommendation π_p,

$$\mathrm{gap}_p^{\mathrm{CE}}(\sigma^t) = \sum_{\pi_p\in\Pi_p^{0:t}} \sigma^t(\pi_p)\ \max\left\{ \max_{\pi_p^*} \mathbb{E}_{\pi_{-p}\sim\sigma^t(\,\cdot\,|\pi_p)}\big[u_p(\pi_p^*, \pi_{-p})\big] - \mathbb{E}_{\pi_{-p}\sim\sigma^t(\,\cdot\,|\pi_p)}\big[u_p(\pi_p, \pi_{-p})\big],\ 0 \right\}.$$

The BR oracle searches the full extensive-form game, so a zero gap certifies an exact (C)CE of the original game (Marris et al. [26]). Convergence plots report the *sum gap* $\sum_p \mathrm{gap}_p^{(\mathrm{C})\mathrm{CE}}(\sigma^t)$. The training-MS gap tracks the distribution that drives best responses; the evaluation-MS gap uses the same welfare-maximising MS across configurations for comparability.

5 Results

5.1 Recreation of experiments

Before evaluating our new method, we first verify that our implementation of JPSRO is correct by attempting to recreate the results from the original JPSRO paper (Marris et al.

[26]). JPSRO was run on the three games used in the paper, and the convergence behaviour of our implementation closely matches that of the paper. The exploitability gaps trace the same trajectory across iterations and converge at the same rate, holding for both MGCE and approximate MGCE as reported in the original JPSRO paper. The remaining diagnostics agree as well; the number of policies per player grows at the same pace across iterations, and the per-player evaluation values settle to the same limits. The convergence plots in Chapter C are provided in the same format as those in Marris et al. [26] for anyone wishing to compare the results directly. Having successfully reproduced the original results to a good accuracy, we are confident that our implementation is correct and that it serves as a reliable baseline against which to compare our proposed method.

5.2 Convergence to equilibrium

In all of the experiments we ran, JPSRO-CG's convergence to an equilibrium either closely matched the convergence seen in standard JPSRO or surpassed it in speed. In particular, in larger games, there is a significant improvement in the computational cost of finding an equilibrium. For example, in 4-player Kuhn Poker, which was the largest game, JPSRO-CG was able to converge to a CCE within 3 minutes, while none of the JPSRO variants were able to converge within the time frame (see figure 5.1). In other games, such as Tiny Bridge and Trade Comm (see figure C.4 and figure C.6), JPSRO-CG was also able to converge to an equilibrium faster than JPSRO.

In 4-player Kuhn Poker, zero tolerance and ϵ-approximate column generation with an evaluation meta-solver on the restricted support outperformed standard JPSRO with both an exact MS and an ϵ-MS. In 3-player Kuhn Poker (figure C.1), the joint action space is much smaller, so the efficiency of column generation is less pronounced. The graphs of the convergence of each algorithm for each game can be found in the appendix C.

Furthermore, in all the experimented games, the convergence of the (C)CE gap over the iterations seems to be comparable for both with and without column generation (see figure 5.1). This suggests that using column generation as a technique to speed up the computation

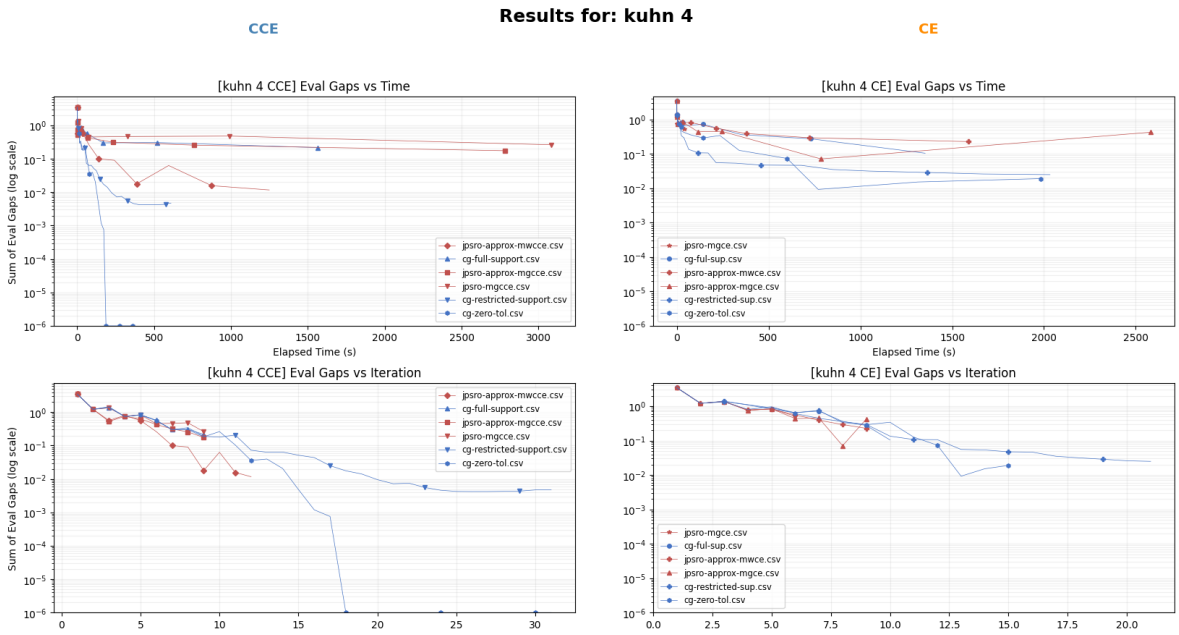


Figure 5.1: Convergence of JPSRO and JPSRO-CG on 4-player Kuhn Poker. Zero tolerance JPSRO-CG was the only algorithm that converged to a CCE and no algorithms were able to converge to a CE.

of equilibria in the meta game still produces effective policies in expanding the meta game.

From Theorem 1 and Theorem 2, we have polynomial upper bounds on the minimum support size required, expressed in terms of the number of actions each player has. Figure 5.2 illustrates that the empirical results of the support size over time were within the same order of magnitude as the theoretical support size.

Running Kuhn Poker with 5 players led to out-of-memory failures during execution. The number of pure strategies per player is $2^{(n+1)2^{n-1}}$, which for $n = 5$ gives 2^{96} policies per player. The total number of joint actions in the meta game is therefore 2^{480}, using an exact BR to traverse the game tree would be intractable, and at this scale, approximate BRs are needed.

5.3 Convergence to optimal equilibrium

There are no theoretical guarantees that JPSRO converges to the optimal equilibrium (such as maximum welfare). However, empirically, JPSRO has been able to find the MWCE and MWCCE of general-sum games like Sheriff and Trade Comm. This section evaluates how well

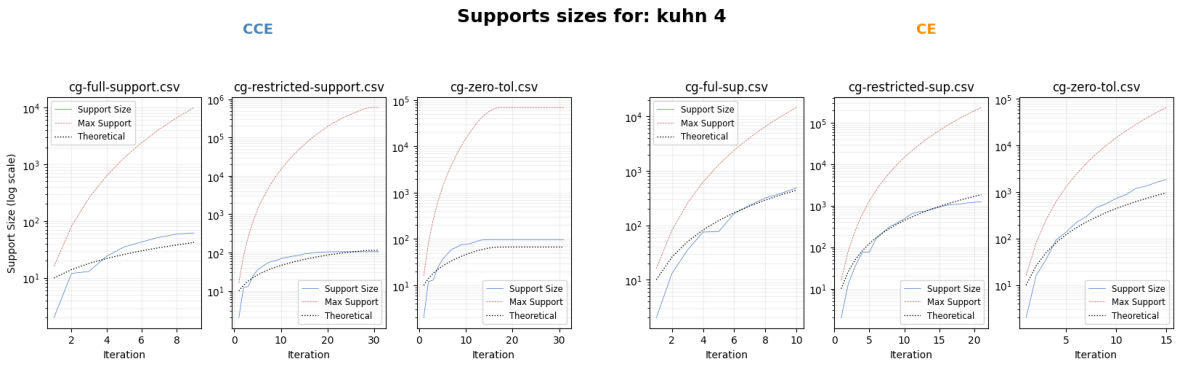


Figure 5.2: In blue is the size of the supports at each iteration. In red is the maximum support possible (the total number of joint actions in the meta game). The dotted black line is the theoretical support bound.

JPSRO-CG recovers the maximum-welfare (C)CE compared to JPSRO with the maximum gini MS.

Trade Comm. For Trade Comm (3 and 4 items), the MWCE and MWCCE each yield an expected payoff of 1.0 per player. For both 3- and 4-item Trade Comm, JPSRO with the MGCE and approximate-MWCE meta-solvers converges to the optimal welfare under both CE and CCE. JPSRO-CG with the zero-tolerance configuration also recovers the MWCE. However, for the MWCCE, JPSRO-CG was unable to converge to the optimal welfare equilibrium.

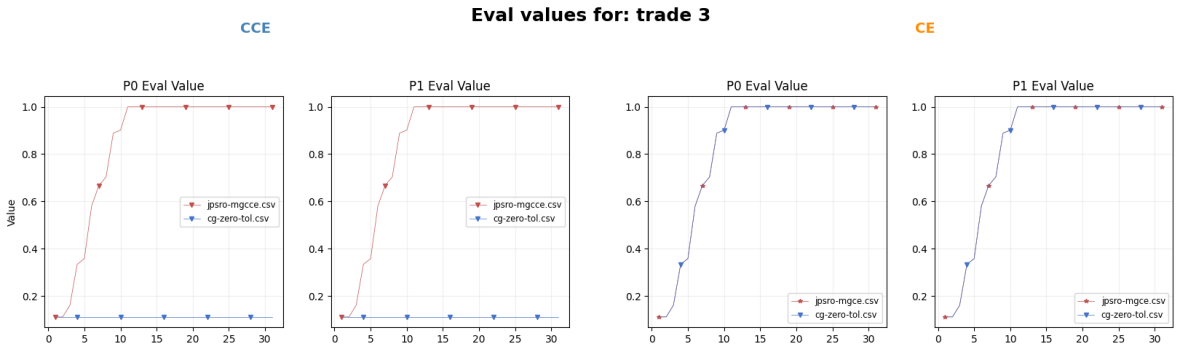


Figure 5.3: Social welfare of the meta-distribution over iterations on 3-item Trade Comm. JPSRO-CG zero-tolerance recovers the MWCE in the same number of iterations as standard JPSRO, but plateaus below the optimum under the CCE objective.

Sheriff. For Sheriff, the MWCCE social welfare is 13.64 and the MWCE social welfare is 0.82 (Marris et al. [26]). For Sheriff, JPSRO with MGCE and approximate-MWCE meta-solvers again finds the optimal welfare for both CE and CCE. JPSRO-CG zero-tolerance successfully recovers the MWCE (0.82), matching the reference value. However, it fails to find the MWCCE (13.64), converging to a suboptimal welfare under the CCE constraint. This mirrors the Trade Comm result and further suggests that the welfare CG stage has a systematic limitation under CCE constraints that is not present for CE.

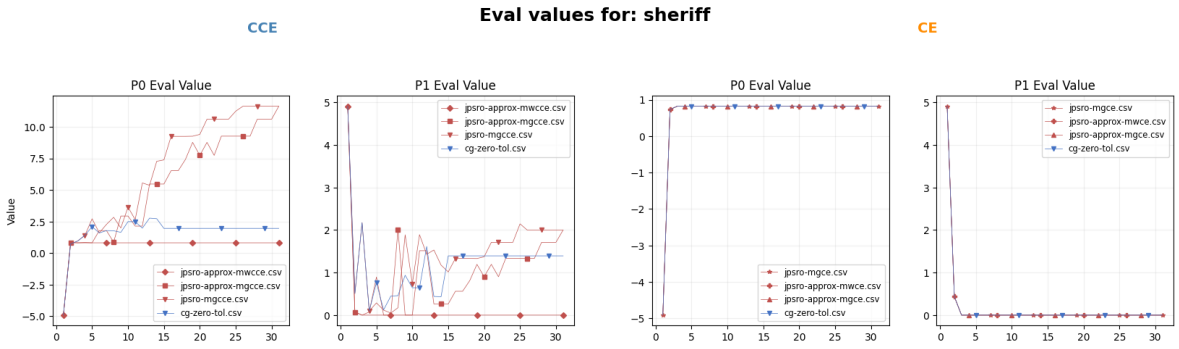


Figure 5.4: Social welfare of the meta-distribution over iterations on Sheriff. JPSRO-CG matches the MWCE reference of 0.82 but converges to a strictly suboptimal value under the CCE objective relative to the 13.64 reference.

Tiny Bridge. No closed-form reference for the optimal welfare in Tiny Bridge is available. Empirically, JPSRO-CG converges to a high-welfare equilibrium very quickly relative to standard JPSRO. However, given more iterations, standard JPSRO continues to improve and eventually finds a higher-welfare equilibrium than JPSRO-CG. This suggests a trade-off where JPSRO-CG is more sample-efficient early in training, but its restricted support may prevent it from exploring the full equilibrium space, causing it to plateau at a locally good but globally suboptimal CE.

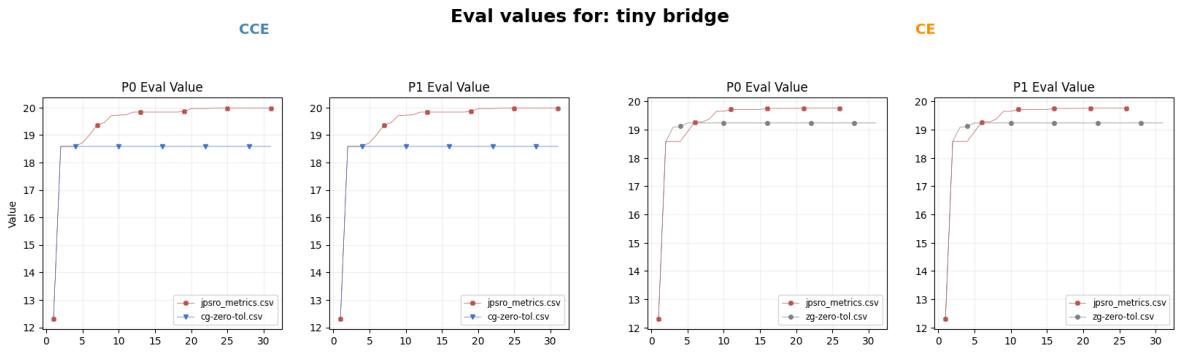


Figure 5.5: Social welfare of the meta-distribution over iterations on Tiny Bridge. JPSRO-CG reaches a high-welfare equilibrium quickly, but standard JPSRO eventually surpasses it given more iterations.

6 Discussion

6.1 Support growth and pricing hardness

A natural hope is that, when JPSRO admits a new policy and grows each $\Pi_p^{0:t}$ by one, the support S^{t+1} of the new (C)CE can be confined to $S^t \cup A_{\text{new}}$, where A_{new} is the set of joint actions involving the freshly added policy. This would give an inductive bound $|S^{t+1}| \leq |S^t| + \prod_{p \neq p^*} |\Pi_p^{0:t}|$ on support size as a function of iteration count. This is false because introducing a new policy can destabilise the previous equilibrium and make joint actions outside $S^t \cup A_{\text{new}}$ necessary. Therefore, old actions that were unused at iteration t may be strictly necessary in any (C)CE of the expanded meta-game. We give an explicit two-player counterexample in Chapter E, where adding a single action to the row player forces the unique CE onto a previously-inactive column action. The practical consequence is that the support is not bounded by the new joint actions alone, consistent with the fact that finding a CE that optimises a linear functional is NP-hard in general (Papadimitriou and Roughgarden [33]).

6.2 Limitations of column generation on the full game

A natural question is whether column generation could be applied directly to the underlying extensive-form game, bypassing the JPSRO outer loop entirely. In principle, the LPs in

Chapter 3 are well-defined when $\mathbf{A}'$ is taken to be a subset of the full joint pure-strategy space of the underlying game rather than the meta-game policy space, and the pricing subproblem (3.3) can be reformulated to search over all pure strategies. We attempted this on three-player Kuhn Poker, and although the column-generation procedure successfully maintains a small support $|\mathbf{A}'|$ that grows by one column per pricing iteration, the restricted master problem itself does not stay small.

The reason is that column generation only addresses the variable side of the LP. The constraint side is the full set of incentive-compatibility constraints of the underlying game, which must be instantiated in the RMP regardless of how few columns are active. The CE LP imposes $\sum_{i\in[n]} |\mathcal{A}_i|(|\mathcal{A}_i| - 1)$ constraints over the full game. For three-player Kuhn Poker each player has $2^{16} = 65{,}536$ pure strategies, yielding roughly 1.29×10^{10} CE constraints. The CCE setting imposes only $\sum_i |\mathcal{A}_i|$ constraints, around 2×10^5 at three players, but the same blow-up returns at four and five players. The JPSRO meta-game perspective is therefore essential; the outer loop ensures both the column space *and* the constraint set are built up incrementally over a small policy pool, so column generation supplies variable-side scalability and JPSRO supplies constraint-side scalability, and neither alone is sufficient.

6.3 Exploration versus exploitation in the training meta-solver

The empirical results expose a tension between two roles played by the training meta-solver. The first is *exploitation*, producing a meta-distribution against which best responses sharpen the current policy pool by extracting the most profitable deviation from a concentrated target. The second is *exploration*, producing a meta-distribution that is diverse enough over the joint action space that the best-response oracle is forced to discover qualitatively different policies rather than refining the same region of policy space. Standard JPSRO with the MGCE/MGCCE training meta-solver leans heavily on exploration, since the Gini objective explicitly maximises spread across the (C)CE polytope. JPSRO-CG leans, by construction, toward exploitation; the pricing rule admits a column with the largest reduced cost, which corresponds to an extremal vertex of the polytope rather than a diffuse interior point, and the support S that

the column-generation procedure maintains is therefore typically much sparser than the dense MGCE optimum.

This trade-off is reflected in the welfare results of the previous chapter. On Trade Comm and Sheriff under the CE objective, JPSRO with MGCE consistently recovers the welfare-optimal CE; on the same games JPSRO-CG also recovers the MWCE. Under the CCE objective, however, JPSRO with MGCCE recovers the welfare-optimal equilibrium while JPSRO-CG plateaus at a strictly worse value. The mechanism behind the JPSRO success is that the diffuse Gini-maximising distribution forces the BR oracle to probe many parts of the joint space, increasing the chance that some best response seeds a high-welfare equilibrium. JPSRO-CG instead concentrates probability mass on a small support S, and best responses against a concentrated σ^t collapse onto whichever opponent strategies happen to be active in S. Whenever the welfare-optimal CCE is supported on a region of policy space disjoint from the columns CG has chosen to admit, the algorithm has no mechanism to escape. Pricing only adds reduced-cost-improving columns, not welfare-improving ones, and the outer BR loop is fed a distribution that lacks the diversity needed to pull the policy population in a new direction.

CE versus CCE best-response structure. The CCE BR returns a single policy per player against the marginal σ^t_{-p}, producing exactly n new candidates per iteration regardless of support. The CE BR is conditional, returning the optimal deviation per recommendation $\pi_p \in \text{supp}(\sigma^t_p)$, and therefore generates up to $|\Pi^{0:t}_p|$ candidates per iteration. The per-iteration multiplicity gives CE a built-in source of exploration that partly compensates for the lost MGCE diffusion signal, and matches the empirical pattern in Chapter 5; JPSRO-CG recovers the MWCE on Sheriff and Trade Comm because the conditional BR keeps injecting fresh policies into the population even when σ^t has small support. Under CCE, the concentrated σ^t collapses the single marginalised BR onto a policy already in the population, the deviation incentive falls to zero, and JPSRO terminates at a welfare-suboptimal CCE.

6.4 The equilibrium selection problem

The pricing subproblem in (3.6) and (3.3) involves an RMP that optimises a linear objective subject to linear (C)CE incentive constraints. Both are linear programs, and linear programs admit non-unique optima whenever the optimal face of the feasible polytope consists of more than a single vertex. In equilibrium terms this is the *equilibrium selection problem* [17]; the (C)CE polytope of a given meta-game typically contains a continuum of equilibria with the same value of any given linear objective, and the LP itself provides no canonical mechanism to prefer one over another.

This has a direct consequence for column generation. When the RMP's optimal dual face is non-singleton, multiple primal–dual pairs (σ^t, λ^t) are simultaneously optimal, each inducing a different reduced-cost vector for the pricing subproblem. A column that is reduced-cost-optimal under one choice of λ^t may have strictly larger reduced cost under another, and similarly the pricing subproblem itself can return different columns from the optimal face under arbitrary tie-breaking rules of the underlying solver. Different paths of column admission expand the support S in different directions, so two runs of CG that are mathematically equivalent at every iteration in terms of LP optimality can still converge to different equilibria of the same meta-game. The procedure may therefore admit columns that are short-term reduced-cost maximisers but long-term inefficient, in the sense that a different tie-breaking choice at an earlier iteration would have led the welfare CG stage to an equilibrium with smaller support.

6.5 Improvements

6.5.1 Stabilising the meta-distribution

The exploration–exploitation analysis of Chapter 6 and the equilibrium-selection discussion both point at the same underlying phenomenon; between consecutive CG iterations, the optimal dual face of the RMP can shift, the admitted column flips, and the support S jumps non-monotonically across the polytope. The standard remedies all attempt to prevent the dual

vector λ^t from making large excursions between iterations.

Proximal stabilisation. du Merle et al. [10] introduce a piecewise-linear penalty on deviations of λ^t from a reference dual, which we expect would directly damp the support oscillation observed on Sheriff and Trade Comm. Rousseau et al. [34] show that an interior-point dual selector achieves a similar effect without explicit penalties, by returning a centroid of the optimal dual face rather than an arbitrary vertex. Both modifications are local to the RMP and would compose with the JPSRO outer loop without changing the convergence guarantee.

Multi-column admission and welfare-aware pricing. Admitting a batch of high-reduced-cost columns per pricing round, rather than a single column, reduces the number of outer iterations needed to populate S and trades smoothly against per-iteration cost. Beyond hand-tuned batch sizes, a parallel strand of work seeks to accelerate column generation itself through learning. Hu et al. [19] train a reinforcement-learning policy to admit a variable number of columns per iteration, reporting large reductions in iteration count on cutting-stock and vehicle-routing benchmarks. Porting this idea to JPSRO-CG would treat the per-iteration column budget as a learned function of the RMP state, with the dual vector λ^t and the current support S as natural inputs.

6.5.2 Approximate best-response oracles

The empirical results in Chapter 5 show that, once column generation has reduced the RMP to a small support, the dominant cost in each outer iteration is the best-response oracle itself; the five-player Kuhn Poker run failed not because the RMP was intractable but because the exact BR traversal of the game tree was. The original JPSRO paper (Marris et al. [26]) already notes that exact best responses are tractable only for small games. The per-player pure strategy count of $2^{(n+1)2^{n-1}}$ for n-player Kuhn Poker, which reaches 2^{96} at $n = 5$, makes this the binding constraint for any scaling work.

Reinforcement-learning best responses, as used in PSRO (Lanctot et al. [22]), train an approximate BR by running a single-agent RL algorithm against the frozen meta-distribution.

Regret-minimisation best responses provide a closely related alternative for extensive-form games, replacing the RL inner loop with no-regret learning over the game tree. Either oracle introduces a second source of approximation in addition to the ϵ-approximate meta-solver studied in Chapter 4; the ϵ-(C)CE convergence theorem (Theorems 3 and 4) assumes an exact best response, and extending it to a setting in which both the meta-solver and the BR oracle are approximate is a natural follow-up.

6.5.3 Further directions

Exploiting structure in the pricing subproblem. Pricing (3.3) is NP-hard in general (Papadimitriou and Roughgarden [33]), but structural assumptions on the game speed it up without changing the framework. The most direct case is *graphical games*, where utilities decompose into per-neighbourhood potentials and the pricing objective inherits the same decomposition, admitting both loopy belief propagation and exact ILP treatments (Chapter D).

Extension to team settings. Team-PSRO (McAleer et al. [27]), which replaces the per-player BR with a cooperative-RL joint oracle and connects to ex-ante team correlation (Farina et al. [12]), is a natural target for the same CG machinery; the pricing subproblem is unchanged, but the column space partitions into team-coordinated columns. Each such column specifies one joint policy for an entire team rather than independent per-player policies, so the meta-distribution correlates teammates ex ante and the CG search ranges over joint team plans instead of single-player deviations.

Engineering improvements. Warm-starting CG from the previous S and λ^{t-1} across outer iterations (Vanderbeck [38]) would amortise early-iteration cost, and parallelising the independent per-player BR and per-row pricing computations would yield a near-linear speedup in the BR-dominated regime.

7 Conclusion

Column generation supplies the variable-side scalability and JPSRO's outer loop supplies the constraint-side scalability, and neither alone is sufficient on the underlying extensive-form game. JPSRO-CG combines the two; it embeds column generation into the JPSRO outer loop in order to compute (coarse) correlated equilibria in n-player general-sum extensive-form games at scales where the full-LP meta-solver becomes intractable.

On the theoretical side, we proved polynomial upper bounds on the minimum support of any optimal (C)CE under a linear objective (Theorems 1 and 2), and generalised JPSRO's convergence guarantee to an ϵ-approximate meta-solver (Theorems 3 and 4), recovering the original result at $\epsilon = 0$. Empirically, JPSRO-CG matched JPSRO's exploitability trajectory on a much smaller support across Kuhn Poker, Sheriff, Trade Comm, and Tiny Bridge, converged to a CCE on 4-player Kuhn Poker within a time budget on which no full-LP variant did, and recovered the welfare-optimal CE on Sheriff and Trade Comm. Under the CCE welfare objective, the algorithm plateaus at a suboptimal value because the marginalised CCE best response depends on diversity in the meta-distribution that column generation, by concentrating mass on extremal vertices, does not provide.

With column generation and JPSRO combined, the binding cost shifts away from the meta-solver and onto the best-response oracle itself, as the failed 5-player Kuhn Poker runs make explicit. The natural next step is therefore to combine an approximate best-response oracle with the ϵ-meta-solver framework developed here, together with dual stabilisation and welfare-aware pricing to close the remaining CCE welfare gap. The structural extensions to graphical games (Chapter D) and team settings outlined in Chapter 6 are the most direct routes to richer game classes.

Bibliography

[1] Nick Abou Risk and Duane Szafron. Using counterfactual regret minimization to create competitive multiplayer poker agents. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '10): Volume 1*, AAMAS '10, pages 159–166, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

[2] Robert J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(1):67–96, 1974. doi: 10.1016/0304-4068(74)90037-8.

[3] David Avis, Gabriel D. Rosenberg, Rahul Savani, and Bernhard von Stengel. Enumeration of nash equilibria for two-player games. *Economic Theory*, 42(1):9–37, 2009. doi: 10.1007/s00199-009-0449-x.

[4] Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh, and Pamela H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998. doi: 10.1287/opre.46.3.316.

[5] Ariyan Bighashdel, Yongzhao Wang, Stephen McAleer, Rahul Savani, and Frans A. Oliehoek. Policy space response oracles: A survey. *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 7951–7961, 2024. doi: 10.24963/ijcai.2024/880.

[6] Andrea Celli, Stefano Coniglio, and Nicola Gatti. Computing optimal ex ante correlated equilibria in two-player sequential games. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019)*, pages 909–917. IFAAMAS, 2019.

[7] George B. Dantzig and Philip Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1):101–111, 1960. doi: 10.1287/opre.8.1.101.

[8] C. Daskalakis, P. Goldberg, and C. Papadimitriou. The complexity of computing a nash equilibrium. *SIAM J. Comput.*, 39:195–259, 2 2009.

[9] Guy Desaulniers, Jacques Desrosiers, and Marius M. Solomon, editors. *Column Generation*. Springer, New York, 2005. doi: 10.1007/b135457.

[10] Olivier du Merle, Daniel Villeneuve, Jacques Desrosiers, and Pierre Hansen. Stabilized column generation. *Discrete Mathematics*, 194(1–3):229–237, 1999. doi: 10.1016/S0012-365X(98)00213-1.

[11] Gabriele Farina, Tommaso Bianchi, and Tuomas Sandholm. Coarse correlation in extensive-form games, 2019. URL `https://arxiv.org/abs/1908.09893`.

[12] Gabriele Farina, Andrea Celli, Nicola Gatti, and Tuomas Sandholm. Connecting optimal ex-ante collusion in teams to extensive-form correlation: Faster algorithms and positive complexity results. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3164–3173. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/farina21a.html`.

[13] Richard Gibson. *Regret minimization in games and the development of champion multiplayer computer poker-playing agents*. Ph.d. dissertation, University of Alberta, Edmonton, AB, Canada, 2014.

[14] Itzhak Gilboa and Eitan Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1(1):80–93, 1989. doi: 10.1016/0899-8256(89)90006-7.

[15] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6):849–859, 1961. doi: 10.1287/opre.9.6.849.

[16] Paul W. Goldberg and Aaron Roth. Bounds for the query complexity of approximate equilibria. *ACM Transactions on Economics and Computation*, 4:1–25, 2016. doi: 10.1145/2956582.

[17] John C. Harsanyi and Reinhard Selten. *A General Theory of Equilibrium Selection in Games*, volume 1. The MIT Press, Cambridge, MA, 1 edition, 1988.

[18] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000. doi: 10.1111/1468-0262.00153. URL https://doi.org/10.1111/1468-0262.00153.

[19] Yi-Xiang Hu, Feng Wu, Shaoang Li, Yifang Zhao, and Xiang-Yang Li. Ffcg: Effective and fast family column generation for solving large-scale linear program, 2024. URL https://arxiv.org/abs/2412.19066.

[20] H. W. Kuhn. A simplified two-person poker. In *Contributions to the Theory of Games*, volume 1, pages 97–103. 1950.

[21] M. Lanctot. Further developments of extensive-form replicator dynamics using the sequence-form representation. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 2, 05 2014. URL https://www.ifaamas.org/Proceedings/aamas2014/aamas/p1257.pdf.

[22] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 4190–4203, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3956340669ee3499427b68630737190b-Abstract.html.

[23] Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas Anthony, Edward Hughes, Ivo Danihelka, and Jonah Ryan-Davis. OpenSpiel: A framework for reinforcement learning in games, 2019. URL https://arxiv.org/abs/1908.09453.

[24] Marco E. Lübbecke. Column generation. In *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, 2010. doi: 10.1002/9780470400531.eorms0158.

[25] Marco E. Lübbecke and Jacques Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005. doi: 10.1287/opre.1050.0234.

[26] Luke Marris, Paul Muller, Marc Lanctot, Karl Tuyls, and Thore Graepel. Multi-agent training beyond zero-sum with correlated equilibrium meta-solvers. 2021. URL `https://arxiv.org/abs/2106.09435`.

[27] Stephen McAleer, Gabriele Farina, Marc Lanctot, and Tuomas Sandholm. Team-psro for learning approximate TMECor in large team games via cooperative reinforcement learning. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023.

[28] Hervé Moulin and Jean-Philippe Vial. Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon. *International Journal of Game Theory*, 7(3-4):201–221, 1978.

[29] Paul Muller, Shayegan Omidshafiei, Mark Rowland, Karl Tuyls, Julien Perolat, Siqi Liu, Daniel Hennes, Luke Marris, Marc Lanctot, Edward Hughes, Zhe Wang, Guy Lever, Nicolas Heess, Thore Graepel, and Remi Munos. A generalized training approach for multiagent learning. *arXiv preprint arXiv:1909.12823*, 2019. doi: 10.48550/arXiv.1909.12823.

[30] John Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

[31] R. Nau, S. G. Canovas, and P. Hansen. On the geometry of nash equilibria and correlated equilibria. *International Journal of Game Theory*, 32(4):443–453, aug 2004.

[32] Shayegan Omidshafiei, Christos Papadimitriou, Georgios Piliouras, Karl Tuyls, Mark Rowland, Jean-Baptiste Lespiau, Wojciech M. Czarnecki, Marc Lanctot, Julien Perolat, and Remi Munos. α-rank: Multi-agent evaluation by evolution. *arXiv preprint arXiv:1903.01373*, 2019. doi: 10.48550/arXiv.1903.01373.

[33] Christos H. Papadimitriou and Tim Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM (JACM)*, 2008. URL `https://www.timroughgarden.org/papers/cor.pdf`. Preliminary versions appeared in SODA 2005 and STOC 2005.

[34] Louis-Martin Rousseau, Michel Gendreau, and Dominique Feillet. Interior point stabilization for column generation. *Operations Research Letters*, 35(5):660–668, 2007. doi: 10.1016/j.orl.2006.11.004.

[35] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Chichester, 1986.

[36] Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: Opponent modelling in poker. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 550–558, 2005.

[37] David Sychrovský, Alberto Solinas, Revan MacQueen, James R. Wright, Dustin Morrill, and Michael Bowling. Approximating nash equilibria in general-sum games via meta-learning. *arXiv preprint arXiv:2504.18868*, 2025. URL `https://arxiv.org/abs/2504.18868`. License: CC BY 4.0.

[38] François Vanderbeck. Implementing mixed integer column generation. *Column Generation*, pages 331–358, 2005. doi: 10.1007/0-387-25486-2_12.

[39] Michael Winsper and Maria Chli. Decentralized supply chain formation using max-sum loopy belief propagation. *Computational Intelligence*, 27, 2011.

[40] Brian Hu Zhang, Gabriele Farina, Andrea Celli, and Tuomas Sandholm. Optimal correlated equilibria in general-sum extensive-form games: Fixed-parameter algorithms, hardness, and two-sided column-generation. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pages 1011–1011, New York, NY, USA, 2022. Association for Computing Machinery. doi: 10.1145/3490486.3538330. URL `https://doi.org/10.1145/3490486.3538330`.

A Notation

The symbols used throughout this dissertation are summarised in Table A.1. Symbols introduced for the underlying game in Chapters 2 and 3 carry over to the JPSRO meta-game in Chapters 4 and 5 under the normal-form reduction (Marris et al. [26]): $a_i \mapsto \pi_i$, $\mathbf{a} \mapsto \pi$, $\mathcal{A}_i \mapsto \Pi_i^{0:t}$, $\mathbf{A} \mapsto \Pi^{0:t}$, with $\lambda^t_{i,\pi_i,\pi_i'}$ playing the role of $\lambda_{i,a_i,a_i'}$.

Table A.1: Notation summary.

Symbol	Meaning
Players and actions	
n	Number of players.
$[n]$	Player index set $\{1, \ldots, n\}$.
i, p	A player index.
$-i$	Indices of all players except i.
$\mathcal{A}_i$	Action set of player i.
$\mathbf{A} = \bigotimes_i \mathcal{A}_i$	Joint action space.
$a_i, a_i' \in \mathcal{A}_i$	An action and a deviation action of player i.
$a_{-i} \in \mathcal{A}_{-i}$	Joint action of all players except i.
$\mathbf{a} = (a_1, \ldots, a_n) \in \mathbf{A}$	A joint action.
$u_i(\mathbf{a})$	Utility of player i at joint action $\mathbf{a}$.
$u_i(x) = \mathbb{E}_{\mathbf{a}\sim x}[u_i(\mathbf{a})]$	Expected utility of i under distribution x.
Distributions and equilibria	
$x \in \Delta(\mathbf{A})$	Joint distribution over actions.
$\text{supp}(x)$	Support set of distribution x.

continued on next page

Table A.1 (continued)

Symbol	Meaning
$f(x), f(\mathbf{a})$	Linear objective on a distribution or joint action (e.g. social welfare).
ϵ	Approximation tolerance for an ϵ-(C)CE.
CE, CCE	(Coarse) correlated equilibrium; see Eq. (2.2).
NE	Nash equilibrium.
LP and column generation	
$\mathbf{A}' \subseteq \mathbf{A}$	Active support over which the RMP is instantiated.
$\lambda_{i,a_i,a_i'}$	Dual on the (C)CE incentive constraint indexed by (i, a_i, a_i').
μ	Dual on the normalisation constraint $\sum_{\mathbf{a}} x(\mathbf{a}) = 1$.
$\bar{c}(\mathbf{a}^\dagger)$	Reduced cost of candidate column $\mathbf{a}^\dagger$.
$\mathbf{a}^\dagger, \mathbf{a}^*$	A candidate column and the pricing-optimal column.
RMP	Restricted master problem.
CG	Column generation.
JPSRO and JPSRO-CG meta-game	
t	Outer JPSRO iteration index.
$\Pi_p^{0:t}$	Policy pool of player p at iteration t.
$\Pi^{0:t} = \bigotimes_p \Pi_p^{0:t}$	Joint meta-policy space at iteration t.
$\pi_p, \pi_p' \in \Pi_p^{0:t}$	A policy of player p and a deviation policy.
$\pi \in \Pi^{0:t}$	A joint policy.
$G^{0:t}$	Empirical meta-game at iteration t.
σ^t	Meta-distribution over $\Pi^{0:t}$ at iteration t.
Δ_p^t	Best-response deviation value of player p at iteration t.

continued on next page

Table A.1 (continued)

Symbol	Meaning
$\lambda^t_{i,\pi_i,\pi'_i}$	Meta-game analogue of λ_{i,a_i,a'_i} at iteration t.
ER	Expected-return map from joint policies to meta-game payoffs.
BR	Best-response oracle.
MS, MS_W	Meta-solver and welfare-stage meta-solver.
S	Active joint-policy support maintained by JPSRO-CG.
$N = \prod_p \lvert\Pi_p^{0:t}\rvert$	Size of the empirical joint policy space.
K	Number of (C)CE incentive constraints in the RMP.
Equilibrium objectives	
MWCE, MWCCE	Maximum-welfare (C)CE.
MGCE, MGCCE	Maximum-Gini (C)CE [26].
NFCE, NFCCE	Normal-form (C)CE; the solution concept JPSRO targets.

B Experimental Setup Details

B.1 Game parameters

Table B.1 summarises the games and the parameters passed to `pyspiel.load_game`. All games are loaded directly from OpenSpiel's game catalogue with no modifications to payoffs or information structure; the parameter columns reproduce the keyword arguments exactly as they appear in the driver scripts.

OpenSpiel name	Players	Type	Parameters
`kuhn_poker`	3	Zero-sum	`players=3`
`kuhn_poker`	4	Zero-sum	`players=4`
`trade_comm`	2	General-sum	`num_items=3`
`trade_comm`	2	General-sum	`num_items=4`
`sheriff`	2	General-sum	`item_penalty=1.0`, `item_value=5.0`, `max_bribe=2`, `max_items=10`, `num_rounds=2`, `sheriff_penalty=1.0`
`tiny_bridge_2p`	2	General-sum	(defaults)

Table B.1: Games and the `pyspiel.load_game` parameters used in all experiments. 5-player Kuhn Poker was attempted but exhausted memory on every configuration and is omitted from the reported results.

B.2 Solver and algorithm parameters

Table B.2 lists the JPSRO outer-loop hyperparameters that are shared across all configurations. These match the defaults from the reference JPSRO implementation; only the meta-solver and target equilibrium are varied across the configurations in Section 4.3.2. Table B.3 lists parameters specific to the column-generation meta-solvers, and Table B.4 lists the numerical tolerances used by the LP solver.

Parameter	Value
`iterations` (max outer iterations)	30
Wall-clock budget per run	1800 s (30 min)
`policy_init`	`uniform`
`update_players_strategy`	`all`
`br_selection`	`largest_gap`
`action_value_tolerance` (BR oracle)	-1.0 (exact)
`prob_cut_threshold` (BR oracle)	0.0
`ignore_repeats`	`False`
`DIST_TOL`	10^{-8}
`GAP_TOL`	10^{-8}
`RETURN_TOL`	10^{-12}

Table B.2: JPSRO outer-loop parameters, shared across all configurations. Runs terminate when either the iteration cap or the wall-clock budget is reached.

Parameter	Value
`target_epsilon` (zero-tolerance CG)	0
`target_epsilon` (ε-approximate CG)	0.01 (scaled by $\max_j \bar{A}_j$)
`max_cg_iterations` (gap-minimisation phase)	500
`max_cg_iterations` (welfare phase)	500
Reduced-cost stopping threshold (gap phase)	10^{-4}
Reduced-cost stopping threshold (welfare phase, CE)	10^{-4}
Reduced-cost stopping threshold (welfare phase, CCE)	10^{-4}
Initial support S_0	uniform-initialised joint policy
Cross-iteration warm start	persistent support mask

Table B.3: Column-generation hyperparameters used in JPSRO-CG. The pricing tolerance is the smallest positive reduced cost that will trigger a new column addition; below it CG terminates.

Parameter	Value
Solver	ECOS (via `cvxpy`)
`max_iters`	10^8
`abstol`	10^{-7}
`reltol`	10^{-7}
`feastol`	10^{-7}
`abstol_inacc`	10^{-7}
`reltol_inacc`	10^{-7}
`feastol_inacc`	10^{-7}
Constraint zero-tolerance	10^{-8}

Table B.4: Numerical tolerances passed to the LP solver. The same tolerances are used for the restricted master problem and for any full-support LP solved in standard JPSRO or in full-support evaluation.

B.3 Compute resources

Runs were carried out on a mix of two environments: a local desktop with an AMD Ryzen 5 7600 (6 cores / 12 threads) and 32 GB of RAM, and Google Colab Pro high-RAM instances. All runs are CPU-only; no GPU is used.

B.4 Experiment matrix

Table B.5 summarises which meta-solver configurations were run on which games. Each cell corresponds to a single run with the parameters above. The CE and CCE objectives are run independently in every cell.

Configuration	Kuhn 3p	Kuhn 4p	Trade 3	Trade 4	Sheriff	Tiny Bridge
JPSRO: MGCE / MGCCE	✓	✓	✓	✓	✓	✓
JPSRO: Approx-MGCE / -MGCCE	✓	✓	×	×	✓	×
JPSRO: Approx-MWCE / -MWCCE	✓	✓	×	×	✓	×
JPSRO-CG: restricted-support	✓	✓	×	×	×	×
JPSRO-CG: full-support	✓	✓	×	×	×	×
JPSRO-CG: zero-tolerance	✓	✓	✓	✓	✓	✓

Table B.5: Experimental matrix. Each ✓ denotes a CE run and a CCE run (two separate processes). The zero-tolerance configuration is paired with the welfare CG phase (Figure 4.2) on the general-sum games (Sheriff, Trade Comm, Tiny Bridge); on zero-sum Kuhn Poker the welfare phase is disabled. 5-player Kuhn Poker was attempted but ran out of memory on every configuration and is omitted.

B.5 Logged metrics and output format

Each run appends one row per JPSRO outer iteration to a CSV file with the columns listed below. The same schema is used for both JPSRO and JPSRO-CG runs; columns that are not meaningful for a given configuration (e.g., `Train_Support_Size` for standard JPSRO) are populated with the full policy-product size so that all runs remain comparable on the same axes.

- `Iteration`: outer JPSRO iteration index.
- `P{p}_Train_Value`, `P{p}_Train_Gap`: per-player expected value and (C)CE gap under the training meta-distribution.
- `P{p}_Eval_Value`, `P{p}_Eval_Gap`: per-player expected value and (C)CE gap under the evaluation meta-distribution.

- `P{p}_Num_Policies`: total number of policy copies for player p (sum of repeats).
- `P{p}_Unique_Policies`: $|\Pi_p^{0:t}|$, the number of unique policies in player p's pool.
- `Train_Support_Size`: $|S|$ for JPSRO-CG; $\prod_p |\Pi_p^{0:t}|$ for standard JPSRO.
- `Eval_Support_Size`: $\prod_p |\Pi_p^{0:t}|$, the full policy-product size.
- `Elapsed_Time_s`: cumulative wall-clock time since the start of the run.

The convergence plots in Chapter C are produced directly from these CSV logs.

C Results

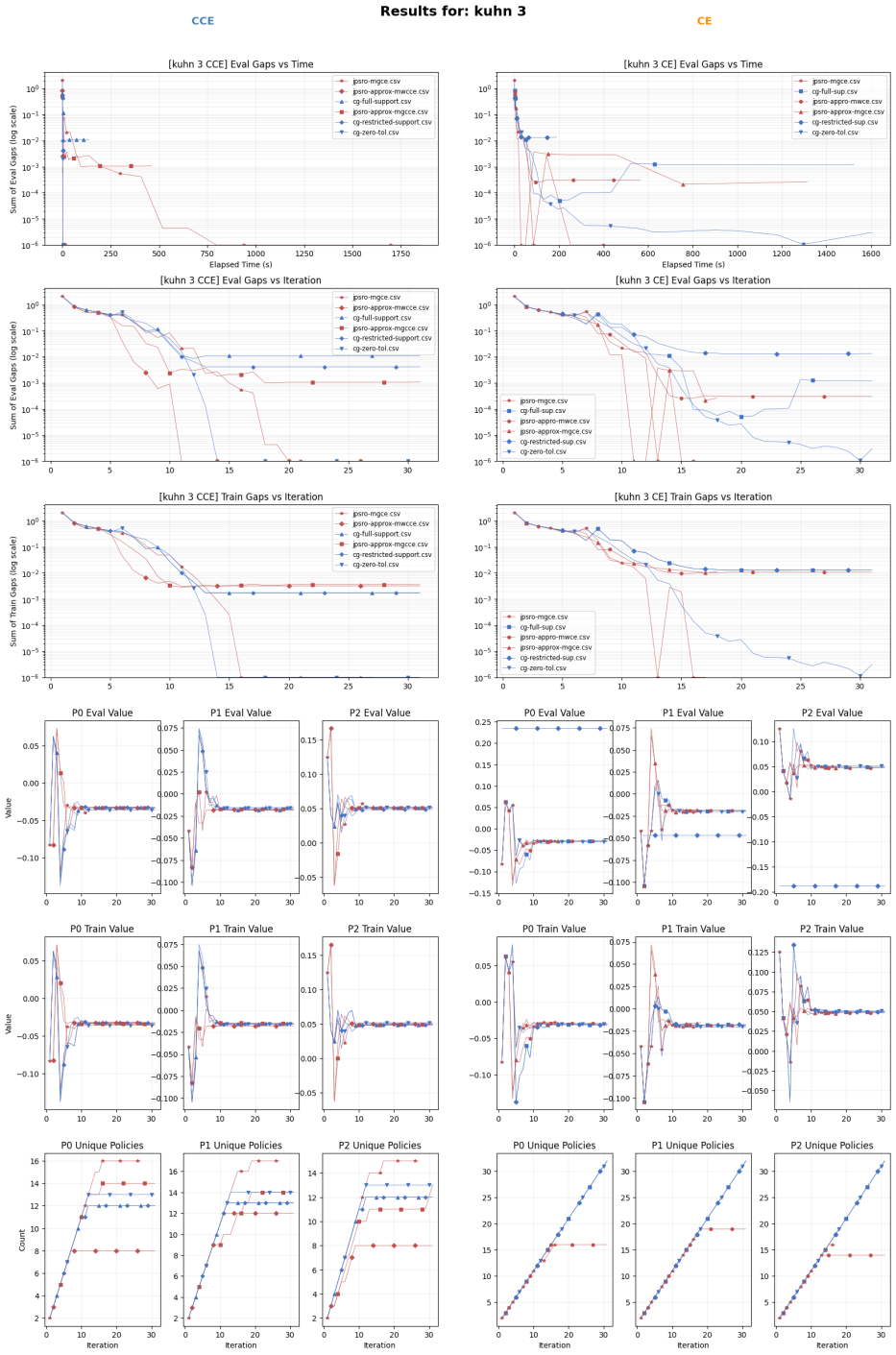


Figure C.1: Kuhn 3

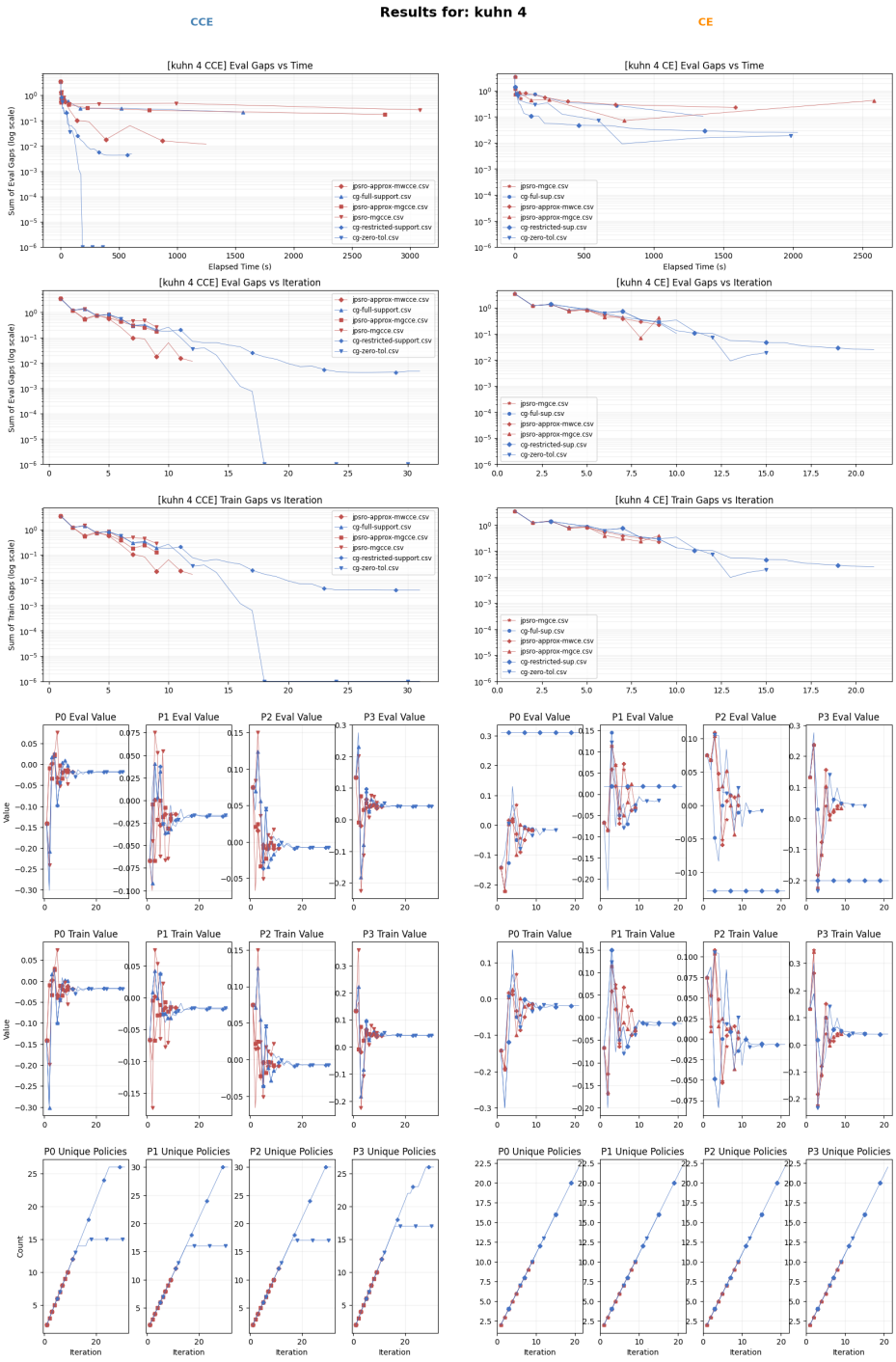


Figure C.2: Kuhn 4

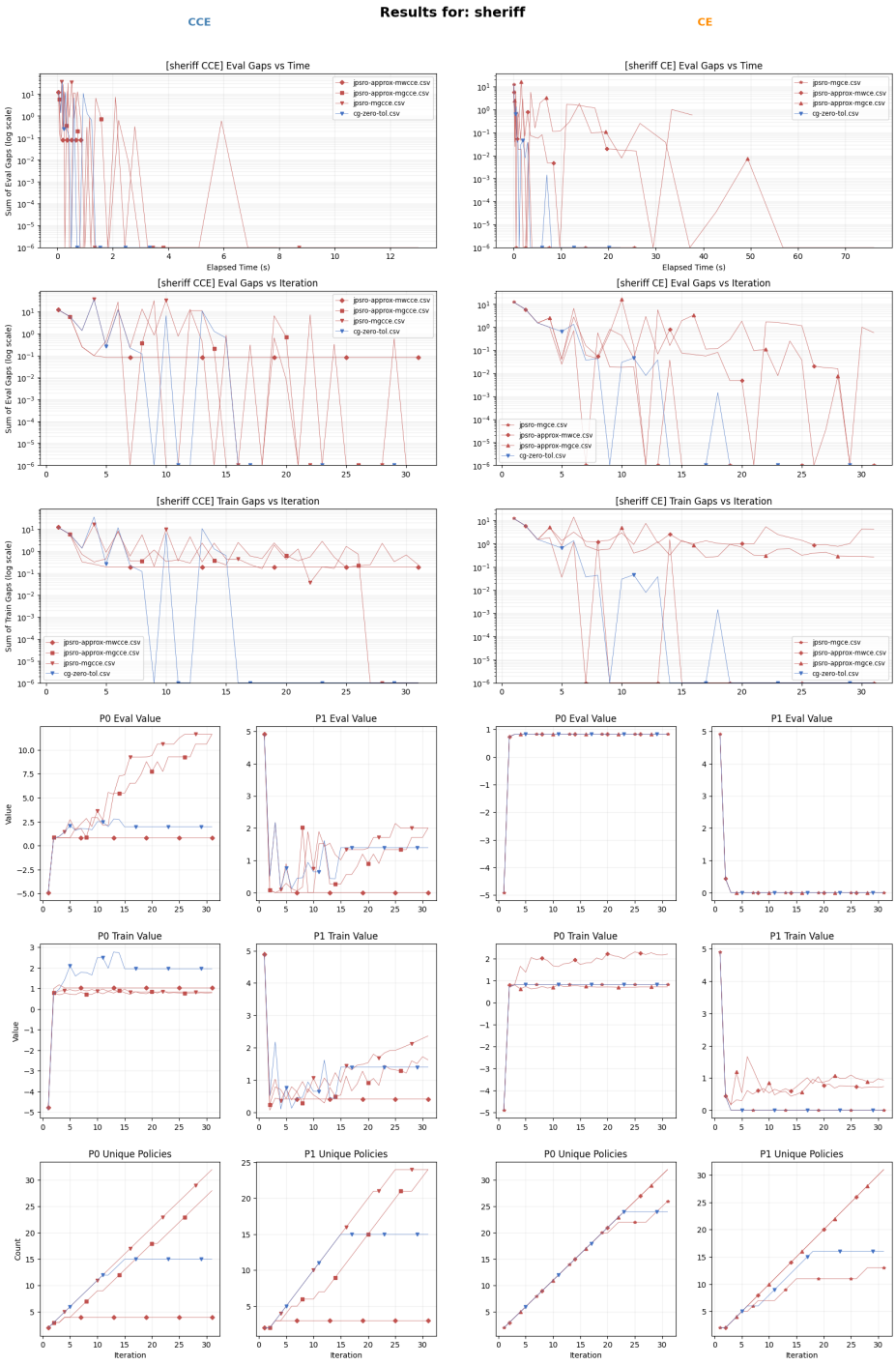


Figure C.3: Sheriff

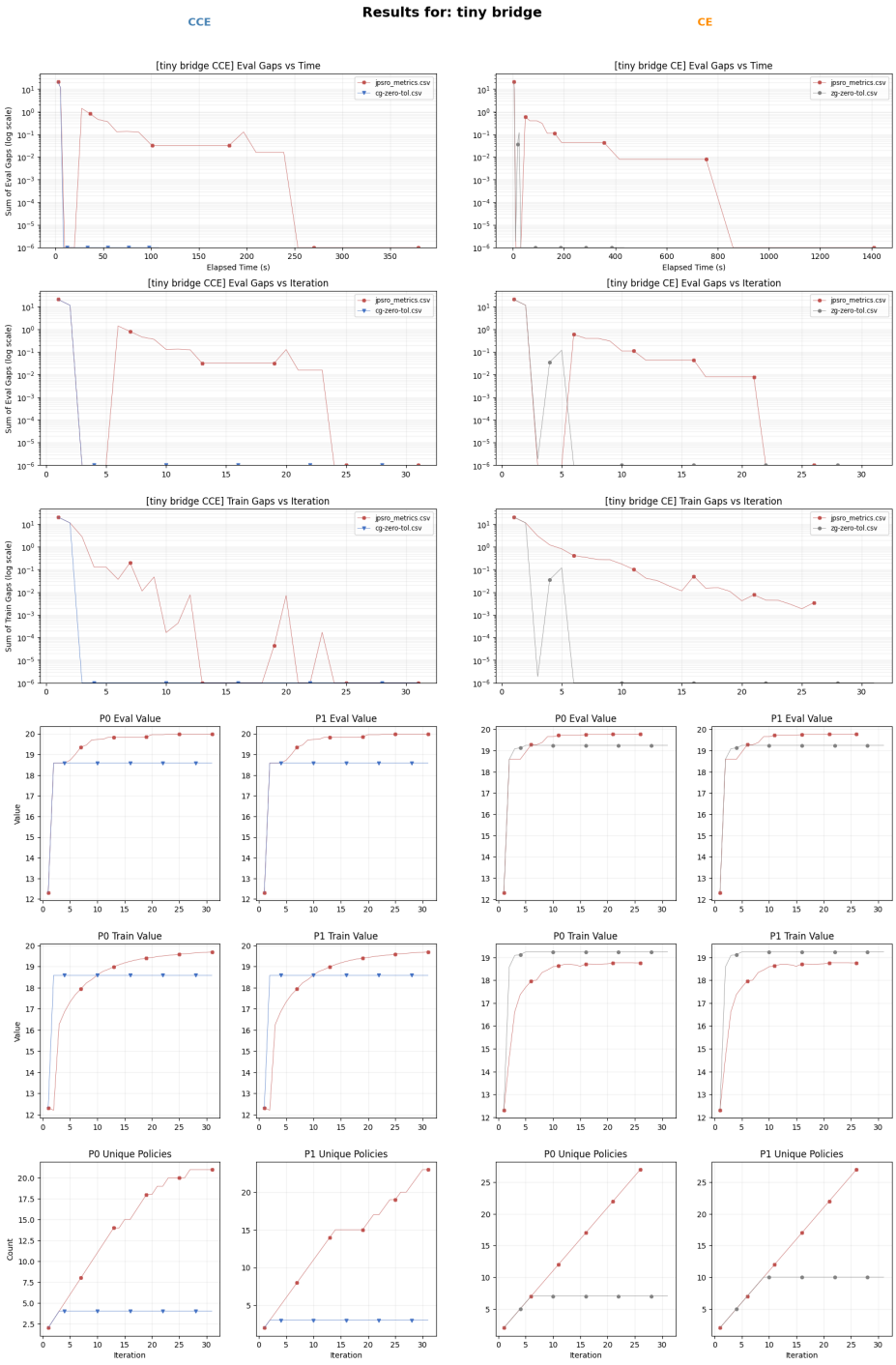


Figure C.4: Tiny Bridge

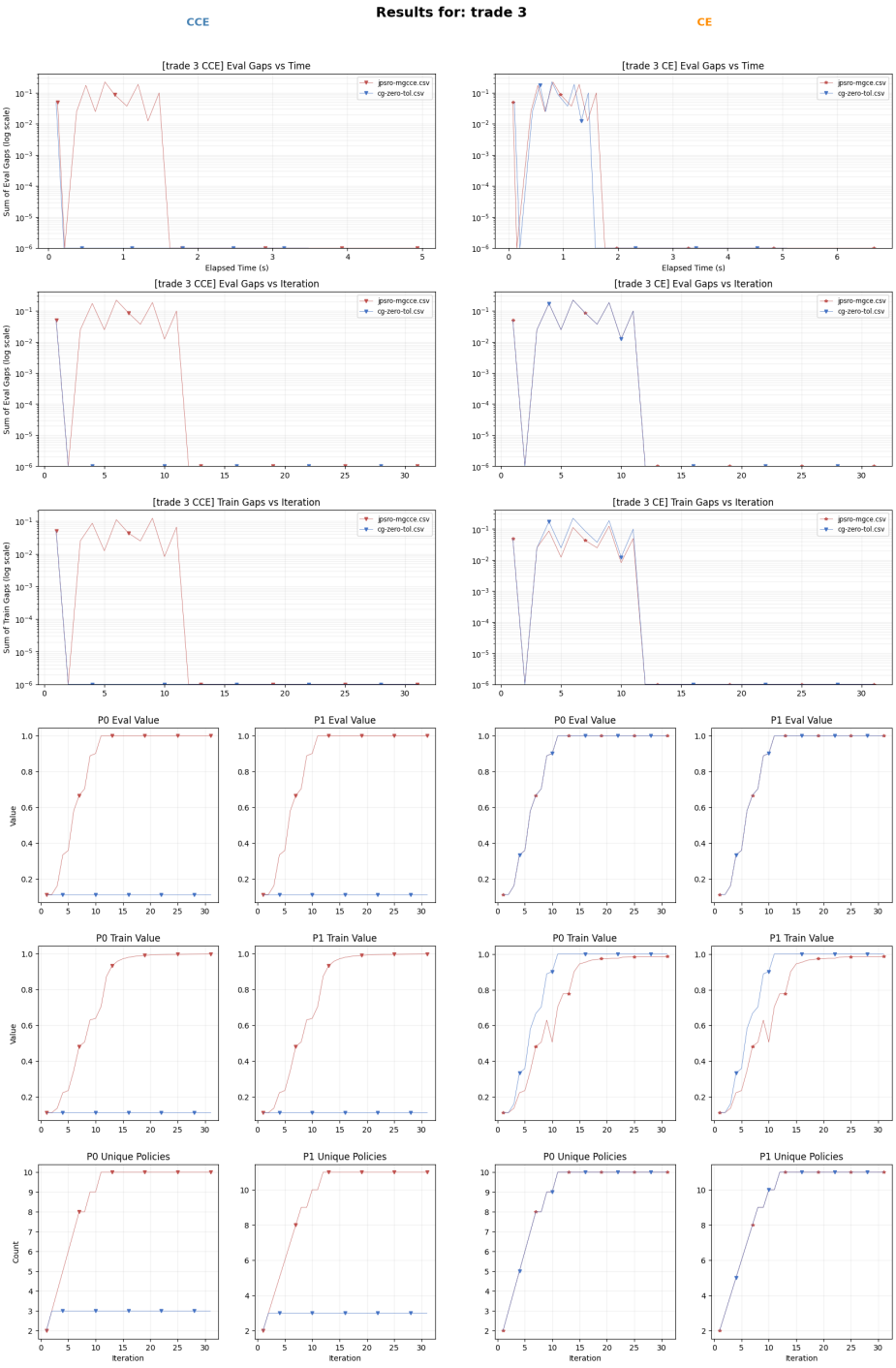


Figure C.5: Trade 3

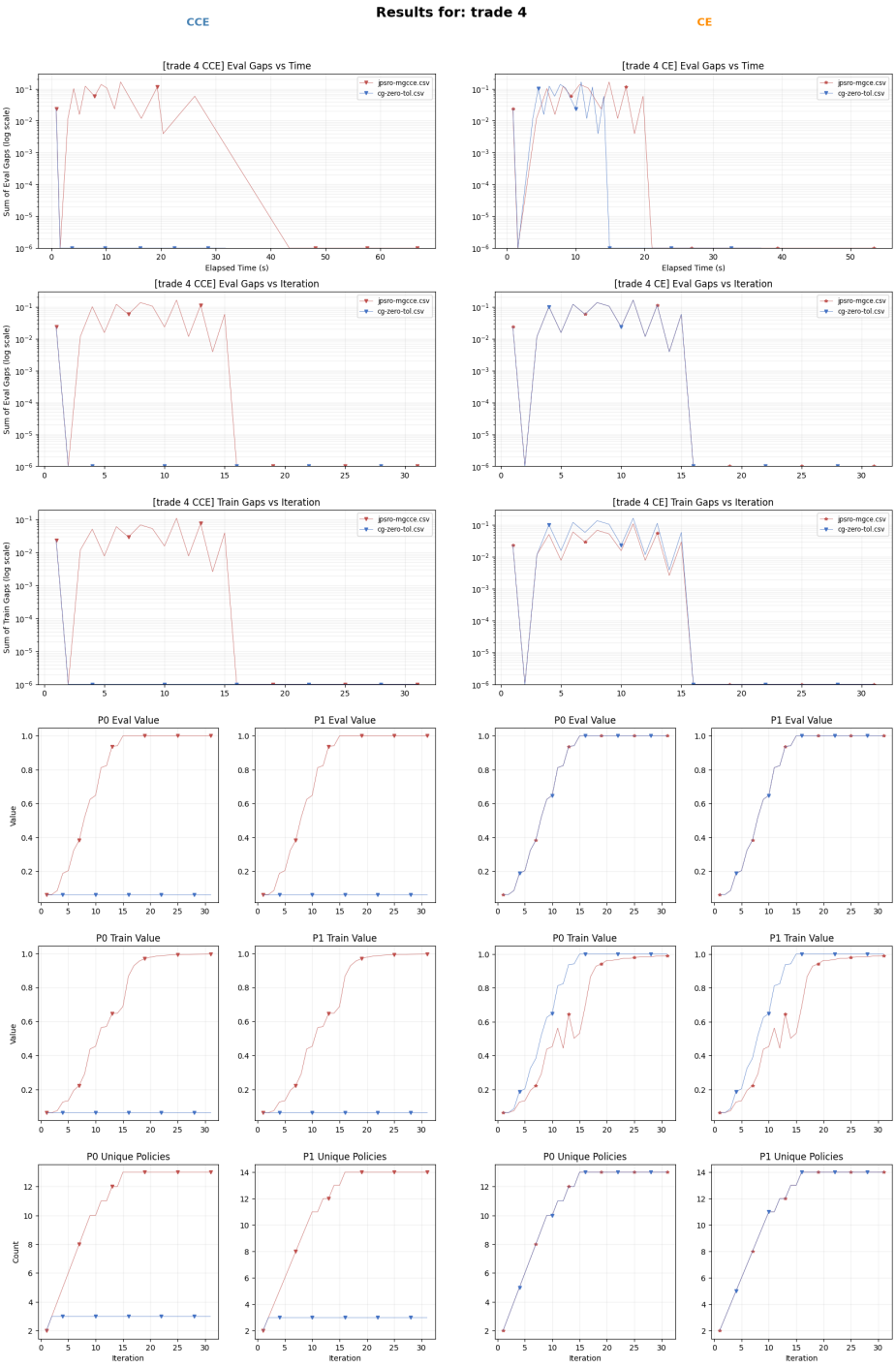
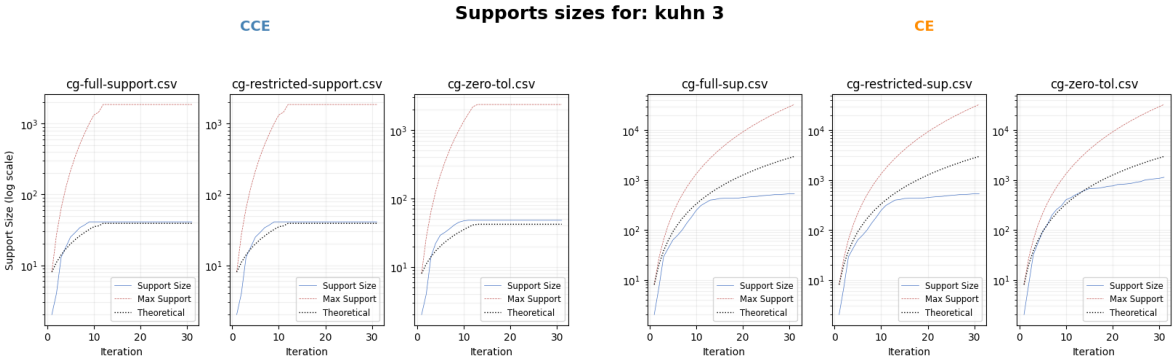
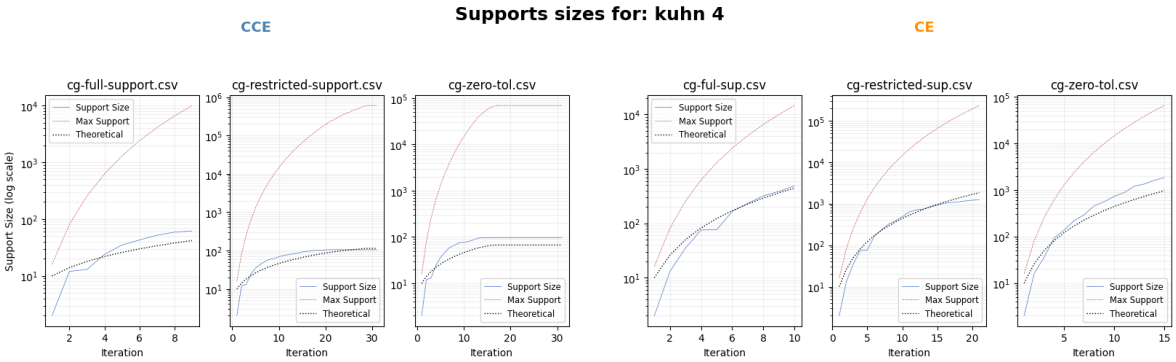


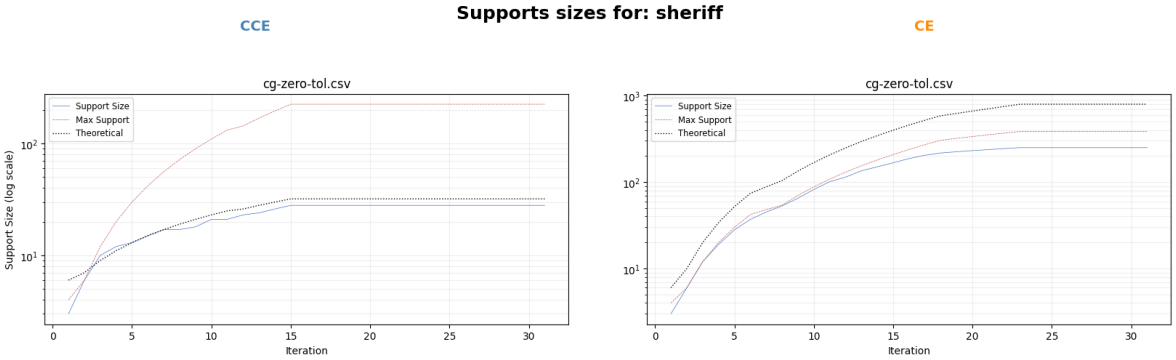
Figure C.6: Trade 4



(a) Kuhn 3

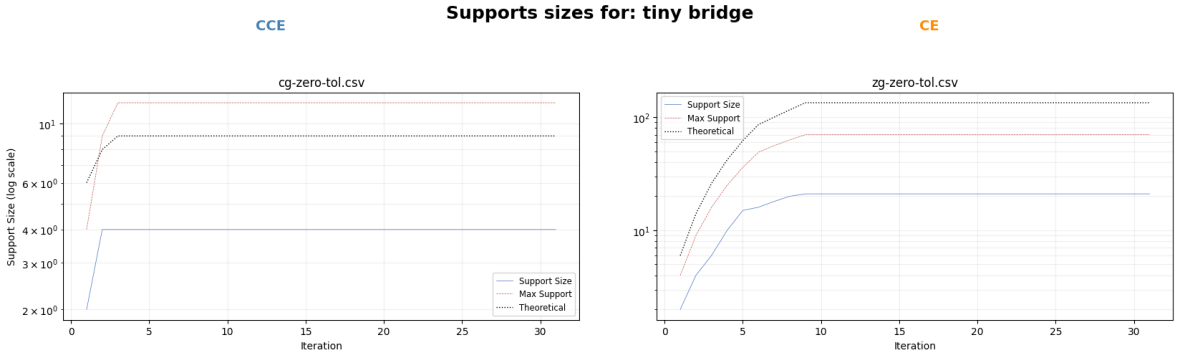


(b) Kuhn 4

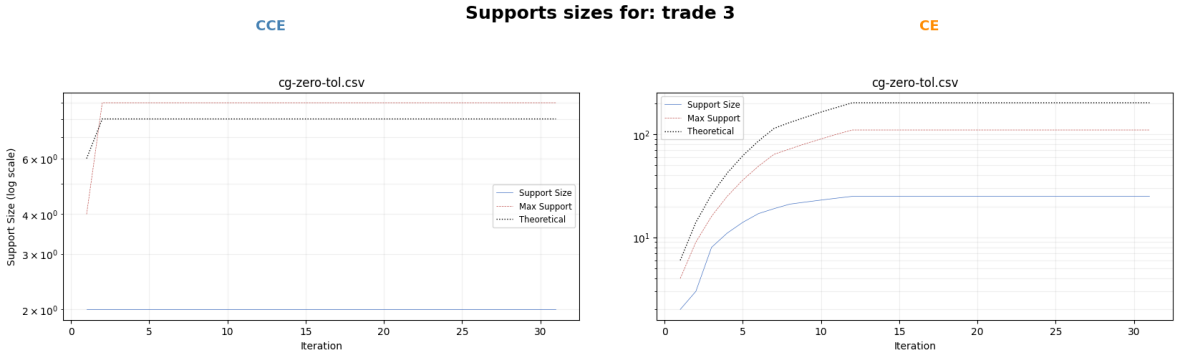


(c) Sheriff

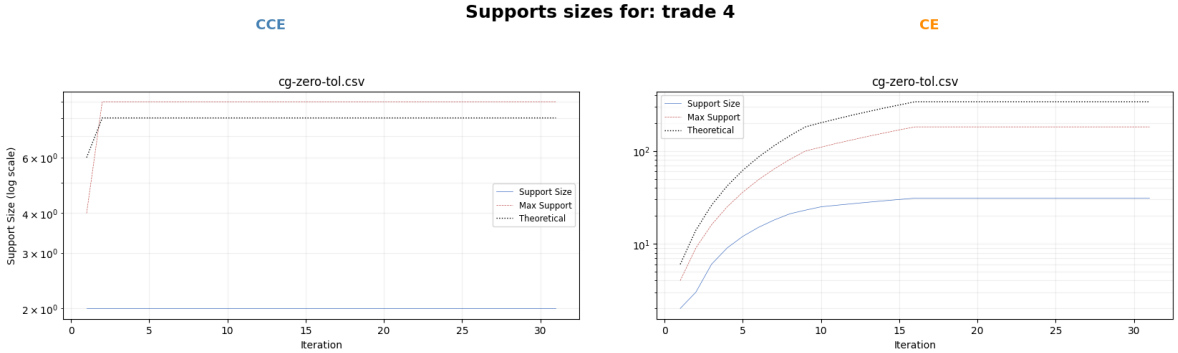
Figure C.7: Support size of JPSRO-CG over iterations on Kuhn 3, Kuhn 4, and Sheriff. In blue is the size of the supports at each iteration, in red is the maximum support possible (total number of joint actions in the meta game), and the dotted black line is the theoretical support bound. For the smaller games, the theoretical bound is larger than the maximum support because the bound is a function of the number of (C)CE incentive constraints, which can exceed the number of joint actions in small games.



(a) Tiny Bridge



(b) Trade 3



(c) Trade 4

Figure C.8: Support size of JPSRO-CG over iterations on Tiny Bridge, Trade 3, and Trade 4. In blue is the size of the supports at each iteration, in red is the maximum support possible (total number of joint actions in the meta game), and the dotted black line is the theoretical support bound.

D Pricing in Graphical Games

This appendix expands the brief reference to graphical games in Chapter 6. In a graphical game, let $N(i)$ denote the neighbourhood of player i (including i itself), and assume $u_i(\mathbf{a}) = u_i(\mathbf{a}_{N(i)})$. The pricing subproblem (3.3) then decomposes as

$$\mathbf{a}^* = \arg\max_{\mathbf{a}\in\mathbf{A}} \sum_{i=1}^{n} \Phi_i(\mathbf{a}_{N(i)}), \qquad \Phi_i(\mathbf{a}_{N(i)}) = \sum_{a_i' \neq a_i} \lambda_{i,a_i,a_i'} \big[u_i(\mathbf{a}_{N(i)}) - u_i(a_i', \mathbf{a}_{N(i)\setminus\{i\}})\big].$$

When the welfare functional f is itself locally decomposable (for example, social welfare $f(\mathbf{a}) = \sum_j u_j(\mathbf{a})$), the second-stage welfare pricing problem inherits the same form, with Φ_i replaced by

$$\Psi_i(\mathbf{a}_{N(i)}) = (1 - \Lambda_i)\, u_i(\mathbf{a}_{N(i)}) + \sum_{a_i' \neq a_i} \lambda^*_{i,a_i,a_i'}\, u_i(a_i', \mathbf{a}_{N(i)\setminus\{i\}}), \qquad \Lambda_i = \sum_{a_i' \neq a_i} \lambda^*_{i,a_i,a_i'}.$$

Maximising a sum of local potentials over a graphical interaction structure is the canonical setting for two families of solvers. Loopy belief propagation (Winsper and Chli [39]) gives an approximate max-product oracle that is efficient on bipartite interaction graphs, and suffices for column generation because any column with strictly positive reduced cost is admissible. Exact maxima can be obtained by an integer program with neighbourhood indicator variables $I_{N(i)}(\mathbf{a}_{N(i)}) \in \{0,1\}$ and per-player indicators $I_i(a_i) \in \{0,1\}$:

$$\max \sum_{i=1}^{n} \sum_{\mathbf{a}_{N(i)}} \Phi_i(\mathbf{a}_{N(i)})\, I_{N(i)}(\mathbf{a}_{N(i)}) \quad \text{s.t.} \quad \sum_{\mathbf{a}_{N(i)\setminus\{i\}}} I_{N(i)}(\mathbf{a}_{N(i)}) = I_i(a_i), \ \sum_{a_i \in A_i} I_i(a_i) = 1.$$

The ILP is used as a fallback when LBP fails to converge or returns a non-improving column (as seen in Barnhart et al. [4]); even for bipartite graphical games the exact welfare problem is NP-hard (Papadimitriou and Roughgarden [33]), so the LBP/ILP split is a pragmatic rather than theoretical division.

E Support is not confined to old support plus new joint actions

Here is an example of the claim in Chapter 6 that, when a new policy is admitted, the support S^{t+1} of a (C)CE of the expanded game is not in general contained in $S^t \cup A_{\text{new}}$, where A_{new} collects the joint actions involving the new policy.

Construction. Consider a two-player game with row actions $\{U, D\}$ and column actions $\{L, R\}$. In the 2×2 subgame (boxed below), the point mass on (U, L) is a (C)CE; when the opponent is known to play their half of (U, L), neither player can profitably deviate. So take $S^t = \{(U, L)\}$. Now add a new row action M:

	L	R
U	$\boxed{2, 2}$	$\boxed{0, 0}$
M	$4, 0$	$0, 3$
D	$\boxed{0, 0}$	$\boxed{1, 1}$

With $A_{\text{new}} = \{(M, L), (M, R)\}$, the hypothesised bound reads $S^{t+1} \subseteq \{(U, L), (M, L), (M, R)\}$.

No CE has support in $S^t \cup A_{\text{new}}$. Suppose p is a CE with $\text{supp}(p) \subseteq \{(U, L), (M, L), (M, R)\}$. Since each of U, L, M appears in exactly one candidate, conditioning on any recommendation pins down the opponent's action, so the CE constraints reduce to pure-strategy best responses. But $U \to M$ improves $2 \mapsto 4$, $L \to R$ improves $0 \mapsto 3$, and $M \to D$ improves $0 \mapsto 1$, ruling out (U, L), (M, L), and (M, R) in turn. The support must be empty, a contradiction. In fact the point mass on (D, R) is itself a CE of the expanded game (no unilateral deviation pays), and it uses the old action $D \notin S^t$.

CCE and implication. The CCE case follows from the same three deviations, now unconditional. The counterexample shows that admitting a new policy can force the next equilibrium onto old joint actions inactive in S^t, so column generation must admit columns from the full joint action space, and no inductive bound on $|S^{t+1}|$ in terms of $|S^t|$ and $|A_{\text{new}}|$ is available.